Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Epistemic GDL: A Logic for Representing and Reasoning about Imperfect Information Games

Guifei Jiang,[1,2] **Dongmo Zhang,**[1] **Laurent Perrussel,**[2] and **Heng Zhang**[3]
[1]AIRG, Western Sydney University, Penrith, Australia
[2]IRIT, Université Toulouse Capitole, Toulouse, France
[3]School of Computer Sci. & Tech., Huazhong University of Sci. & Tech., Wuhan, China

Abstract

This paper proposes a logical framework for representing and reasoning about imperfect information games. We first extend the game description language (GDL) with the standard epistemic operators and provide it with a semantics based on the epistemic state transition model. We then demonstrate how to use the language to represent the rules of an imperfect information game and formalize its epistemic properties. We also show how to use the framework to reason about player's own as well as other players' knowledge during game playing. Finally we prove that the model-checking problem of the framework is in Δ_2^p, which is the lowest among the existing similar frameworks, even though its lower bound is Θ_2^p. These results indicate that the framework makes a good balance between expressive power and computational efficiency.

1 Introduction

General Game Playing (GGP) is concerned with creating intelligent agents that understand the rules of arbitrary new games and learn to play these games without human intervention [Genesereth *et al.*, 2005]. Representing and reasoning about games is a core technique in GGP. A formal game description language (GDL) has been introduced as an official language for GGP since 2005. GDL is defined as a high-level, machine-processable language for representing the rules of arbitrary games [Love *et al.*, 2006]. Originally designed for perfect information games, GDL has recently been extended to GDL-II so as to incorporate imperfect information games, such as Poker, Backgammon [Thielscher, 2010].

Playing games with imperfect information poses an intricate reasoning challenge for players since imperfect information requires a player to use the rules of a game to infer legal actions, draw conclusions from her own knowledge about the current game state and about knowledge of other agents. However, as a purely descriptive language, GDL-II is only a tool for describing the rules of an imperfect information game but does not provide a facility for reasoning about how a player infers unveiled information based on the rules [Schiffel and Thielscher, 2011; 2014]. Indeed, some information is essential for players to proceed with a game. For example, players should always know their own available actions in non-terminal states and know their results in terminal states. Such epistemic properties of a game are normally implied by the game rules and thus need reasoning facilities to infer and verify them. Unfortunately, GDL-II (or GDL) is not designed for this purpose. To handle this issue, a few approaches have been proposed, mostly embedding GDL-II into a logical system, such as Situation Calculus or Alternating-time Temporal Epistemic Logic (ATEL), to use their reasoning facilities [Schiffel and Thielscher, 2011; Ruan and Thielscher, 2011; 2012; Huang *et al.*, 2013]. As long as the targeting logics are expressive enough to interpret any GDL description, it is possible to use the inference mechanisms of these logics for reasoning about GDL-II games. However, a highly expressive logic may incur high complexity for reasoning tasks. For instance, Ruan and Thielscher [2012] propose an adaption of ATEL to verify epistemic properties of GDL-II games, and show that the model-checking problem in that setting is 2EXPTIME-hard. Such high computational complexity may not be what we want.

This paper aims to propose a different approach to deal with this problem. We introduce a logical framework, called EGDL, equipped with a language for representing imperfect information games and a semantical model that can be used for reasoning about game information and players' epistemic status. More importantly, we develop a model-checking algorithm for EGDL and show that the complexity of the model-checking problem for the logic can be significantly reduced to Δ_2^p. There are two major reasons that help us to reduce the complexity. Firstly, our language is a conservative extension of GDL with the standard epistemic operators [Fagin *et al.*, 2003]. We take a cautious way of doing that without introducing the until operator or coalition operators. Secondly, we provide an imperfect recall semantics for knowledge. Other cases could be considered; nevertheless, the addition of perfect recall to GDL-II renders the model-checking problem of ATEL undecidable in general [Ruan and Thielscher, 2012]. Also, in many applications, especially when modeling extremely large games, imperfect recall may provide considerably empirical and practical advantages [Piccione and Rubinstein, 1997; Waugh *et al.*, 2009; Busard *et al.*, 2015]. Despite of a moderate expressive power, we demonstrate with a running example that our language is enough for expressing game rules, formalizing essential epis-

temic properties and specifying the interactions of knowledge and actions. In this sense, EGDL makes a good balance between expressive power and computational efficiency.

The rest of this paper is structured as follows: Section 2 establishes the syntax and the semantics of EGDL. Section 3 demonstrates its expressiveness and reasoning facility. Section 4 investigates the model-checking problem of EGDL. Finally, we conclude with a discussion of related work and future work.

2 The Framework

All games we consider in this paper are assumed to be played in multi-agent environments. A *game signature* $\mathcal{S}$ is a triple $(N, \mathcal{A}, \Phi)$, where

- $N = \{1, 2, \cdots, k\}$ is a non-empty finite set of agents,
- $\mathcal{A} = \bigcup_{i \in N} A^i$, where A^i consists of a non-empty finite set of *actions* for agent i s.t. $A^i \cap A^j = \emptyset$ if $i \neq j$, and
- $\Phi = \{p, q, \cdots\}$ is a finite set of propositional atoms for specifying individual features of a game state.

Hereafter we will fix a game signature $\mathcal{S}$ and all concepts will be based on the same game signature unless otherwise specified.

2.1 Epistemic State Transition Models

We begin by introducing the semantical structures used to model synchronous games with imperfect information. By synchronous we mean that all players move simultaneously. In particular, turn-based asynchronous games are modelled by only allowing the "noop" action for a player, when it is not her turn.

Definition 1. *An* epistemic state transition (ET) model M *is a tuple* $(W, \overline{w}, T, \{R_i\}_{i \in N}, L, U, g, V)$*, where*

- W *is a non-empty set of* possible states*.*
- $\overline{w} \in W$*, representing the* initial *state.*
- $T \subseteq W$*, representing the set of* terminal *states.*
- $R_i \subseteq W \times W$ *is an equivalence relation for agent* i*, indicating the states that are indistinguishable for* i*.*
- $L \subseteq W \times \mathcal{A}$ *is a* legality *relation, describing the legal actions at each state.*
- $U : W \times D \to W$ *is an* update *function, specifying state transitions, where* D *is the set of joint actions* $\prod_{i \in N} A^i$*.*
- $g : N \to 2^W$ *is a* goal *function, specifying the winning states for each agent.*
- $V : W \to 2^\Phi$ *is a standard valuation function.*

For $d \in D$, let $d(i)$ denote the i-th component of d. That is, agent i's action in the joint action d. For convenience, let $L(w) = \{a \in \mathcal{A} \mid (w, a) \in L\}$ denote the set of all legal actions at state w. We now define the set of all possible ways in which a game can develop.

Definition 2. *Let* $M = (W, \overline{w}, T, \{R_i\}_{i \in N}, L, U, g, V)$ *be an ET-model. A path* δ *is a sequence of states and joint actions* $\overline{w} \xrightarrow{d_1} \cdots \xrightarrow{d_e} w_e$ *such that for all* $1 \leq j \leq e$ *and any* $i \in N$*,*

1. $d_j(i) \in L(w_{j-1})$ *(that is, any action that is taken must be legal),*
2. $w_j = U(w_{j-1}, d_j)$ *(state transition), and*
3. $\{\overline{w}, \cdots, w_{e-1}\} \cap T = \emptyset$ *(that is, only the last state may be terminal).*

A path δ is *complete* if $w_e \in T$. Let $\mathcal{P}(M)$ denote the set of all complete paths in M. When M is fixed, we simply write $\mathcal{P}$. Given $\delta \in \mathcal{P}$, the states on δ are called *reachable states*. Let $\delta[j]$ denote the j-th reachable state of δ and $\theta_i(\delta, j)$ denote the action of agent i taken at stage j of δ. The length of δ, written $|\delta|$, is defined as the number of actions.

The following definition, by extending equivalence relations over states to complete paths, characterizes precisely what an agent with imperfect recall and perfect reasoning can in principle know at a specific stage of a game.

Definition 3. *Two complete paths* δ*,* $\delta' \in \mathcal{P}$ *are imperfect recall (also called memoryless) equivalent for agent* i *at stage* $j \in \mathbb{N}$*, written* $\delta \approx_i^j \delta'$*, iff* $\delta[j] R_i \delta'[j]$*.*

That is, imperfect recall requires an agent to be only aware of the present state but forget everything that happened. This is similar to the notion of imperfect recall in ATL [Schobbens, 2004]. It should be noted that the paper focuses on imperfect recall; nevertheless, the framework is flexible to define ATL state-based perfect recall [Jamroga and van der Hoek, 2004] and GDL-II perfect recall [Thielscher, 2010].

To illustrate our framework, we use as our running example a variant of the Tic-Tac-Toe, called Krieg-Tictactoe in [Schiffel and Thielscher, 2011].

Example. *Krieg-Tictactoe is played by two players, cross* x *and naught* o*, who take turns marking grids in a* 3×3 *board. Different from standard Tic-Tac-Toe, each player can see her own marks, but not those of her opponent, just like the chess variant* Kriegspiel *[Pritchard, 1994]. Players are informed of the turn-taking. The game ends if the board is completely filled or one player wins by having completed a horizontal, vertical or diagonal line of three with her own symbol.*

To represent Krieg-Tictactoe in terms of the ET-model, we first describe the game signature, written $\mathcal{S}_{KT}$, as follows: let $N_{KT} = \{\mathsf{x}, \mathsf{o}\}$, $A^i_{KT} = \{a^i_{j,k} \mid 1 \leq j, k \leq 3\} \cup \{noop^i\}$, and $\Phi_{KT} = \{p^i_{j,k}, tried(a^i_{j,k}), turn(i) \mid i \in \{\mathsf{x}, \mathsf{o}\} \text{ and } 1 \leq j, k \leq 3\}$, where $a^i_{j,k}$ denotes the action that player i fills grid (j, k) with her symbol, $noop^i$ denotes that player i does action noop, $p^i_{j,k}$ represents the fact that grid (j, k) is filled with player i's symbol, $tried(a^i_{j,k})$ represents the fact that player i has tried to fill grid (j, k) before, and $turn(i)$ says that it is player i's turn now.

We next specify the ET-model for this game, written M_{KT}, as follows: let $W_{KT} = \{(t_\mathsf{x}, t_\mathsf{o}, x_{1,1}, \cdots, x_{3,3}) : t_\mathsf{x}, t_\mathsf{o} \in \{0, 1\} \;\&\; x_{1,1}, \cdots, x_{3,3} \in \{\square, \boxtimes, \boxdot, \otimes, \odot\}\}$ be the set of possible states, where t_x, t_o specify the turn taking and $x_{i,j}$ represents the fact that grid (i, j) is occupied by the cross and not tried by the nought $\boxtimes$, occupied by the nought and not tried by the cross $\boxdot$, occupied by the nought and tried by the cross $\otimes$, occupied by the cross and tried by the nought $\odot$, or empty $\square$. The initial state $\overline{w}$ is $(1, 0, \square, \cdots, \square)$.

For any two states $w = (t_{\mathsf{x}}, t_{\mathsf{o}}, x_{1,1}, \cdots, x_{3,3})$ and $w' = (t'_{\mathsf{x}}, t'_{\mathsf{o}}, x'_{1,1}, \cdots, x'_{3,3})$ in W_{KT}, $wR_{\mathsf{x}}w'$ iff (1) $t_i = t'_i$ for any $i \in N_{KT}$, (2) $x_{j,k} \in \{\boxtimes, \odot\}$ iff $x'_{j,k} \in \{\boxtimes, \odot\}$ for any $1 \leq j, k \leq 3$, and (3) $x_{j,k} = \otimes$ iff $x'_{j,k} = \otimes$ for any $1 \leq j, k \leq 3$. The equivalence relation for o is defined in a similar way. Let V_{KT} be a valuation such that for each state $w = (t_{\mathsf{x}}, t_{\mathsf{o}}, x_{1,1}, \cdots, x_{3,3}) \in W_{KT}$, $V_{KT}(w) = \{turn(i) : t_i = 1\} \cup \{p^{\mathsf{x}}_{j,k} : x_{j,k} \in \{\boxtimes, \odot\}\} \cup \{p^{\mathsf{o}}_{j,k} : x_{j,k} \in \{\boxdot, \otimes\}\} \cup \{tried(a^{\mathsf{x}}_{j,k}) : x_{j,k} = \otimes\} \cup \{tried(a^{\mathsf{o}}_{j,k}) : x_{j,k} = \odot\}$. Moreover, we assume that each player takes the same action at stages of all her indistinguishable complete paths, i.e., $\theta_i(\delta, j) = \theta_i(\delta', j)$ whenever $\delta \approx^j_i \delta'$. Due to the space limit, we refrain from explicitly listing the legality relation, the update function, and the terminal and goal states for the agents as this is possible but considerably lengthy. However, the syntactic descriptions of the game given in the following section detail them in a more compact way.

2.2 The Language

Let us now introduce an epistemic extension of a variant of GDL [Zhang and Thielscher, 2015b] to represent games with imperfect information, and further provide a semantics for the language based on the epistemic state transition model. We call this framework EGDL for short.

Definition 4. *The language, denoted by $\mathcal{L}$, consists of*

- *the finite set Φ of propositional atoms;*
- *pre-defined propositions: $initial$, $terminal$, $wins(\cdot)$, $legal(\cdot)$ and $does(\cdot)$;*
- *logical connectives $\neg$ and $\wedge$;*
- *temporal operator $\bigcirc$;*
- *epistemic operators: K and C.*

A formula φ in $\mathcal{L}$ is defined by the following BNF:

$$\varphi ::= p \mid initial \mid terminal \mid legal(a^i) \mid wins(i) \mid does(a^i) \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi \mid \mathsf{K}_i\varphi \mid \mathsf{C}\varphi$$

where $p \in \Phi$, $i \in N$ and $a^i \in A^i$.

Other connectives $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ are defined by $\neg$ and $\wedge$ in a standard way. Intuitively, $initial$ and $terminal$ specify the initial state and the terminal states of a game, respectively; $does(a^i)$ asserts that agent i takes action a at the current state; $legal(a^i)$ asserts that action a is available to agent i at the current state; and $wins(i)$ asserts that agent i wins at the current state. The formula $\bigcirc\varphi$ means "φ holds in the next state". All these components are inherited from GDL. The epistemic operators K and C are taken from the Modal Epistemic Logic [Fagin *et al.*, 2003]. The formula $\mathsf{K}_i\varphi$ is read as "agent i knows φ", and $\mathsf{C}\varphi$ as "φ is common knowledge among all the agents in N".

We use the following abbreviations in the rest of the paper:

$$\widehat{\mathsf{K}}_i\varphi =_{def} \neg\mathsf{K}_i\neg\varphi \quad \widehat{\mathsf{C}}\varphi =_{def} \neg\mathsf{C}\neg\varphi \quad \mathsf{E}\varphi =_{def} \bigwedge_{i\in N} \mathsf{K}_i\varphi$$

where $\widehat{\mathsf{K}}_i$ and $\widehat{\mathsf{C}}$ are the dual operators of K_i and C, respectively. $\widehat{\mathsf{K}}_i\varphi$ says "φ is compatible with agent i's knowledge" and it is similar to $\widehat{\mathsf{C}}\varphi$. $\mathsf{E}\varphi$ says "every agent in N knows φ".

Let us illustrate the intuition of the language with our running example.

Example (*continued*) The rules of Krieg-Tictactoe are specified by EGDL in Figure 1.

1. $initial \leftrightarrow turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge^3_{j,k=1}(\neg(p^{\mathsf{x}}_{j,k} \vee p^{\mathsf{o}}_{j,k}) \wedge \neg(tried(a^{\mathsf{x}}_{j,k}) \vee tried(a^{\mathsf{o}}_{j,k})))$
2. $wins(i) \leftrightarrow (\bigvee^3_{j=1}\bigwedge^2_{l=0} p^i_{j,1+l}) \vee (\bigvee^3_{k=1}\bigwedge^2_{l=0} p^i_{1+l,k}) \vee (\bigwedge^2_{l=0} p^i_{1+l,1+l}) \vee (\bigwedge^2_{l=0} p^i_{1+l,3-l})$
3. $teminal \leftrightarrow wins(\mathsf{x}) \vee wins(\mathsf{o}) \vee \bigwedge^3_{j,k=1}(p^{\mathsf{x}}_{j,k} \vee p^{\mathsf{o}}_{j,k})$
4. $turn(i) \rightarrow \bigcirc\neg turn(i) \wedge \bigcirc turn(-i)$
5. $legal(noop^i) \leftrightarrow turn(-i)$, where $-i$ represents i's opponent
6. $legal(a^i_{j,k}) \leftrightarrow turn(i) \wedge \neg p^i_{j,k} \wedge \neg tried(a^i_{j,k})$
7. $\bigcirc p^i_{j,k} \leftrightarrow terminal \vee p^i_{j,k} \vee (does(a^i_{j,k}) \wedge \neg(p^{\mathsf{x}}_{j,k} \vee p^{\mathsf{o}}_{j,k}))$
8. $\bigcirc tried(a^i_{j,k}) \leftrightarrow terminal \vee tried(a^i_{j,k}) \vee (does(a^i_{j,k}) \wedge p^{-i}_{j,k})$
9. $does(a^i_{j,k}) \rightarrow \mathsf{K}_i(does(a^i_{j,k}))$
10. $initial \rightarrow \mathsf{E}initial$
11. $(turn(i) \rightarrow \mathsf{E}turn(i)) \wedge (\neg turn(i) \rightarrow \mathsf{E}\neg turn(i))$
12. $(p^i_{j,k} \rightarrow \mathsf{K}_i p^i_{j,k}) \wedge (\neg p^i_{j,k} \rightarrow \mathsf{K}_i \neg p^i_{j,k})$
13. $(tried(a^i_{j,k}) \rightarrow \mathsf{K}_i tried(a^i_{j,k})) \wedge (\neg tried(a^i_{j,k}) \rightarrow \mathsf{K}_i \neg tried(a^i_{j,k}))$

Figure 1: An EGDL description of Krieg-Tictactoe.

The initial state, each player's winning states, the terminal states and the turn-taking are given by rules 1-4.

The preconditions of each action are specified by Rule 5 and Rule 6. *The player who has the turn can fill any grid s.t. (i) it is not filled by herself, and (ii) she has never tried to fill the grid before. The other player can only do action $noop$.*

Rules 7 and 8 are the combination of the frame axioms and the effect axioms [Reiter, 1991]. Rule 7 states that *a grid is marked with a player's symbol in the next state if the player takes the corresponding action at the current state, or the grid has been filled by herself, or the game ends.* Similarly, Rule 8 says that *an action is tried by a player in the next state if the action is ineffective while still taken by the player at the current state, or it has been tried before, or the game ends.*

The others are the epistemic rules. Rule 9 states each player knows which action she is taking. Rule 10 and Rule 11 says both players know the initial state and the turn-taking, respectively. Rule 12 says that *each player knows which grid is filled or not by her own symbol.* Similarly, Rule 13 states that *each player knows which grid is tried or not by herself.*

Let Σ_{KT} be the set of rules 1-13. It should be noted that rules 11-13 together specify the epistemic relations for each player: *two states are indistinguishable for a player if their configurations of the game board are the same in her view*.

2.3 Semantics

The semantics of the language is based on the epistemic state transition model.

Definition 5. *Let M be an ET-model. Given a complete path δ in M, a stage j of δ and a formula $\varphi \in \mathcal{L}$, we say φ is true at j of δ under M, denoted by $M, \delta, j \models \varphi$, according to the following definition*:

$M, \delta, j \models p$ *iff* $p \in V(\delta[j])$
$M, \delta, j \models \neg\varphi$ *iff* $M, \delta, j \not\models \varphi$
$M, \delta, j \models \varphi_1 \wedge \varphi_2$ *iff* $M, \delta, j \models \varphi_1$ *and* $M, \delta, j \models \varphi_2$
$M, \delta, j \models initial$ *iff* $\delta[j] = \overline{w}$
$M, \delta, j \models terminal$ *iff* $\delta[j] \in T$
$M, \delta, j \models wins(i)$ *iff* $\delta[j] \in g(i)$
$M, \delta, j \models legal(a^i)$ *iff* $a^i \in L(\delta[j])$
$M, \delta, j \models does(a^i)$ *iff* $\theta_i(\delta, j) = a^i$
$M, \delta, j \models \bigcirc\varphi$ *iff* *if* $j < |\delta|$, *then* $M, \delta, j+1 \models \varphi$
$M, \delta, j \models \mathsf{K}_i\varphi$ *iff* *for any* $\delta' \in \mathcal{P}$ *with* $\delta \approx_i^j \delta'$, $M, \delta', j \models \varphi$
$M, \delta, j \models \mathsf{C}\varphi$ *iff* *for any* $\delta' \in \mathcal{P}$ *with* $\delta \approx_N^j \delta'$, $M, \delta', j \models \varphi$

where $\approx_N^j$ *is its transitive closure of* $\bigcup_{i\in N} \approx_i^j$.

A formula φ is *globally true* in an ET-model M, written $M \models \varphi$, if $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}$ and any $0 \le j \le |\delta|$. A formula φ is *valid*, written $\models \varphi$, if $M \models \varphi$ for any ET-model M. A formula φ is called *true at a state* w in M, written $M, w \models \varphi$, if it is true for all complete paths going through w, i.e., $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}$ and any $j \ge 0$ with $\delta[j] = w$. Finally, let Σ be a set of formulas in $\mathcal{L}$, then M is a *model* of Σ if $M \models \varphi$ for all $\varphi \in \Sigma$.

We now show that EGDL provides a sound description for Krieg-Tictactoe.

Proposition 1. *M_{KT} is a model of Σ_{KT}.*

It follows that these game rules are common knowledge among two players, which is just what we expect.

Corollary 1. *$M_{KT} \models \mathsf{C}\varphi$ for all $\varphi \in \Sigma_{KT}$.*

3 Epistemic and Strategic Reasoning

In this section, we demonstrate the expressive power and flexibility of EGDL by showing how it allows us to specify epistemic properties and reason about agents' knowledge.

3.1 Epistemic Properties

The introduction of imperfect information raises new epistemic properties of a game. For instance, to make a game playable, each player should always know her own legal actions in the course of the game. This property as well as some other well-known properties can be naturally formulated by EGDL as follows: given $i \in N$ and $a^i \in A^i$,

(1) $initial \to \mathsf{C}\, initial$ (2) $legal(a^i) \to \mathsf{K}_i(legal(a^i))$
(3) $does(a^i) \to \mathsf{K}_i(does(a^i))$ (4) $wins(i) \to \mathsf{K}_i(wins(i))$
(5) $terminal \to \mathsf{C}\, terminal$

Formulas (1) and (5) express that the initial state, the terminal states are common knowledge, respectively. Formula (2) says that each agent knows her own legal actions. In ATEL, this is a required semantic property yet with no syntactic expression [Ågotnes, 2006]. Formula (3) asserts that each agent is aware of her own actions. This is called the "uniform" property of actions (strategies) also with no syntactic counterpart in ATEL [Van der Hoek and Wooldridge, 2003; Jamroga and van der Hoek, 2004]. Finally, formula (4) specifies that each agent should know her winning result.

Moreover, the above epistemic properties are precisely characterised by indistinguishable complete paths as follows:

Proposition 2. *Let M be an ET-model. Then*

1. $M \models initial \to \mathsf{C}\, initial$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_N^j \delta'$, *then* $(\delta[j] = \overline{w}$ *iff* $\delta'[j] = \overline{w})$.
2. $M \models legal(a^i) \to \mathsf{K}_i(legal(a^i))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_i^j \delta'$, *then* $(a^i \in L(\delta[j])$ *iff* $a^i \in L(\delta'[j]))$.
3. $M \models does(a^i) \to \mathsf{K}_i(does(a^i))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_i^j \delta'$, *then* $(\theta_i(\delta, j) = a^i$ *iff* $\theta_i(\delta', j) = a^i)$.
4. $M \models wins(i) \to \mathsf{K}_i(wins(i))$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_i^j \delta'$, *then* $(\delta[j] \in g(i)$ *iff* $\delta'[j] \in g(i))$.
5. $M \models terminal \to \mathsf{C}\, terminal$ *iff for all* $\delta, \delta' \in \mathcal{P}$ *and any* $j \in \mathbb{N}$, *if* $\delta \approx_N^j \delta'$, *then* $(\delta[j] \in T$ *iff* $\delta'[j] \in T)$.

Obviously, not all games with imperfect information satisfy these epistemic properties. For instance, property (5) does not hold for Krieg-Tictactoe. Consider the two reachable states depicted in Figure 2. They are indistinguishable

o	x	o
	x	
	x	

o		o
	x	
x		x

Figure 2: The indistinguishable states for o

for player o. Yet the left one is a terminal state while the right one is not. In fact, according to the game rules, Krieg-Tictactoe satisfies all the other properties.

Observation 1. *Formulas (1)-(4) are globally true in M_{KT}.*

It should be noted that each EGDL-formula may be interpreted as a property of a game. Typically, globally true formulas describe properties for a particular game, such as the rules for Krieg-Tictactoe, while valid formulas specify general properties of a class of games and thus can be used to classify games. For instance, different from Krieg-style board games, most card games have the property (5).

3.2 Strategic Reasoning

Let us now show how to use EGDL to reason about agents' knowledge and actions based on the game rules. In the context of imperfect information, epistemic reasoning is closely related to strategic reasoning. To start with, the following proposition shows that EGDL is suitable for reasoning about players' knowledge as it is a conservative extension of the standard Epistemic Modal Logic $S5_n^C$ [Fagin *et al.*, 2003].

Proposition 3. *Given an EGDL-formula φ without involving the operator $\bigcirc$ and the pre-defined propositions, φ is valid in EGDL iff it is valid in $S5_n^C$.*

This result indicates that EGDL is sufficient to provide a static characterization of agents' knowledge at a certain stage. For instance, with $S5_n^C$, we can derive the following formulas from the rules of Krieg-Tictactoe.

Observation 2.

1. $M_{KT} \models turn(i) \to \mathsf{C}turn(i)$
2. $M_{KT} \models \mathsf{K}_i(\mathsf{K}_{-i}p_{j,k}^{-i} \vee \mathsf{K}_{-i}\neg p_{j,k}^{-i})$
3. $M_{KT} \models \mathsf{K}_i tried(a_{j,k}^i) \to \mathsf{K}_i p_{j,k}^{-i}$

Clause 1 says the turn-taking is common knowledge. Clause 2 says a player knows the opponent knows whether or not a grid is filled by herself. The last one says if a player knows she has tried an action, then she knows the corresponding grid has been filled by the opponent. The last two properties are important when players gather information.

Furthermore, with the full expressive power of the language, we can use EGDL to specify agents' knowledge of particular game features and reason about how agent's knowledge changes as a game progresses.

Observation 3.

1. $M_{KT} \models \mathsf{K}_i p^i_{j,k} \to \mathsf{K}_i \bigcirc p^i_{j,k}$
2. $M_{KT} \models \mathsf{K}_i \bigcirc tried(a^i_{j,k}) \to \bigcirc \mathsf{K}_i tried(a^i_{j,k})$
3. $M_{KT} \models initial \to \mathsf{C}(\bigwedge^3_{j,k=1} legal(a^{\mathsf{x}}_{j,k}) \wedge legal(noop^{\mathsf{o}}))$
4. $M_{KT} \models does(a^i_{j,k}) \to \bigcirc \mathsf{K}_i(p^i_{j,k} \vee tried(a^i_{j,k}))$

Intuitively, clause 1 says that if a player knows a grid has been filled by herself, then she still knows this fact holds at the next state. Clause 2 says that a player is able to remember the grid she has tried to fill before. Clause 3 says that at the initial state the legal actions are common knowledge among two players, and Clause 4 expresses that if a player takes an action now, then at the next state she will know either the corresponding grid has been filled by her symbol or she has tried that action.

Most importantly, the interactions of actions and knowledge can be naturally formulated using EGDL. Specifically, they interact in three different ways:

(i) Knowledge is necessary for an agent to perform an action, which may be formulated by $does(a^i) \to \mathsf{K}_i\varphi$. For instance, in Krieg-Tictactoe, with partial observation, a player might take an ineffective action by trying to fill a grid which has been filled by the opponent. Then we say a player i takes a good action $a^i_{j,k}$, written $good(a^i_{j,k})$, if it is effective. It follows that, to take a good action, a player needs to know the grid she attempts to fill is empty. Formally, $M_{KT} \models good(a^i_{j,k}) \to \mathsf{K}_i(\neg(p^{\mathsf{x}}_{j,k} \vee p^{\mathsf{o}}_{j,k}))$.

(ii) Performing an action may increase an agent's knowledge, which may be specified by $does(a^i) \to \bigcirc \mathsf{K}_i\varphi$. For example, if a player takes an ineffective action, then she would know the corresponding grid has been filled by the other player. Consider the following complete path

$$\delta = \overline{w} \overset{\langle a^{\mathsf{x}}_{2,2}, noop^{\mathsf{o}}\rangle}{\to} w_1 \overset{\langle noop^{\mathsf{x}}, a^{\mathsf{o}}_{2,2}\rangle}{\to} w_2 \overset{\langle a^{\mathsf{x}}_{1,1}, noop^{\mathsf{o}}\rangle}{\to} w_3 \cdots$$

At stage 2 after player o tries to fill grid (2,2), by Rule 7 and Rule 13, she knows that the grid has been filled by player x. Thus, $M_{KT}, \delta, 1 \models does(a^{\mathsf{o}}_{2,2}) \to \bigcirc \mathsf{K}_{\mathsf{o}}(tried(a^{\mathsf{o}}_{2,2}) \wedge p^{\mathsf{x}}_{2,2})$.

(iii) An agent makes her choice of actions based on her knowledge, which may be captured by $\mathsf{K}_i\varphi \to does(a^i)$. Let us consider the following two basic actions:

$$attack^i =_{def} \mathsf{K}_i(does(a^i_{j,k}) \wedge \bigcirc wins(i)) \to does(a^i_{j,k})$$
$$block^i =_{def} \mathsf{K}_i \bigcirc (does(a^{-i}_{j,k}) \wedge \bigcirc wins(-i)) \to does(a^i_{j,k})$$

Intuitively, $attack$ says if a player knows that filling a grid leads to win, then she should fill that grid. Instead, $block$ says if a player knows her opponent makes to win by filling a grid in the next state, then the player must fill that grid at the current state to avoid an immediate loss.

4 Model Checking

In this section, we investigate the complexity of the model-checking problem for EGDL and develop a model-checking algorithm for EGDL.

The *model checking problem* for EGDL, denoted by EGDL-MC, is the following: Given an EGDL-formula φ, an ET-model M, a path δ of M and a stage j on δ, determine whether $M, \delta, j \models \varphi$ or not. In principle, two variants of EGDL-MC can be defined as follows: Given an ET-model M, a state w of M and an EGDL-formula φ, determine whether $M, w \models \varphi$ and determine whether $M \models \varphi$. It should be noted that all the bounds presented in this section remain true for these variants. Proofs are similar to those of EGDL-MC, or can be obtained by simple reductions to/from EGDL-MC.

Let us first consider the upper bound of the complexity for model-checking. Our goal is to show the following bound.

Theorem 1. *EGDL-MC is in Δ^p_2.*

To prove this upper bound, according to the definition of Δ^p_2, we need to prove that there is a polynomial-time deterministic Turing machine $\mathcal{M}$ with an NP-oracle such that $\mathcal{M}$ solves the model-checking problem for EGDL. To show the existence, let us start with a simple property of EGDL.

Let φ be an EGDL-formula, and $M = (\mathcal{G}, V)$ be an ET-model over $\mathcal{S}$. Take ψ to be any subformula of φ of the form $\oplus\vartheta$, where $\oplus$ is either C or K_i for some $i \in N$. We introduce a fresh propositional atom p_ψ for ψ. Let M_ψ be the ET-model $(\mathcal{G}, V_\psi)$ where V_ψ is a valuation function defined by

$$V_\psi(w) := \begin{cases} V(w) \cup \{p_\psi\} & \text{if } M, w \models \psi; \\ V(w) & \text{otherwise} \end{cases}$$

for each state w of M. Let φ_ψ denote the formula obtained from φ by replacing ψ by p_ψ. Then, by the definition of semantics for EGDL, the following property is clearly true.

Lemma 1. *For every path δ of M and every stage j on δ, it holds that $M, \delta, j \models \varphi$ iff $M_\psi, \delta, j \models \varphi_\psi$.*

Thus, by applying the above lemma, the epistemic operators can be eliminated from the formula in a recursive way. For the EGDL-formulas without epistemic operators, we can show that its model-checking problem is tractable.

Lemma 2. *The following problem is in PTIME: Given an ET-model M, a path δ of M, a stage j on δ and an EGDL-formula φ without involving any epistemic operators, determine whether $M, \delta, j \models \varphi$ or not.*

Due to the space limit, we omit the proof here. Roughly speaking, in the context of model checking, the operator $\bigcirc$ can be simply regarded as a standard modal operator. Note that model-checking for the basic modal logic is in PTIME.

To construct the model M_ψ, the truth value of ψ under M and a given state is needed to be evaluated. To simplify the question, let us first consider a simple case as follows.

Lemma 3. *The following problem is in NP: Given an ET-model M, a state w of M and an EGDL-formula $\oplus\varphi$ where $\oplus \in \{\widehat{\mathsf{C}}\} \cup \{\widehat{\mathsf{K}}_i : i \in N\}$ and φ does not involve any epistemic operators, determine whether $M, w \models \oplus\varphi$ or not.*

This lemma holds due to the observation that, given any path δ of M and any $j \geq 0$, we have $M, \delta, j \models \varphi$ if, and

only if, $M, \delta[j, j+k], 0 \models \varphi$, where k is the number of occurrences of $\bigcirc$ in φ, and $\delta[j, j+k]$ denotes the segment of δ starting from position j with length k. With it, one can design a nondeterministic Turing machine which first guesses a path of length $\leq k$, and then check the truth value of φ under this path. By Lemma 2, the later can be done in PTIME.

To eliminate all the epistemic operators, it remains to consider the formulas with nested epistemic operator. With this complexity result for the non-nested case, we are now able to design an algorithm for the general case. Roughly speaking, the idea is to carry out the elimination of epistemic operator in a bottom-up way. As we can see in Algorithm 1, such an idea is implemented in the algorithm $elimeop$. It's easy to check

Input : an ET-model M and an EGDL-formula φ
Output: an ordered pair (M_0, φ_0)
begin
  switch φ **do**
    case φ *is atomic* **do**
      $M_0 \leftarrow M$; $\varphi_0 \leftarrow \varphi$;
    case φ *is of the form* $\oplus\psi$*, where* $\oplus \in \{\neg, \bigcirc\}$ **do**
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $M_0 \leftarrow N_0$; $\varphi_0 \leftarrow \oplus\psi_0$;
    case φ *is of the form* $\psi \wedge \chi$ **do**
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $(N_0, \chi_0) \leftarrow elimeop(N_0, \chi)$;
      $M_0 \leftarrow N_0$; $\varphi_0 \leftarrow \psi_0 \wedge \chi_0$;
    case φ *is of the form* $\oplus\psi$*, where* $\oplus \in \{\mathsf{C}, \mathsf{K}_i\}$ **do**
      $(N_0, \psi_0) \leftarrow elimeop(M, \psi)$;
      $V \leftarrow$ the valuation function of N_0;
      for all w **in** W **do**
        if $N_0, w \models \widehat{\oplus}\neg\psi_0$ *is false* **then**
          $V(w) \leftarrow V(w) \cup \{p_{\oplus\psi}\}$;
        end
      end
      $M_0 \leftarrow$ the model obtained from N_0 by replacing the valuation function with V;
      $\varphi_0 \leftarrow p_{\oplus\psi}$;
  end
  return (M_0, φ_0);
end

Algorithm 1: $elimeop(M, \varphi)$

that the algorithm can be implemented in a polynomial-time deterministic Turing machine, but with an NP-oracle. Here the procedure of checking $N_0, w \models \widehat{\oplus}\neg\psi_0$ is used as the NP-oracle. By Lemma 3, the checking is in NP. In addition, Algorithm $elimeop$ visits each subformula of φ at most once, and that the number of subformulas of φ is not greater than the size of φ. These assure that the Turing machine will terminate in a polynomial number of stages.

With this algorithm, we can then devise an algorithm mc for the model-checking of EGDL such that, given any proper M, δ, j and φ as input, mc works as follows:

- Firstly, mc call the algorithm $elimeop(M, \varphi)$, and let (M_0, φ_0) be the results of this call.
- Next, mc check whether $M_0, \delta, j \models \varphi_0$ or not, and return "true" if it holds, "false" otherwise.

By Lemma 1 and the definition of algorithm $elimeop$, it is not difficult to verify that $M, \delta, j \models \varphi$ if, and only if, $M_0, \delta, j \models \varphi_0$. This assures the soundness of the algorithm mc. On the other hand, by the previous analysis, the first stage can be implemented in a polynomial-time determinitic Turing machine with an NP-oracle; by Lemma 2, the second stage can be done in PTIME. Thus, the algorithm mc can be implemented in a polynomial-time determinitic Turing machine with an NP-oracle, which proves Theorem 1.

Next we identify a lower bound of the complexity of model-checking for EGDL.

Theorem 2. *EGDL-MC is* Θ_2^p*-hard.*

Due to the space limit, we skip the proof here. The basic idea is to reduce the validity problem for Carnap's modal logic $\mathbb{C}$ to the model-checking problem of EGDL. The former has been proved to be Θ_2^p-complete [Gottlob, 1995].

It should be noted that the lower bound shows that the algorithm mc is nearly optimal, since Δ_2^p and Θ_2^p are very close together, which both lie in the second level of the polynomial hierarchy.

5 Conclusion

We have presented a logical framework for representing and reasoning about imperfect information games with imperfect recall players. The framework allows us to represent game rules, formalize epistemic properties, specify the interactions of knowledge and actions as well as reason about agents' knowledge during game playing. We have also investigated the model-checking problem for the logic. These results indicate that we have made a reasonable compromise between expressive power and computational efficiency.

Most of the related work has been discussed in the Introduction. Besides that, the following is also worth mentioning. Ruan and Thielscher [2011] study the epistemic structure and expressiveness of GDL-II in terms of epistemic modal logic. They mainly provide a static characterization of players' knowledge at a certain stage without involving the temporal dimension. Haufe and Thielscher [2012] develop an automated reasoning method to deal with epistemic properties for GDL-II. Different from ours, their method is restricted to positive-epistemic formulas. Our underlying language is from [Zhang and Thielscher, 2015b]. It was originally proposed for reasoning about strategies of asynchronous games with perfect information, while we investigate its epistemic extension for representing and reasoning about synchronous games with imperfect information.

Directions of future research are manifold. As we have mentioned, besides imperfect recall, the framework is flexible enough to specify other memory types. To obtain a complete picture of the relation between perfect or imperfect information, and perfect or imperfect recall, we plan to study properties of these memory types; We also want to investigate the satisfiability problem and the axiomatization of EGDL based on the current literature [Zhang and Thielscher, 2015a; Halpern *et al.*, 2004].

Acknowledgments

We are grateful to Michael Thielscher for his valuable suggestions, and special thanks are due to four anonymous ref-

erees for their insightful comments. This research was partially supported by ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02.

References

[Ågotnes, 2006] Thomas Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):375–407, 2006.

[Busard *et al.*, 2015] Simon Busard, Charles Pecheur, Hongyang Qu, and Franco Raimondi. Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Information and Computation*, 242:128–156, 2015.

[Fagin *et al.*, 2003] Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about Knowledge*. MIT press, 2003.

[Genesereth *et al.*, 2005] Michael Genesereth, Nathaniel Love, and Barney Pell. General game playing: Overview of the aaai competition. *AI magazine*, 26(2):62–72, 2005.

[Gottlob, 1995] G. Gottlob. NP trees and Carnap's modal logic. *Journal of the ACM*, 42(2):421–457, 1995.

[Halpern *et al.*, 2004] Joseph Y Halpern, Ron Van Der Meyden, and Moshe Y Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703, 2004.

[Haufe and Thielscher, 2012] Sebastian Haufe and Michael Thielscher. Automated verification of epistemic properties for general game playing. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pages 339–349, 2012.

[Huang *et al.*, 2013] Xiaowei Huang, Ji Ruan, and Michael Thielscher. Model checking for reasoning about incomplete information games. In *Proceedings of the 26th Australasian Joint Conference on Advances in Artificial Intelligence (AI'13)*, pages 246–258, 2013.

[Jamroga and van der Hoek, 2004] Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2):185–219, 2004.

[Love *et al.*, 2006] Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza, and Michael Genesereth. General game playing: Game description language specification. Stanford Logic Group Computer Science Department Stanford University. http://logic.stanford.edu/reports/LG-2006-01.pdf, 2006.

[Piccione and Rubinstein, 1997] Michele Piccione and Ariel Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, 20(1):3–24, 1997.

[Pritchard, 1994] David Brine Pritchard. *The Encyclopedia of Chess Variants*. Games & Puzzles, 1994.

[Reiter, 1991] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, 27:359–380, 1991.

[Ruan and Thielscher, 2011] Ji Ruan and Michael Thielscher. The epistemic logic behind the game description language. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 840–845, 2011.

[Ruan and Thielscher, 2012] Ji Ruan and Michael Thielscher. Strategic and epistemic reasoning for the game description language GDL-II. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*, pages 696–701, 2012.

[Schiffel and Thielscher, 2011] Stephan Schiffel and Michael Thielscher. Reasoning about general games described in GDL-II. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 846–851, 2011.

[Schiffel and Thielscher, 2014] Stephan Schiffel and Michael Thielscher. Representing and reasoning about the rules of general games with imperfect information. *Journal of Artificial Intelligence Research*, 49:171–206, 2014.

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

[Thielscher, 2010] Michael Thielscher. A general game description language for incomplete information games. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 994–999, 2010.

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

[Waugh *et al.*, 2009] Kevin Waugh, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein, and Michael H Bowling. A practical use of imperfect recall. In *Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation (SARA'09)*, pages 175–182, 2009.

[Zhang and Thielscher, 2015a] Dongmo Zhang and Michael Thielscher. A logic for reasoning about game strategies. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 1671–1677, 2015.

[Zhang and Thielscher, 2015b] Dongmo Zhang and Michael Thielscher. Representing and reasoning about game strategies. *Journal of Philosophical Logic*, 44(2):203–236, 2015.