Characterizing the Expressivity of Game Description Languages

Guifei Jiang[1], Laurent Perrussel[2], Dongmo Zhang[3], Heng Zhang[4], and Yuzhi Zhang[1]

[1] Nankai University, Tianjin, China
[2] University of Toulouse, Toulouse, France
[3] Western Sydney University, Penrith, Australia
[4] Tianjin University, Tianjin, China

Abstract. Bisimulations are a key notion to study the expressive power of a modal language. This paper studies the expressiveness of Game Description Language (GDL) and its epistemic extension EGDL through a bisimulations approach. We first define a notion of bisimulation for GDL and prove that it coincides with the indistinguishability of GDL-formulas. Based on it, we establish a characterization of the definability of GDL in terms of k-bisimulations. Then we define a novel notion of bisimulation for EGDL, and obtain a characterization of the expressive power of EGDL. In particular, we show that a special case of the bisimulation for EGDL can be used to characterize the expressivity of GDL. These characterizations not only justify the notions of bisimulation are appropriate for game description languages, but also provide a powerful tool to identify their expressive power.

Keywords: Bisimulation Equivalence· Expressive Power· Game Description Languages.

1 Introduction

General Game Playing (GGP) is concerned with creating intelligent agents that understand the rules of previously unknown games and learn to play these games without human intervention [8]. To represent the rules of arbitrary games, a formal game description language (GDL) was introduced as an official language for GGP in 2005. GDL is originally a machine-processable, logic programming language [14]. Most recently, it has been adapted as a minimal logical language for game specification and strategic reasoning [21]. The epistemic extension EGDL has been also developed to incorporate imperfect information games [12].

Although GDL and EGDL are logical languages for representing game rules and specifying game properties, their logical properties, especially their expressive power have not been fully investigated yet. For instance, which game properties are definable or non-definable in GDL and EGDL? How to show a game property is not definable in GDL and EGDL? When two game descriptions are equivalent? The existing work about the expressiveness of game description languages is rare, mostly investigating the relationships of these languages with other strategic logics. In particular, Ruan *et al.* study the relationship between GDL and Alternating-time Temporal Logic (ATL) by transferring a GDL game specification into an ATL specification [18]. Lorini and Schwarzentruber investigate the relation between GDL and Seeing-to-it-that Logics (STITs) by providing a polynomial embedding of GDL into STIT [13]. In this paper, we propose a different approach to address these questions via *bisimulation*.

The notion of *bisimulation* plays a pivotal role to identify the expressive power of a logic. It was independently defined and developed in the areas of theoretical computer science [11, 15] and the model theory of modal logic [2, 3]. Since bisimulation-equivalent structures can simulate each other in a stepwise manner, they cannot be distinguished by the concerned logic. An appropriate notion of bisimulation for a logic allows us to study the expressive power of that logic in terms of structural invariance and language indistinguishability [10].

On the basis of the above consideration, we use in this paper a bisimulation approach to investigate the expressive power of GDL and EGDL. To this end, we first define a notion of *bisimulation equivalence* for GDL and prove that *it preserves the invariance of GDL-formulas*. Based on this, we provide a characterization for the definability of GDL, and show that *a class of state transition models is definable in GDL iff they are closed under* k*-bisimulations*. This allows us to establish the non-definability of a property in GDL. For instance, we show that GDL does not allow to express the property that a player has a winning strategy. More importantly, to characterize the expressivity of EGDL, we define a novel notion of bisimulation, called (m, n)*-bisimulation*. We not only prove that (m, n)-bisimulation can be logically characterized by EGDL, but also establish a characterization of the definability of EGDL. These characterizations not only justify that the notions of bisimulation are appropriate for GDL and EGDL, but also provide a powerful tool to identify their expressive power. To the best of our knowledge, this work is the first to conduct a systematic study on the expressive power of Game Description Languages. This would help us to identify the roles of Game Description Languages compared to other existing strategic logics, such as ATL, STIT, and choose the right language for the intended application.

The rest of this paper is structured as follows: Section 2 introduces the framework of GDL. Section 3 defines the notion of bisimulation equivalence for GDL and characterizes its definability. Section 4 defines the notion of bisimulation for EGDL and characterizes its expressivity. Finally, we conclude with future work.

2 The GDL-Based Framework

Let us now introduce the GDL-based framework from [21]. Each game is associated with a *game signature*. A *game signature* $\mathcal{S}$ is a triple $(N, \mathcal{A}, \Phi)$, where

- $N = \{1, 2, \cdots, m\}$ is a non-empty finite set of agents,
- $\mathcal{A}$ is a non-empty finite set of *actions* such that it contains *noop*, an action without any effect, and

– $\Phi = \{p, q, \cdots\}$ is a finite set of propositional atoms for specifying individual features of a game state.

Through the rest of the paper, we will consider a fixed game signature $\mathcal{S}$, and all concepts are based on the game signature unless otherwise specified.

2.1 State Transition Models

The structure for modelling games is defined as follows:

Definition 1. *A* state transition (ST) model M *is a tuple* $(W, w_0, T, L, U, g, \pi)$, *where*

– W *is a non-empty finite set of* possible states.
– $w_0 \in W$*, representing the unique* initial *state.*
– $T \subseteq W$*, representing a set of* terminal *states.*
– $L \subseteq W \times N \times \mathcal{A}$ *is a* legality *relation, specifying legal actions for each agent at game states. Let* $L_r(w) = \{a \in \mathcal{A} : (w, r, a) \in L\}$ *be the set of all legal actions for agent* r *at state* w*. To make a game playable, it is assumed that (i)* $L_r(w) \neq \emptyset$ *for any* $r \in N$ *and* $w \in W$*, and (ii)* $L_r(w) = \{noop\}$ *for any* $r \in N$ *and* $w \in T$*.*
– $U : W \times \mathcal{A}^{|N|} \hookrightarrow W \backslash \{w_0\}$ *is a partial* update *function, specifying the state transition for each state and legal joint action, such that* $U(w, \langle noop^r \rangle_{r \in N}) = w$ *for any* $w \in W \setminus \{w_0\}$*.*
– $g : N \to 2^W$ *is a* goal *function, specifying the winning states of each agent.*
– $\pi : W \to 2^\Phi$ *is a standard valuation function.*

Note that to make the framework as general as possible, here we consider synchronous games and as demonstrated by Example 1, turn-based games involved in [21] are special cases by allowing a player only to do "noop" when it is not her turn. For convenience, let D denote the set of all joint actions $\mathcal{A}^{|N|}$. Given $d \in D$, we use $d(r)$ to specify the action taken by agent r.

The following notion specifies all possible ways in which a game can develop.

Definition 2. *Let* $M = (W, w_0, T, L, U, g, \pi)$ *be an ST-model. A path* δ *is an infinite sequence of states and joint actions* $w_0 \xrightarrow{d_1} w_1 \xrightarrow{d_2} \cdots \xrightarrow{d_j} \cdots$ *s.t. for any* $j \geq 1$ *and* $r \in N$*,*

1. $w_j \neq w_0$ *(that is, only the first state is initial.)*
2. $d_j(r) \in L_r(w_{j-1})$ *(that is, any action taken by each agent must be legal.)*
3. $w_j = U(w_{j-1}, d_j)$ *(state update)*
4. *if* $w_{j-1} \in T$*, then* $w_{j-1} = w_j$ *(self-loop after reaching a terminal state.)*

Let $\mathcal{P}(M)$ denote the set of all paths in M. For $\delta \in \mathcal{P}(M)$ and a stage $j \geq 0$, we use $\delta[j]$ to denote the j-th state on δ, and $\theta_r(\delta, j)$ the action taken by agent r at stage j of δ.

2.2 The Language

The language for game specification is given as follows:

Definition 3. *The language $\mathcal{L}_{GDL}$ for game description is generated by the following BNF:*

$$\varphi \ ::= \ p \mid initial \mid terminal \mid legal(r,a) \mid wins(r) \mid does(r,a) \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi$$

where $p \in \Phi$, $r \in N$ and $a \in \mathcal{A}$.

Other connectives $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ are defined by $\neg$ and $\wedge$ in the standard way. Intuitively, *initial* and *terminal* specify the initial state and the terminal states of a game, respectively; $does(r,a)$ asserts that agent r takes action a at the current state; $legal(r,a)$ asserts that agent r is allowed to take action a at the current state, and $wins(r)$ asserts that agent r wins at the current state. Finally, the formula $\bigcirc\varphi$ means that φ holds in the next state. We use the following abbreviations in the rest of paper. For $d = \langle a_r \rangle_{r\in N}$, $does(d) =_{def} \bigwedge_{r\in N} does(r,a_r)$, and $\bigcirc^k\varphi =_{def} \underbrace{\bigcirc\cdots\bigcirc}_{k}\varphi$.

Note that our language is slightly different from [21] by introducing the agent parameter in $legal(\cdot)$ and $does(\cdot)$. To help the reader capture the intuition of the language, let us consider the following example.

Example 1 (Tic-Tac-Toe). Two players take turns in marking either a cross 'x'or a nought 'o' on an 3×3 board. The player who first gets three consecutive marks of her own symbol in a row wins this game.

The game signature for Tic-Tac-Toe, written $\mathcal{S}_{TT}$, is given as follows: $N_{TT} = \{\mathsf{x},\mathsf{o}\}$ denoting the two game players; $\mathcal{A}_{TT} = \{a_{i,j} \mid 1 \leq i,j \leq 3\} \cup \{noop\}$, where $a_{i,j}$ denotes filling cell (i,j), and $\Phi_{TT} = \{p^r_{i,j}, turn(r) \mid r \in \{\mathsf{x},\mathsf{o}\}$ and $1 \leq i,j \leq 3\}$, where $p^r_{i,j}$ represents the fact that cell (i,j) is filled by player r. The rules of this game is given in Figure 4.

1. $initial \leftrightarrow turn(\mathsf{x}) \wedge \neg turn(\mathsf{o}) \wedge \bigwedge_{i,j=1}^{3} \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
2. $wins(r) \leftrightarrow \bigvee_{i=1}^{3}\bigwedge_{j=0}^{2} p^r_{i,1+j} \vee \bigvee_{j=1}^{3}\bigwedge_{i=0}^{2} p^r_{1+i,j} \vee \bigwedge_{i=0}^{2} p^r_{1+i,1+i} \vee \bigwedge_{i=0}^{2} p^r_{1+i,3-i}$
3. $teminal \leftrightarrow wins(\mathsf{x}) \vee wins(\mathsf{o}) \vee \bigwedge_{i,j=1}^{3} (p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j})$
4. $legal(r, a_{i,j}) \leftrightarrow \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}) \wedge turn(r) \wedge \neg terminal$
5. $legal(r, noop) \leftrightarrow turn(-r) \vee terminal$
6. $\bigcirc p^r_{i,j} \leftrightarrow p^r_{i,j} \vee (does(r, a_{i,j}) \wedge \neg(p^{\mathsf{x}}_{i,j} \vee p^{\mathsf{o}}_{i,j}))$
7. $turn(r) \wedge \neg terminal \rightarrow \bigcirc\neg turn(r) \wedge \bigcirc turn(-r)$

Fig. 1. A GDL description of Tic-Tac-Toe.

The initial state, each player's winning states, the terminal states and the turn-taking are given by formulas 1-3 and 7, respectively. The preconditions of each action (legality) are specified by Formula 4 and 5. Formula 6 is the combination of the frame axioms and the effect axioms [17].

2.3 The Semantics

The semantics of this language is specified as follows:

Definition 4. *Let $M = (W, w_0, T, L, U, g, \pi)$ be an ST-model. Given a path δ of M, a stage $j \in \mathbb{N}$ and a formula $\varphi \in \mathcal{L}_{GDL}$, we say φ is true (or satisfied) at j of δ under M, denoted $M, \delta, j \models \varphi$, according to the following definition*:

$$
\begin{array}{lll}
M, \delta, j \models p & \textit{iff} & p \in \pi(\delta[j]) \\
M, \delta, j \models \neg\varphi & \textit{iff} & M, \delta, j \not\models \varphi \\
M, \delta, j \models \varphi_1 \wedge \varphi_2 & \textit{iff} & M, \delta, j \models \varphi_1 \textit{ and } M, \delta, j \models \varphi_2 \\
M, \delta, j \models \mathit{initial} & \textit{iff} & \delta[j] = w_0 \textit{ and } j = 0 \\
M, \delta, j \models \mathit{terminal} & \textit{iff} & \delta[j] \in T \\
M, \delta, j \models \mathit{wins}(r) & \textit{iff} & \delta[j] \in g(r) \\
M, \delta, j \models \mathit{legal}(r, a) & \textit{iff} & a \in L_r(\delta[j]) \\
M, \delta, j \models \mathit{does}(r, a) & \textit{iff} & \theta_r(\delta, j) = a \\
M, \delta, j \models \bigcirc\varphi & \textit{iff} & M, \delta, j+1 \models \varphi
\end{array}
$$

An ST-model M *satisfies* a formula φ if there are a path $\delta \in \mathcal{P}(M)$ and a stage $j \in \mathbb{N}$ such that $M, \delta, j \models \varphi$. We say a formula φ is *satisfied at the initial state* w_0 in M, written $M, w_0 \models \varphi$, if $M, \delta, 0 \models \varphi$ for all $\delta \in \mathcal{P}(M)$.

3 Bisimulation and Definability of GDL

In this section, we first define the notion of bisimulation equivalence for GDL, and prove the invariance result of GDL-formulas. Then we present a characterization of the definability of GDL in terms of k-bisimulation.

3.1 Bisimulation and Invariance for GDL

We consider two types of bisimulation for ST-models. The first one is inspired by the notion of bisimulation in [5, 6] defined as follows:

Definition 5. *Let $M = (W, w_0, T, L, U, g, \pi)$ and $M' = (W', w'_0, T', L', U', g', \pi')$ be two ST-models (based on the same game signature). We say M and M' are bisimulation-equivalent (bisimilar, for short), written $M \simeq M'$, if there is a binary relation $Z \subseteq W \times W'$ s.t. $w_0 Z w'_0$, and for all states $w \in W$ and $w' \in W'$ with wZw', we have*

1. *All the following hold:*
   (a) $\pi(w) = \pi'(w')$;
   (b) $w = w_0$ *iff* $w' = w'_0$;
   (c) $w \in T$ *iff* $w' \in T'$;
   (d) $L_r(w) = L'_r(w')$ *for any* $r \in N$;
   (e) $w \in g(r)$ *iff* $w' \in g'(r)$ *for any* $r \in N$.
2. *For every $d \in D$ and $u \in W$, if $U(w, d) = u$, then there is $u' \in W'$ s.t. $U'(w', d) = u'$ and uZu';*

3. *For every* $d \in D$ *and* $u' \in W'$, *if* $U'(w', d) = u'$, *then there is* $u \in W$ *s.t.* $U(w, d) = u$ *and* uZu'.

When Z is a bisimulation linking two states w in M and w' in M', we say that w and w' are *bisimilar*, written $M, w \simeq M', w'$. In particular, if $M \simeq M'$, then their initial states are bisimilar, i.e., $M, w_0 \simeq M', w_0'$. In the following, for convenience we denote Condition (a)-(e) in Definition 5 as *the local properties* of a state.

With path-based semantics, we define the second type of bisimulation, called *path bisimulation* as follows:

Definition 6. *Consider two ST-models* $M = (W, w_0, T, L, U, g, \pi)$ *and* $M' = (W', w_0', T', L', U', g', \pi')$. *Let* $\delta \in \mathcal{P}(M)$ *and* $\delta' \in \mathcal{P}(M')$, *we say* δ *and* δ' *are bisimilar, written* $M, \delta \simeq M', \delta'$, *iff for every* $j \geq 0$ *and* $r \in N$, *the local properties hold for* $\delta[j]$ *and* $\delta'[j]$, *and* $\theta_r(\delta, j) = \theta_r(\delta', j)$.

This asserts that two paths are bisimilar if all the corresponding states satisfy the same local properties, and each agent takes the same action at every stage.

It turns out that with the deterministic property, the two types of bisimulation are equivalent. Formally, we have the following result.

Lemma 1. *Given two ST-models* M *and* M', $M \simeq M'$ *iff*

1. *for every* $\delta \in \mathcal{P}(M)$, *there is* $\delta' \in \mathcal{P}(M')$ *such that* $M, \delta \simeq M', \delta'$, *and*
2. *for every* $\delta' \in \mathcal{P}(M')$, *there is* $\delta \in \mathcal{P}(M)$ *such that* $M, \delta \simeq M', \delta'$.

That is, *M is bisimilar to M' iff each path that can be developed in one model can also be induced in the other.*

Let us now turn to the logical characterization of bisimulation equivalence. We first have the invariance result of GDL-formulas under path-bisimulation.

Proposition 1. *Let* M *and* M' *be two ST-models. For every* $\delta \in \mathcal{P}(M)$ *and* $\delta' \in \mathcal{P}(M')$, *the following are equivalent.*

1. $M, \delta \simeq M', \delta'$
2. $(M, \delta, j \models \varphi$ *iff* $M', \delta', j \models \varphi)$ *for any* $j \in \mathbb{N}$ *and* $\varphi \in \mathcal{L}_{GDL}$.

This result asserts that bisimulation equivalence and the invariance of GDL-formulas match on ST-models. On the one hand, this result justifies that the notion of bisimulation equivalence is natural and appropriate for GDL. On the other hand, two bisimilar ST-models cannot be distinguished by GDL language. This allows us to show the failure of bisimulation-equivalence easily. That is, *two ST-models are not bisimulation-equivalent if there is a GDL-formula that holds in one model and fails in the other.* For instance, let us consider the two ST-models depicted in **Fig. 2.** where $N = \{r\}$ and $\Phi = \emptyset$. Formula $initial \wedge \bigcirc^2 does(r, c)$ is satisfied in M, but unsatisfied in M'. This leads to $M \not\simeq M'$.

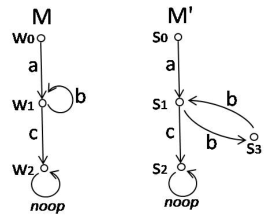

Fig. 2. M and M' are not bisimulation-equivalent.

3.2 *k*-Bisimulation and Definability of GDL

To show that a property of ST-models is definable in GDL, it suffices to find a defining formula. However, showing that a property is not definable in GDL is not so straightforward. It is well known that the expressive power of basic modal logic with respect to Kripke semantics can be completely characterized in terms of k-bisimulation [6]. In this section, we provide an analogous characterization result for GDL.

Here we consider the definability of properties that are satisfied at the initial state of an ST-model. For $\varphi \in \mathcal{L}_{GDL}$, let $\|\varphi\|$ be the set of all ST-models that satisfy φ at the initial state. i.e., $\|\varphi\| := \{M \mid M, w_0 \models \varphi\}$. The concept of the *definability* is specified as follows:

Definition 7. *A class $\mathcal{M}$ of ST-models is GDL-definable, if there is a formula $\varphi \in \mathcal{L}_{GDL}$ s.t. $\mathcal{M} = \|\varphi\|$.*

Similar to [6], we define the concept of *k-bisimulation* as follows:

Definition 8. *Let $M = (W, w_0, T, L, U, g, \pi)$ and $M' = (W', w'_0, T', L', U', g', \pi')$ be two ST-models. We say M and M' are k-bisimilar, written $M \simeq_k M'$, if there exists a sequence of binary relations $Z_k \subseteq Z_{k-1} \cdots \subseteq Z_0$ s.t. for any $w \in W$, $w' \in W'$ and $i \in \{0, \cdots, k-1\}$,*

1. *$w_0 Z_k w'_0$*
2. *If $w Z_0 w'$, then the local properties hold for w and w';*
3. *If $w Z_{i+1} w'$ and $U(w, d) = u$, then there is $u' \in W'$ s.t. $U'(w', d) = u'$ and $u Z_i u'$;*
4. *If $w Z_{i+1} w'$ and $U'(w', d) = u'$, then there is $u \in W$ s.t. $U(w, d) = u$ and $u Z_i u'$.*

The intuition is that if two ST-models are k-bisimilar, then their initial states w_0 and w'_0 bisimulate up to depth k. Clearly, if $M \simeq M'$, then $M \simeq_k M'$ for all $k \in \mathbb{N}$. We say a class $\mathcal{M}$ of ST-models is *closed under k-bisimulations* if for all ST-models M and M', if $M \in \mathcal{M}$ and $M \simeq_k M'$ then $M' \in \mathcal{M}$.

Before providing the characterization of the definability of GDL, we need some additional notions and results. The depth of next operators for a formula $\varphi \in \mathcal{L}_{GDL}$, written $deg_N(\varphi)$, is inductively defined as follows:

$$deg_N(\varphi) = \begin{cases} 0, & \text{for } \varphi \text{ is } \bigcirc\text{−free} \\ deg_N(\psi), & \text{for } \varphi = \neg\psi \\ Max\{deg_N(\varphi_1), deg_N(\varphi_2)\}, & \text{for } \varphi = \varphi_1 \wedge \varphi_2 \\ deg_N(\psi) + 1, & \text{for } \varphi = \bigcirc\psi \end{cases}$$

Definition 9. *Let M, M' be two ST-models and $k \in \mathbb{N}$. We say M and M' are k-equivalent, written $M \equiv_k M'$, if at the initial states, they satisfy the same GDL-formulas of degree at most k, i.e.,* $\{\varphi \in \mathcal{L}_{GDL} \mid deg(\varphi) \leq k \text{ and } M, w_0 \models \varphi\} = \{\psi \in \mathcal{L}_{GDL} \mid deg(\psi) \leq k \text{ and } M', w'_0 \models \psi\}$.

We use the fact that *for every ST-model M and every $k \in \mathbb{N}$ there is a formula that completely characterizes M up to k-equivalence.* With action operator and path-based semantics, the way to construct the k-th characteristic formula of an ST-model is non-standard. We need take the following steps.

1. Redefine the set of *atomic propositions*, written Atm, as follows: $Atm = \Phi \cup \{initial, terminal\} \cup \{wins(r), legal(r, a) \mid r \in N, a \in \mathcal{A}\}$.
2. Encode the atomic propositions through a valuation V rather than through separate relations or functions. For every $w \in W$, let $V(w) = \{p \in \Phi \mid p \in \pi(w)\} \cup \{initial \mid w = w_0\} \cup \{terminal \mid w \in T\} \cup \{wins(r) \mid w \in g(r)\} \cup \{legal(r, a) \mid a \in L_r(w)\}$. Note $V(w)$ is finite since N, $\mathcal{A}$ and Φ are all finite.
3. For each path $\delta := w_0 \stackrel{d_1}{\to} w_1 \stackrel{d_2}{\to} \cdots \stackrel{d_j}{\to} \cdots$ in M, induce a *trace* $V(\delta) = V(w_0) \cdot does(d_1) \cdot V(w_1) \cdots does(d_j) \cdot V(w_j) \cdots$. Let φ^k_δ be the syntactical representation of δ up to depth k, i.e., $\varphi^k_\delta := (\bigwedge V(\delta[0]) \wedge does(d_1)) \wedge \bigcirc(\bigwedge V(\delta[1]) \wedge does(d_2)) \wedge \cdots \wedge \bigcirc^k(\bigwedge V(\delta[k]) \wedge does(d_{k+1}))$.
4. Define the *k-th characteristic formula* Γ^k_M of M as the disjunctions of all the syntactical representations of paths in M up to depth k, i.e.,

$$\Gamma^k_M := \bigvee_{\delta \in \mathcal{P}(M)} \varphi^k_\delta.$$

Note that Γ^k_M is well-formed as M is finite-branching and all paths are bounded to depth k. It is easy to check that $deg(\Gamma^k_M) = k$ and $M, w_0 \models \Gamma^k_M$.

To illustrate this idea, let us consider the ST-model M depicted in **Fig. 3.**, where $N = \{r\}$, $\Phi = \emptyset$, $T = \{w_{22}, w_{23}\}$ and $g(r) = \{w_{23}\}$. Then the 2-th characteristic formula of M is $\Gamma^2_M = \varphi^2_{\delta_1} \vee \varphi^2_{\delta_2} \vee \varphi^2_{\delta_3}$, where

$\varphi^2_{\delta_1} = initial \wedge \bigwedge^2_{i=1} legal(r, a_i) \wedge does(r, a_1) \wedge \bigcirc(\bigwedge^2_{i=1} legal(r, b_i) \wedge does(r, b_1)) \wedge \bigcirc^2(legal(r, c) \wedge does(r, c))$,

$\varphi^2_{\delta_2} = initial \wedge \bigwedge^2_{i=1} legal(r, a_i) \wedge does(r, a_1) \wedge \bigcirc(\bigwedge^2_{i=1} legal(r, b_i) \wedge does(r, b_2)) \wedge \bigcirc^2(terminal \wedge legal\ (r, noop) \wedge does(r, noop))$, and

$\varphi^2_{\delta_3} = initial \wedge \bigwedge^2_{i=1} legal(r, a_i) \wedge does(r, a_2) \wedge \bigcirc(legal(r, b_3) \wedge does(r, b_3)) \wedge \bigcirc^2\ (wins(r) \wedge terminal \wedge\ legal(r, noop) \wedge does(r, noop))$.

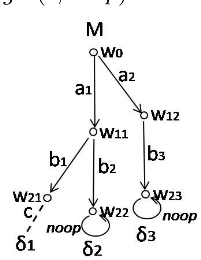

Fig. 3. Characteristic Formula Γ^2_M.

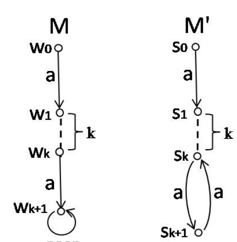

Fig. 4. A non-definable property.

The following lemma shows that the characteristic formula Γ^k_M captures the essence of k-bisimulation.

Lemma 2. *Let M, M' be two ST-models, and $k \in \mathbb{N}$. The following are equivalent.*

1. $M \simeq_k M'$
2. $M \equiv_k M'$
3. $M', w'_0 \models \Gamma^k_M$

This result asserts that (i) k-bisimulation coincides with k-equivalence on ST-models, and (ii) two ST-models are k-bisimilar if and only if for any path developed in one model, its k-th prefix can also be developed in the other.

We are now in the position to provide a characterization for the definability of GDL with respect to k-bisimulation.

Theorem 1. *A class $\mathcal{M}$ of ST-models is GDL-definable iff there is $k \in \mathbb{N}$ s.t. $\mathcal{M}$ is closed under k-bisimulations.*

This theorem indicates that *exactly the properties of ST-models that are closed under k-bisimulation for some $k \in \mathbb{N}$ are definable in GDL.* This provides a feasible approach to test the non-definability of GDL. We can show, for instance, that GDL can express that a player r will win in i steps, i.e., $\bigcirc^i wins(r)$, but it cannot express that *a player has a winning strategy* in general. Indeed for an arbitrary $k \in \mathbb{N}$, we can always construct two ST-models depicted in **Fig. 4.**, where $N = \{r\}$, $\Phi = \emptyset$, $w_{k+1} \in g(r)$ and $s_i \notin g'(r)$ for all $i \in \{0, \cdots, k+1\}$. It is easy to check that $M \simeq_k M'$, but player r has a winning strategy in M while she does not have in M'. By a slight change of M and M' with $w_{k+1} \in T$ and $s_{k+1} \notin T'$, we obtain another GDL-undefinable property that *a game will always reach a terminal state.* It is worth noting that GDL is a lightweight language for describing game rules and specifying game properties, compared to other strategic logics such as ATL [1] and Strategy Logic [7] which can express those properties. This is the price paid for the low complexity of GDL.

4 Bisimulation and Definability of EGDL

In this section, we first define a notion of bisimulation especially designed for EGDL, and then prove that it can be logically characterized by EGDL. We finally provide a characterization of the definability of EGDL.

Let us first introduce the language and semantics of EGDL in [12]. The language of EGDL is obtained by extending GDL with the standard epistemic operators. A formula $\varphi \in \mathcal{L}_{EGDL}$ is defined by the following BNF:

$$\varphi \ ::= \ p \mid initial \mid terminal \mid legal(r,a) \mid wins(r) \mid$$

$$does(r,a) \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi \mid \mathsf{K}_r\varphi \mid \mathsf{C}\varphi$$

where $p \in \Phi$, $r \in N$ and $a \in \mathcal{A}$.

Besides the GDL-components, the formula $\mathsf{K}_r\varphi$ is read as "agent r knows φ", and $\mathsf{C}\varphi$ as "φ is common knowledge among all the agents in N".

The semantics of EGDL is based on epistemic state transition models. An *epistemic state transition (ET)* model is obtained by associating the state transition model with an equivalence relation $R_r \subseteq W \times W$ for each agent r, indicating the states that are indistinguishable for r. To interpret epistemic formulas, the equivalence relation over states is generalized to paths: *two paths δ, $\lambda \in \mathcal{P}(M)$ are imperfect recall equivalent for agent r at stage $j \in \mathbb{N}$, written $\delta \approx_r^j \lambda$, iff $\delta[j] R_r \lambda[j]$.*

The semantics of EGDL is obtained by adding the following interpretation clauses to Definition 4:

$$M, \delta, j \models \mathsf{K}_r\varphi \text{ iff for all } \lambda \approx_r^j \delta,\ M, \lambda, j \models \varphi$$
$$M, \delta, j \models \mathsf{C}\varphi \text{ iff for all } \lambda \approx_N^j \delta,\ M, \lambda, j \models \varphi$$

where $\approx_N^j$ is the transitive closure of $\bigcup_{r \in N} \approx_r^j$.

4.1 Bisimulation Equivalence for ET-models

Different from ST-models, there are two dimensions in ET-models: the temporal and the epistemic dimension. The epistemic relation is actually determined by the stage-path pair, since path equivalence requires not only the corresponding states are undistinguishable, but also the states are reached at the same stage. The state-based bisimulation for GDL fails to capture the latter. Based on above analysis, we define a new notion of path-based bisimulation between ET-models.

Definition 10. *Let $M = (W, w_0, T, \{R_r\}_{r \in N}, L, U, g, \pi)$, $M' = (W', w_0', T', \{R_r'\}_{r \in N}, L', U', g', \pi')$ be two ET-models, $\delta \in \mathcal{P}(M)$, $\delta' \in \mathcal{P}(M')$ and $j \in \mathbb{N}$. We say M, δ, j and M', δ', j are (m,n)-bisimilar, written $M, \delta, j \leftrightarrows_m^n M', \delta', j$, if we have*

1. *The base case*
   (a) *the local properties hold for $\delta[j]$ and $\delta'[j]$.*
   (b) *$\theta_r(\delta, j) = \theta_r(\delta', j)$ for any $r \in N$.*
2. *If $m > 0$, $M, \delta, j+1 \leftrightarrows_{m-1}^n M', \delta', j+1$.*
3. *If $n > 0$, then for all $o \in N \cup \{N\}$,*
   (a) *for any $\lambda \in \mathcal{P}(M)$, $\delta \approx_o^j \lambda$, then there is $\lambda' \in \mathcal{P}(M')$ s.t. $\delta' \approx_o^j \lambda'$ and $M, \lambda, j \leftrightarrows_m^{n-1} M', \lambda', j$;*
   (b) *for any $\lambda' \in \mathcal{P}(M')$, $\delta' \approx_o^j \lambda'$, then there is $\lambda \in \mathcal{P}(M)$ s.t. $\delta \approx_o^j \lambda$ and $M, \lambda, j \leftrightarrows_m^{n-1} M', \lambda', j$.*

This recursively asserts that two paths from ET-models are (m,n)-bisimilar iff (i) at the current stage, they can not be distinguished, i.e., the same local properties hold and each agent takes the same action (Condition 1), and (ii) at the next stage, they also bisimulate each other from both the temporal and epistemic perspectives. This is specified by Condition (2) and Condition (3). Specifically, Condition (2) takes care of the temporal dimension: the depth m

of path bisimulation is reduced stage by stage until 0, and Condition (3) deals with the epistemic dimension: for each path indistinguishable from δ, there is a path indistinguishable from δ' that bisimulates it. The number of such bisimilar pairs at stage j is specified by the parameter n.

With this, we define the concept of (m, n)-bisimulation over ET-models as follows:

Definition 11. *Let M and M' be two ET-models. We say M is globally (m, n)-similar to M', written $M \propto_m^n M'$, if for every $\delta \in \mathcal{P}(M)$, there is $\delta' \in \mathcal{P}(M')$ such that $M, \delta, 0 \leftrightarrows_m^n M', \delta', 0$.*

We say M and M' are (m, n)-bisimilar, written $M \leftrightarrows_m^n M'$, if $M \propto_m^n M'$ and $M' \propto_m^n M$.

This asserts that two ET-models are (m, n)-bisimilar iff for every path in one model there is a path in the other such that their initial states are (m, n)-bisimilar. In particular, we say two ET-models M, M' are *path-based bisimilar*, written $M \leftrightarrows M'$, if $M \leftrightarrows_m^n M'$ for all $m, n \in \mathbb{N}$. Finally, a class $\mathcal{M}$ of ET-models is *closed under global (m, n)-simulations* if for all ET-models M and M', if $M \propto_m^n M'$ and $M' \in \mathcal{M}$ then $M \in \mathcal{M}$.

To obtain the characterization results, we need some additional notions. For any EGDL-formula $\varphi \in \mathcal{L}_{EGDL}$, the depth of next operators, written $deg_N(\varphi)$, is defined as for GDL-formulas except for $\varphi \in \{\mathsf{K}_r\psi, \mathsf{C}\psi\}$, $deg_N(\varphi) = deg_N(\psi)$. The depth of epistemic operators, written $deg_E(\varphi)$, is inductively defined as follows:

$$deg_E(\varphi) = \begin{cases} 0, & \text{for } \varphi \text{ is } \mathsf{K}_r, \mathsf{C}-\text{free} \\ deg_E(\psi), & \text{for } \varphi \in \{\neg\psi, \bigcirc\psi\} \\ Max\{deg_E(\varphi_1), deg_E(\varphi_2)\}, & \text{for } \varphi = \varphi_1 \wedge \varphi_2 \\ deg_E(\psi) + 1, & \text{for } \varphi \in \{\mathsf{K}_r\psi, \mathsf{C}\psi\} \end{cases}$$

Let EGDL(m, n) denote the set of all formulas with the depth of next operators and epistemic operators at most m, n, respectively, i.e, EGDL(m, n)=$\{\varphi \in \mathcal{L}_{EGDL} \mid deg_N(\varphi) \leq m$ and $deg_E(\varphi) \leq n\}$. Then we have the following logical characterization result for (m, n)-bisimilar paths.

Proposition 2. *Let M, M' be two ET-models. For every $\delta \in \mathcal{P}(M)$, $\delta' \in \mathcal{P}(M')$ and $j \in \mathbb{N}$, the following are equivalent.*

1. *$M, \delta, j \leftrightarrows_m^n M', \delta', j$*
2. *$(M, \delta, j \models \varphi$ iff $M', \delta', j \models \varphi)$ for all $\varphi \in$EGDL(m, n)*

This result asserts that (m, n)-bisimulation coincides with the indistinguishability of EGDL-formulas for paths. In particular, this also holds for the initial states.

Similarly, we say two ET-models M and M' are (m, n)-*equivalent*, written $M \equiv_m^n M'$, if at the initial states, they satisfies the same EGDL(m, n)-formulas, i.e., $\{\varphi \in \mathcal{L}_{EGDL(m,n)} \mid M, w_0 \models \varphi\} = \{\psi \in \mathcal{L}_{EGDL(m,n)} \mid M', w_0' \models \psi\}$. Then the following shows that (m, n)-bisimulation is logically characterized by EGDL(m, n).

Theorem 2. *Let M and M' be two ET-models. Then $M \leftrightarroweq^n_m M'$ iff $M \equiv^n_m M'$.*

This asserts (m, n)-bisimulation equivalence and the invariance of EGDL(m, n)-formulas coincides over ET-models. This result not only justifies that the notion of (m, n)-bisimulation is appropriate for EGDL, but also provides a feasible way to verify the failure of (m, n)-bisimulation. For instance, consider two ET-models M_1, M_2 depicted in **Fig. 5.**, where $N = \{r\}$, $\Phi = \emptyset$. The dotted line denotes the indistinguishability relation of agent r. Notice that the reflexive loops are omitted. Formula $\bigcirc \mathsf{K}_r(does(r, b) \to \neg \bigcirc^2 terminal)$ is not satisfied at w_0 of M_1, but holds at s_0 of M_2. Then $M_1 \not\equiv^1_3 M_2$. This leads to $M_1 \not\leftrightarroweq^1_3 M_2$.

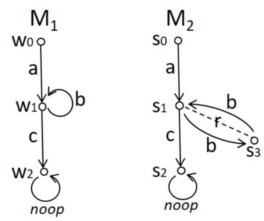

Fig. 5. M_1 and M_2 are not $(3, 1)-$bisimilar.

In particular, we have the following result about the characterization of bisimilar ET-models in terms of EGDL. Recall that $M \leftrightarroweq M'$ if $M \leftrightarroweq^n_m M'$ for all $m, n \in \mathbb{N}$.

Proposition 3. *Let M and M' be two ET-models. Then $M \leftrightarroweq M'$ iff they satisfy the same EGDL-formulas.*

4.2 Logical Characterization of Definability of EGDL

Let us present the characterization of the definability of EGDL in terms of global (m, n)-simulations. Note that the notion of the *definability* of EGDL is defined the same as GDL in Definition 7.

Theorem 3. *A class $\mathcal{M}$ of ET-models is EGDL-definable iff there are $m, n \in \mathbb{N}$ such that $\mathcal{M}$ is closed under global (m, n)-simulations.*

This theorem asserts that *exactly the properties of ET-models that are closed under global (m, n)-simulations for some $m, n \in \mathbb{N}$ are definable in EGDL*. Similar to Theorem 1, this provides a feasible way to verify the non-definability of EGDL. For instance, we can show that EGDL cannot express that a player knows that she has a winning strategy.

We end this section with the following results that the expressivity of GDL is characterized by a special case of (m, n)-bisimulation with $n = 0$.

Proposition 4. *Let M and M' be two ET-models. Then the following are equivalent.*

1. *$M \leftrightarroweq^0_m M'$ for any $m \in \mathbb{N}$*
2. *they satisfy the same GDL formulas.*

Proposition 5. *A class $\mathcal{M}$ of ET-models is GDL-definable iff there is $m \in \mathbb{N}$ s.t. $\mathcal{M}$ is closed under global $(m, 0)$-simulations.*

These results indicates that path-based bisimulation provide a different way to characterize the expressivity of GDL. The first result shows that $(m, 0)$-bisimulation and the invariance of GDL-formulas match over ET-models, and the last characterizes the definability of GDL under $(m, 0)$-bisimulations. This indicates that without considering bisimulation for epistemic relations, $(m, 0)$-bisimulation actually boils down to m-bisimulation for ST-models.

5 Conclusion

In this paper, we have used a bisimulation approach to investigate the expressive power of GDL and EGDL. Specifically, we have defined notions of bisimulations for GDL and EGDL, and obtained the logical characterizations, respectively. We have also shown that a special case of path-based bisimulation can be used to characterize the expressivity of GDL. These results provide a feasible tool to identify the expressive power of game description languages. Finally, it is worth mentioning that bisimulation is a generic approach to identify the expressivity of a logic. Yet it is also sensitive to the logic it applies. Special techniques have to be developed for specific logics. With action operator and path-based semantics, the notions of bisimulation for GDL and EGDL are actually not a trivial and standard generalization of that for modal logic.

Directions of future research are manifold. We intend to explore the van Benthem Characterization Theorem for GDL and EGDL [3]. More recently, GDL has been extended to GDL-II and GDL-III for representing and reasoning about imperfect information games in GGP [19, 20]. We plan to study the expressivity of these languages and compare them with EGDL. Last but not least, bisimulation equivalence provides a natural yet overly strict criterion on game equivalence. It would be interesting to investigate different types of game equivalence in GGP [4, 9, 16, 22].

Acknowledgments. Guifei Jiang acknowledges the support of the National Natural Science Foundation of China (NO.61806102), the Fundamental Research Funds for the Central Universities, and the Major Program of the National Social Science Foundation of China (NO.17ZDA026). Laurent Perrussel acknowledges the support of the ANR project AGAPE ANR-18-CE23-0013.

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM **49**(5), 672–713 (2002)
2. van Benthem, J.: Modal Correspondence Theory. Ph.D. thesis, University of Amsterdam (1977)
3. van Benthem, J.: Correspondence theory. In: Handbook of philosophical logic, pp. 167–247. Springer (1984)

4. van Benthem, J., Bezhanishvili, N., Enqvist, S.: A new game equivalence and its modal logic. In: Proceedings Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK'17). pp. 57–74 (2017)
5. Blackburn, P., De Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2002)
6. Blackburn, P., van Benthem, J., Wolter, F.: Handbook of modal logic, vol. 3. Elsevier (2006)
7. Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. Information and Computation **208**(6), 677–693 (2010)
8. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. AI magazine **26**(2), 62–72 (2005)
9. Goranko, V.: The basic algebra of game equivalences. Studia Logica **75**(2), 221–238 (2003)
10. Grädel, E., Otto, M.: The freedoms of (guarded) bisimulation. In: Johan van Benthem on Logic and Information Dynamics, pp. 3–31. Springer (2014)
11. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. Journal of the ACM (JACM) **32**(1), 137–161 (1985)
12. Jiang, G., Zhang, D., Perrussel, L., Zhang, H.: Epistemic GDL: A logic for representing and reasoning about imperfect information games. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16). pp. 1138–1144 (2016)
13. Lorini, E., Schwarzentruber, F.: A path in the jungle of logics for multi-agent system: On the relation between general game-playing logics and seeing-to-it-that logics. In: Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'17). pp. 687–695 (2017)
14. Love, N., Hinrichs, T., Haley, D., Schkufza, E., Genesereth, M.: 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)
15. Park, D.: Concurrency and automata on infinite sequences. In: Theoretical computer science, pp. 167–183. Springer (1981)
16. Pauly, M.: Logic for Social Software. PhD thesis, University of Amsterdam (2001), `ILLCDissertationSeries2001-10`
17. Reiter, R.: 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)
18. Ruan, J., van Der Hoek, W., Wooldridge, M.: Verification of games in the game description language. Journal of Logic and Computation **19**(6), 1127–1156 (2009)
19. Thielscher, M.: A general game description language for incomplete information games. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10). pp. 994–999 (2010)
20. Thielscher, M.: GDL-III: A description language for epistemic general game playing. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17). pp. 1276–1282 (2017)
21. Zhang, D., Thielscher, M.: Representing and reasoning about game strategies. Journal of Philosophical Logic **44**(2), 203–236 (2015)
22. Zhang, H., Liu, D., Li, W.: Space-consistent game equivalence detection in general game playing. In: Workshop on Computer Games. pp. 165–177 (2015)