Game Equivalence and Bisimulation for Game Description Language

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. This paper investigates the equivalence between games represented by state transition models and its applications. We first define a notion of bisimulation equivalence between state transition models and prove that it can be logically characterized by Game Description Language (GDL). Then we introduce a concept of quotient state transition model. As the minimum equivalent of the original model, it allows us to improve the efficiency of model checking for GDL. Finally, we demonstrate with real games that bisimulation equivalence can be generalized to characterize more general game equivalence.

Keywords: Game Equivalence · Bisimulation Equivalence · General Game Playing.

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 [9]. 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 logical language for game specification and strategic reasoning [24]. Moreover, its epistemic and dynamic extensions have also been developed [13, 23].

As a logical language for representing game rules and specifying game properties, the logical properties, especially the expressive power of GDL have not been fully investigated yet. For instance, which game properties are definable in GDL? When two GDL game descriptions are equivalent? How to distinguish two GDL-defined games? In this paper, we will address these questions through a *bisimulation* approach.

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 [12, 16] and the model theory of modal logic [3, 4].

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 [11].

Besides identifying the expressivity of a logic, bisimulation equivalence also allows us to obtain the minimum equivalent of the original model, called *the quotient model*, which can be used to improve the efficiency of model checking [2]. Moreover, in terms of GDL, bisimulation equivalence tells us when two game structures are essentially the *same*, and thus gives us a natural criterion on the equivalence between games. Exploiting game equivalence may provide a bridge for knowledge transfer between a new game and a well-studied game in GGP. In particular, Zhang *et al.* considered that two games are equivalent exactly if the state machines described them are identical (isomorphic) [25]. Such definition might be too strong, as it would rule out many non-identical but essentially equivalent games, such as bisimulation-equivalent games.

Based on the above considerations, we will use in this paper a concept of bisimulation as a tool to investigate the expressive power of GDL and capture the equivalence between games represented by state transition models. We first define a concept of bisimulation equivalence between state transition models and prove that it coincides with the invariance of GDL-formulas on state transition models. This justifies that the notion of bisimulation equivalence is appropriate for GDL. Then we introduce a concept of quotient state transition model and show that it is bisimulation-equivalent to its original model. Considering its smaller size, this provides a way to improve the efficiency of model checking. Finally we demonstrate with real games that bisimulation equivalence can be generalized to capture a wider range of game equivalence.

The rest of this paper is structured as follows: Section 2 introduces the framework for game description. Section 3 defines the concept of bisimulation equivalence and introduces the notion of quotient model. Section 4 generalizes bisimulation equivalence to characterize more general game equivalence. Finally, we conclude with related work and future work.

2 The Framework

Let us now introduce the GDL-based framework from [24]. All games are assumed to be played in multi-agent environments. 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

This paper focuses on synchronous games where all players move simultaneously. These games can be specified by *state transition models* 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, we assume that (i) each agent has at least one available action at each state, i.e.,* $L_r(w) \neq \emptyset$ *for any* $r \in N$ *and* $w \in W$, *and (ii) each agent can only do action noop at terminal states, i.e.,* $L_r(w) = \{noop\}$ *for any* $r \in N$ *and* $w \in T$.
- $U : W \times \mathcal{A}^{|N|} \to W \backslash \{w_0\}$ *is an* 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* $w \in W \backslash \{w_0\}$.
- $g : N \to 2^W$ *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, we use the concurrent game structure and the turn-based game structure involved in [24] is a special case 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* $w_0 \xrightarrow{d_1} w_1 \xrightarrow{d_2} \cdots \xrightarrow{d_j} \cdots$ *such that 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 that is 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 of δ and $\theta_r(\delta, j)$ to denote the action taken by agent r at stage j of δ.

2.2 The Language

Let us now introduce the GDL-based language from [24] for game specification.

Definition 3. *The language $\mathcal{L}$ 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 [24] 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 (Number Scrabble). Two players take turns to select numbers from 1 to 9 without repeating any numbers previously used. The first player who selects three numbers that add up to 15 wins.

The game signature $\mathcal{S}_{NS}$ is given as follows: $N_{NS} = \{\mathsf{b}, \mathsf{w}\}$ denoting two game players; $\mathcal{A}_{NS} = \{\alpha(n) \mid 1 \leq n \leq 9\} \cup \{noop\}$, where $\alpha(n)$ denotes selecting number n, and $\Phi_{NS} = \{s(r,n), turn(r) \mid r \in \{\mathsf{b}, \mathsf{w}\}$ and $1 \leq n \leq 9\}$, where $s(r,n)$ represents the fact that number n is selected by player r, and $turn(r)$ says that player r has the turn now. The rules of Number Scrabble can be naturally formulated by GDL-formulas as shown in Figure 1 (where $r \in \{\mathsf{b}, \mathsf{w}\}$ and $-r$ represents r's opponent).

1. $initial \leftrightarrow turn(\mathsf{b}) \wedge \neg turn(\mathsf{w}) \wedge \bigwedge_{i=1}^{9} \neg(s(\mathsf{b},i) \vee s(\mathsf{w},i))$
2. $wins(r) \leftrightarrow \big(\bigvee_{i=2}^{3}(s(r,i) \wedge s(r,4) \wedge s(r,11-i)) \vee \bigvee_{i=1}^{2}(s(r,i) \wedge s(r,6) \wedge s(r,9-i)) \vee \bigvee_{l=1}^{4}(s(r,5-l) \wedge s(r,5) \wedge s(r,5+l))\big)$
3. $teminal \leftrightarrow wins(\mathsf{b}) \vee wins(\mathsf{w}) \vee \bigwedge_{i=1}^{9} (s(\mathsf{b},i) \vee s(\mathsf{w},i))$
4. $legal(r,\alpha(n)) \leftrightarrow \neg(s(\mathsf{b},n) \vee s(\mathsf{w},n)) \wedge turn(r) \wedge \neg terminal$
5. $legal(r,noop) \leftrightarrow turn(-r) \vee terminal$
6. $\bigcirc s(r,n) \leftrightarrow s(r,n) \vee (\neg(s(\mathsf{b},n) \vee s(\mathsf{w},n)) \wedge does(r,\alpha(n)))$
7. $turn(r) \wedge \neg terminal \rightarrow \bigcirc\neg turn(r) \wedge \bigcirc turn(-r)$

Fig. 1. A GDL description of Number Scrabble.

The formulas are intuitive. Formula 1 says at the initial state, player b has the first turn and all numbers are not selected. The next two formulas specify winning states of each player and the terminal states, respectively. The preconditions of each action (legality) are specified by Formula 4 and Formula 5. Formula 6 is the combination of the frame axioms and the effect axioms [19]. The last formula specifies the turn-taking.

2.3 The Semantics

The semantics of this language is based on ST-models with respect to a path and a stage of the path.

Definition 4. *Let $M = (W, w_0, T, L, U, g, \pi)$ be an ST-model. Given a path δ of M, a stage $j \geq 0$ and a formula $\varphi \in \mathcal{L}$, we say φ is true (or satisfied) at j of δ under M, denoted $M, \delta, j \models \varphi$, according to the following definition*:

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

A formula φ is *valid* in an ST-model M, written $M \models \varphi$, if $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}(M)$ and $j \geq 0$. A formula φ is called *satisfied at a state* w in M, written $M, w \models \varphi$, if it is true for all paths going through w, i.e., $M, \delta, j \models \varphi$ for any $\delta \in \mathcal{P}(M)$ and any $j \geq 0$ with $\delta[j] = w$. It follows that $M, w_0 \models \varphi$ iff $M, \delta, 0 \models \varphi$ for all $\delta \in \mathcal{P}(M)$.

3 Bisimulation Equivalence

In this section, we define the concept of bisimulation equivalence over state transition models and show it coincides with the invariance of GDL-formulas. We also introduce the quotient state transition model in terms of such relation.

3.1 Bisimulation and Invariance

Inspired by the notion of bisimulation in [7], we define the concept of bisimulation equivalence between ST-models 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. We say M and M' are bisimulation-equivalent, (bisimilar, for short), written $M \approx M'$, if there is a binary relation $Z \subseteq W \times W'$ such that $w_0 Z w'_0$, and for all states $w \in W$ and $w' \in W'$ with wZw', the following conditions hold:*

1. $\pi(w) = \pi'(w')$;

2. $w = w_0$ *iff* $w' = w'_0$*;*
3. $w \in T$ *iff* $w' \in T'$*;*
4. $a \in L_r(w)$ *iff* $a \in L'_r(w')$ *for any* $r \in N$ *and* $a \in \mathcal{A}$*;*
5. $w \in g(r)$ *iff* $w' \in g'(r)$ *for any* $r \in N$*;*
6. If $U(w, d) = u$*, then there is* $u' \in W'$ *s.t.* $U'(w', d) = u'$ *and* uZu'*;*
7. If $U'(w', d) = u'$*, then there is* $u \in W$ *s.t.* $U(w, d) = u$ *and* uZu'*.*

Note that $\approx$ is an equivalence relation over ST-models. When Z is a bisimulation linking two states w in M and w' in M', we say that w and w' are *bisimilar*, written $M, w \Leftrightarrow M', w'$. In particular, if $M \approx M'$, then their initial states are bisimilar, i.e., $M, w_0 \Leftrightarrow M', w'_0$.

Another way to understand bisimulation equivalence is to observe that M *is bisimilar to* M' *iff each path that can be developed in one model can also be induced in the other.* To formalize this idea, we need generalize the notion of bisimilar over states to paths as follows:

Definition 6. *Consider two ST-models* M *and* M'*. Given two paths* $\delta := w_0 \xrightarrow{d_1} w_1 \xrightarrow{d_2} \cdots$ *in* M *and* $\delta' := w'_0 \xrightarrow{d'_1} w'_1 \xrightarrow{d'_2} \cdots$ *in* M'*, we say* δ *and* δ' *are bisimilar, written* $M, \delta \Leftrightarrow M', \delta'$*, iff for every* $j \geq 0$ *and* $r \in N$*,* $M, \delta[j] \Leftrightarrow M', \delta'[j]$ *and* $\theta_r(\delta, j) = \theta_r(\delta', j)$*.*

That is, two paths are bisimilar if (i) all the corresponding states are bisimilar, and (ii) each agent takes the same action at every stage. With this, the above idea is restated as follows:

Lemma 1. *Given two ST-models* M *and* M'*,* $M \approx M'$ *iff for every* $\delta \in \mathcal{P}(M)$*, there is* $\delta' \in \mathcal{P}(M')$ *such that* $M, \delta \Leftrightarrow M', \delta'$*, and vice versa.*

Proof. ($\Rightarrow$) This direction holds directly by Condition 6 & 7 of Def 5.

($\Leftarrow$) Let $Z = \{(w, w') \mid$ there are $\delta \in \mathcal{P}(M), \delta' \in \mathcal{P}(M')$ and $j \geq 0$ such that $\delta[j] = w, \delta'[j] = w'$, the local properties Condition 1-5 in Def 5 hold for $\delta[j]$ and $\delta'[j]$, and $\theta_r(\delta, j) = \theta_r(\delta', j)\}$. Such a relation Z exists due to the assumption. It is easy to show that Z is a bisimulation between M and M'. ($\Box$)

Let us now turn to the logical characterization of bisimulation equivalence. We begin with the invariance of GDL-formulas under path-bisimulation.

Proposition 1. *Let* M*,* M' *be two ST-models. For every* $\delta \in \mathcal{P}(M)$ *and* $\delta' \in \mathcal{P}(M')$*, if* $M, \delta \Leftrightarrow M', \delta'$*, then* ($M, \delta, j \models \varphi$ *iff* $M', \delta', j \models \varphi$) *for any* $j \geq 0$ *and* $\varphi \in \mathcal{L}$*.*

It is routine to prove this by induction on φ. This result asserts that two bisimilar paths preserve GDL-formulas at each stage. Note that the other direction does not hold. Here is a simple counter-example. Let M and M' be two ST-model depicted in Figure 2, where $N = \{r\}$ and $\Phi = \emptyset$. Now consider two paths $\delta = w_0 \xrightarrow{a} w_1 \xrightarrow{b} \cdots$ in M and $\delta' = w'_0 \xrightarrow{a} w'_1 \xrightarrow{b} \cdots$ in M'. As $w_3 \notin T$ and $w'_3 \in T'$, then $M, w_3 \not\Leftrightarrow M', w'_3$, i.e., the successors of w_1 and w'_1 are not bisimilar, so $M, w_1 \not\Leftrightarrow M', w'_1$. Thus, δ and δ' are not bisimilar, i.e., $M, \delta \not\Leftrightarrow M', \delta'$. But it is easy to check that at each stage, δ and δ' satisfy the same GDL-formulas.

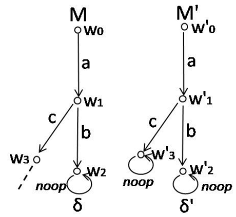

Fig. 2. δ and δ' are not bisimilar.

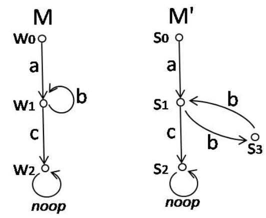

Fig. 3. M and M' are not bisimulation-equivalent.

To prove the characterization result, we need one additional notion. For each path $\delta := w_0 \stackrel{d_1}{\to} w_1 \stackrel{d_2}{\to} \cdots \stackrel{d_j}{\to} \cdots$ in M, we induce a *trace* $V(\delta) = V(w_0) \cdot does(d_1) \cdot V(w_1) \cdots does(d_{j-1}) \cdot V(w_j) \cdots$, where $V(w) = \{p \in \Phi \mid p \in \pi(w)\} \cup \{initial \mid w = w_0\} \cup \{terminal \mid w \in T\} \cup \{wins(r) \mid w \in g(r) \text{ for } r \in N\} \cup \{legal(r,a) \mid a \in L_r(w) \text{ for } r \in N, a \in \mathcal{A}\}$. Let $trace(M)$ denote the set of all traces in M, i.e., $trace(M) = \{V(\delta) \mid \delta \in \mathcal{P}(M)\}$. Then it holds that

Lemma 2. *For two ST-models M and M', $M \approx M'$ iff $trace(M) = trace(M')$.*

We now provide the logical characterization of bisimulation equivalence.

Theorem 1. *Let M and M' be any two ST-models. Then $M \approx M'$ iff they satisfy the same GDL-formulas.*

Proof. Assume $M \approx M'$. For symmetry, it suffices to prove one case. For every $\varphi \in \mathcal{L}$, assume φ is satisfied in M. then there is $\delta \in \mathcal{P}(M)$ and stage $j \geq 0$ such that $M, \delta, j \models \varphi$. By the assumption and Lemma 1, there is $\delta' \in \mathcal{P}(M')$ such that $M, \delta \leftrightarroweq M', \delta'$. And by Proposition 1, we have $M', \delta', j \models \varphi$. Thus, φ is satisfied in M'.

Now assume $M \not\approx M'$, then by Lemma 2 there is $\delta \in \mathcal{P}(M)$ for all $\delta' \in \mathcal{P}(M')$ $V(\delta) \neq V(\delta')$. It follows that for each $\delta' \in \mathcal{P}(M)$ there is $k \geq 0$ such that either $does(d_{k+1}) \neq does(d'_{k+1})$ or $V(\delta[k]) \neq V(\delta'[k])$. From the former, we obtain a formula $does(r, a_{k+1})$ (for $r \in N$ and $a_{k+1} \in \mathcal{A}$) such that $M, \delta, k \models does(r, a_{k+1})$ and $M', \delta', k \not\models does(r, a_{k+1})$. From the latter, we obtain a formula $\chi \in Atm$ such that either (i) $M, \delta, k \models \chi$ and $M', \delta', k \not\models \chi$, or (ii) $M, \delta, k \models \neg\chi$ and $M', \delta', k \not\models \neg\chi$. Let $\varphi_{\delta'}$ be the formula of the form $\bigcirc^k does(r, a_{k+1})$, $\bigcirc^k \chi$ or $\bigcirc^k \neg\chi$ to distinguish δ from δ'. It follows by the construction that $M, \delta, 0 \models \varphi_{\delta'}$ and $M', \delta', 0 \not\models \varphi_{\delta'}$. Let Δ be the conjunctions of all such obtained formulas for all paths in M', i.e., $\Delta := \bigwedge_{\delta' \in \mathcal{P}(M')} \varphi_{\delta'}$. Note that Δ is well-formed due to the fact that M' is finite-branching. Let us now consider formula $initial \wedge \Delta$. Then it is satisfied in M, i.e., $M, \delta, 0 \models initial \wedge \Delta$. But it is unsatisfied in M'. Otherwise, there are some $\delta' \in \mathcal{P}(M')$ and $j \geq 0$ such that $M', \delta', j \models initial \wedge \Delta$, then $M', \delta', 0 \models \Delta$, so $M', \delta', 0 \models \varphi_{\delta'}$, contradicting with $M', \delta', 0 \not\models \varphi_{\delta'}$. Thus, M and M' fail to satisfy the same set of GDL-formulas. (□)

This theorem 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, it allows us to show the failure of bisimulation-equivalence easily. *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 two ST-models depicted in Figure 3, where $N = \{r\}$ and $\Phi = \emptyset$. One can find formula $initial \wedge \bigcirc^2(does(r, c) \wedge \bigcirc terminal)$ that holds in M, but fails in M'. This leads to $M \not\approx M'$. Thus, two ST-models are bisimulation-equivalent if and only if they enjoy exactly the same properties. Alternatively, two ST-models are not bisimulation-equivalent if one has a property that the other does not have.

3.2 Bisimulation Quotient

In this subsection, we provide an alternative perspective to consider bisimulation as a relation between states within a single ST-model. Then we introduce the quotient ST-model under such relation.

Definition 7. *Let $M = (W, w_0, T, L, U, g, \pi)$ be an ST-models. A bisimulation is a binary relation $Z \subseteq W \times W$ s.t. for all states $w_1, w_2 \in W$ with $w_1 Z w_2$,*

1. $\pi(w_1) = \pi(w_2)$;
2. $w_1 = w_0$ *iff* $w_2 = w_0$;
3. $w_1 \in T$ *iff* $w_2 \in T$;
4. $a \in L_r(w_1)$ *iff* $a \in L_r(w_2)$ *for any* $r \in N$ *and* $a \in \mathcal{A}$;
5. $w_1 \in g(r)$ *iff* $w_2 \in g(r)$ *for any* $r \in N$;
6. *If* $U(w_1, d) = u_1$, *then there is* $u_2 \in W$ *s.t.* $U(w_2, d) = u_2$ *and* $u_1 Z u_2$;
7. *If* $U(w_2, d) = u_2$, *then there is* $u_1 \in W$ *s.t.* $U(w_1, d) = u_1$ *and* $u_1 Z u_2$.

States w_1 and w_2 are bisimulation-equivalent, denoted by $w_1 \sim_M w_2$, if there is a bisimulation Z for M with $w_1 Z w_2$.

It follows that a bisimulation over states for ST-model M is a bisimulation over ST-models for the pair (M, M). Clearly, $\sim_M$ is an equivalence relation on W. For $w \in W$, let $[w]_{\sim_M}$ be the equivalence class of state w under $\sim_M$, i.e., $[w]_{\sim_M} = \{w' \in W \mid w \sim_M w'\}$. We next define the quotient ST-model under such bisimulation equivalence.

Definition 8. *For an ST-model $M = (W, w_0, T, L, U, g, \pi)$ and a bisimulation equivalence $\sim_M$, the quotient ST-model $M/\sim_M = (W', w'_0, T', L', U', g', \pi')$ is defined as follows:*

- $W' = \{[w]_{\sim_M} \mid w \in W\}$ *is the set of all* $\sim_M$*-equivalence classes;*
- $w'_0 = [w_0]_{\sim_M}$;
- $T' = \{[w]_{\sim_M} \mid w \in T\}$;
- $a \in L'_r([w]_{\sim_M})$ *iff* $a \in L_r(w)$ *for any* $r \in N$ *and* $a \in \mathcal{A}$;
- $U'([w]_{\sim_M}, d) = [u]_{\sim_M}$ *iff* $U(w, d) = u'$ *for some* $u' \in [u]_{\sim_M}$;
- $[w]_{\sim_M} \in g'(r)$ *iff* $w \in g(r)$ *for any* $r \in N$;
- $p \in \pi'([w]_{\sim_M})$ *iff* $p \in \pi(w)$ *for any* $p \in \Phi$.

Note that the defined quotient ST-model is indeed a state transition model, and it is minimum as $\sim_M$ is the coarsest bisimulation for M. Moreover, an ST-model and its quotient ST-model are bisimulation-equivalent.

Proposition 2. *For any ST-model M, $M \approx M/{\sim_M}$.*

This follows from the fact that $Z = \{(w, [w]_{\sim_M}) \mid w \in W\}$ is a bisimulation between M and $M/{\sim_M}$. Combining this result and Theorem 1 allows us to perform model checking on the bisimulation-equivalent quotient ST-model. *A GDL-formula holds for the quotient if and only if it also holds for the original ST-model.* This provides a way to improve the efficiency of model checking for GDL in [20, 13]. Note that an adaption of bisimulation-quotienting algorithms for a finite transition system in [2] can be used to compute the quotient ST-model.

4 Bisimulation and Game Equivalence

State transition models may be viewed as representations of games, and bisimulation equivalence tells us when two state transition models are essentially the same. Thus, bisimulation equivalence provides a criterion on the equivalence between games, i.e., two games are equivalent if their state transition models are bisimulation-equivalent. In this section, we generalize this concept to capture more general game equivalence.

Let us first consider the following two games: Number Scrabble in Example 1 and Tic-Tac-Toe specified as follows:

Example 2 (Tic-Tac-Toe). Two players take turns in marking either a cross ‘x’or a nought ‘o’ on a 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; $\mathcal{A}_{TT} = \{a_{i,j} \mid 1 \le i, j \le 3\} \cup \{noop\}$, 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 $1 \le i, j \le 3\}$, 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. $wins(r) \leftrightarrow \bigvee_{i=1}^{3} \bigwedge_{l=0}^{2} p^r_{i,1+l} \vee \bigvee_{j=1}^{3} \bigwedge_{l=0}^{2} p^r_{1+l,j} \vee \bigwedge_{l=0}^{2} p^r_{1+l,1+l} \vee \bigwedge_{l=0}^{2} p^r_{1+l,3-l}$
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. 4. 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 specifies the state transitions.

Although the two games appear different in their game descriptions, they are actually equivalent (isomorphic) [15, 18]. Unfortunately, bisimulation equivalence is not able to capture such game equivalence as they are based on different game signatures. Then we generalize the notion of bisimulation equivalence as follows:

Definition 9. *Consider two ST-models* $M_{\mathcal{S}} = (W, w_0, T, L, U, g, \pi)$ *with* $\mathcal{S} = (N, \mathcal{A}, \Phi)$ *and* $M'_{\mathcal{S}'} = (W', w'_0, T', L', U', g', \pi')$ *with* $\mathcal{S}' = (N', \mathcal{A}', \Phi')$. $M_{\mathcal{S}}$ *and* $M'_{\mathcal{S}'}$ *are structure-equivalent, written* $M_{\mathcal{S}} \sim M'_{\mathcal{S}'}$*, if there are bijections* $f_1 : N \mapsto N'$, $f_2 : \mathcal{A} \mapsto \mathcal{A}'$, $f_3 : \Phi \mapsto \Phi'$*, and a relation* $Z \subseteq W \times W'$ *such that* $w_0 Z w'_0$ *and for all states* $w \in W$ *and* $w' \in W'$ *with* wZw'*, the following conditions hold:*

1. $p \in \pi(w)$ *iff* $f_3(p) \in \pi'(w')$*;*
2. $w = w_0$ *iff* $w' = w'_0$*;*
3. $w \in T$ *iff* $w' \in T'$*;*
4. $a \in L_r(w)$ *iff* $f_2(a) \in L'_{f_1(r)}(w')$ *for any* $r \in N$ *and* $a \in \mathcal{A}$*;*
5. $w \in g(r)$ *iff* $w' \in g'(f_1(r))$ *for any* $r \in N$*;*
6. *If* $U(w, d) = u$*, then there is* $u' \in W'$ *s.t.* $U'(w', \langle f_2(d(r))\rangle_{r \in N}) = u'$ *and* uZu'*;*
7. *If* $U'(w', d') = u'$*, then there is* $u \in W$ *s.t.* $U(w, \langle f_2^{-1}(d'(r'))\rangle_{r' \in N'}) = u$ *and* uZu'.

Note that $\sim$ is an equivalence relation over ST-models (with different game signatures). Clearly, $\approx$ is a special case of $\sim$ when $\mathcal{S} = \mathcal{S}'$. We say that two games are *equivalent* if their ST-models are structure-equivalent.

Let us back to the examples. As we expected, the equivalence between Number Scrabble and Tic-Tac-Toe can be captured by the structure equivalence. The mapping between their state transition models is demonstrated in Table 1. The basic idea is that *filling a cell corresponds to selecting the number in the cell, and the fact that a cell is filled amounts to the fact that the corresponding number is selected.* For instance, filling the left-bottom cell corresponds to selecting number 4, i.e., $f_2(a_{1,1}) = \alpha(4)$, and the fact that the center is filled by player x maps the fact that number 5 is selected by player b, i.e., $f_3(p^{\mathsf{x}}_{2,2}) = s(\mathsf{b}, 5)$. And the structure-bisimulation relation starts from their initial states and can be constructed step by step according to the mapping. For example, the states depicted in Table 2 and Figure 5 are structure-bisimilar.

2	7	6
9	5	1
4	3	8

Table 1: The Mapping.

	X	

Table 2: Filling the Center.

$\{1,2,3,4,\underline{5},6,7,8,9\}$
b

Fig. 5. Selecting Number 5.

Similarly, we say that w and w' are structure-bisimilar, written $M_{\mathcal{S}}, w \leftrightarrows_s M'_{\mathcal{S}'}, w'$, if Z links two states w in $M_{\mathcal{S}}$ and w' in $M'_{\mathcal{S}'}$. In particular, for two paths $\delta := w_0 \xrightarrow{d_1} w_1 \xrightarrow{d_2} \cdots \xrightarrow{d_j} \cdots$ in $M_{\mathcal{S}}$ and $\delta' := w'_0 \xrightarrow{d'_1} w'_1 \xrightarrow{d'_2} \cdots \xrightarrow{d'_j} \cdots$ in $M'_{\mathcal{S}'}$, we say that δ and δ' are *structure-bisimilar*, written $M_{\mathcal{S}}, \delta \leftrightarrows_s M'_{\mathcal{S}'}, \delta'$, iff for every $j \geq 0$ and $r \in N$, $M_{\mathcal{S}}, \delta[j] \leftrightarrows_s M'_{\mathcal{S}'}, \delta'[j]$ and $f_2(\theta_r(\delta, j)) = \theta_{f_1(r)}(\delta', j)$. Similar to Bisimulation Equivalence, the following result displays that two ST-models are structure-equivalent iff each path that can be developed in one model can be also *simulated* in the other.

Lemma 3. *Given two ST-models* $M_{\mathcal{S}}$ *and* $M'_{\mathcal{S}'}$, $M_{\mathcal{S}} \sim M'_{\mathcal{S}'}$ *iff for every* $\delta \in \mathcal{P}(M_{\mathcal{S}})$, *there is* $\delta' \in \mathcal{P}(M'_{\mathcal{S}'})$ *such that* $M_{\mathcal{S}}, \delta \leftrightarrows_s M'_{\mathcal{S}'}, \delta'$, *and vice versa.*

Let us turn to the logical characterization of structure equivalence. To this end, we begin with the transformation of GDL-formulas. The translation between languages is defined as follows:

Definition 10. *Consider two game signatures* $\mathcal{S} = (N, \mathcal{A}, \Phi)$ *and* $\mathcal{S}' = (N', \mathcal{A}', \Phi')$ *with the same bijections* $f_1 : N \mapsto N'$, $f_2 : \mathcal{A} \mapsto \mathcal{A}'$ *and* $f_3 : \Phi \mapsto \Phi'$ *of Definition 9. A translation* tr *is a bijective mapping from* $\mathcal{L}_{\mathcal{S}}$ *onto* $\mathcal{L}_{\mathcal{S}'}$ *such that for* $p \in \Phi$, $r \in N$ *and* $a \in \mathcal{A}$,

$$\begin{array}{ll} \mathrm{tr}(p) = f_3(p) & \mathrm{tr}(initial) = initial \\ \mathrm{tr}(terminal) = terminal & \mathrm{tr}(wins(r)) = wins(f_1(r)) \\ \mathrm{tr}(legal(r,a)) = legal(f_1(r), f_2(a)) & \mathrm{tr}(does(r,a)) = does(f_1(r), f_2(a)) \\ \mathrm{tr}(\neg\varphi) = \neg\mathrm{tr}(\varphi) & \mathrm{tr}(\varphi \wedge \psi) = \mathrm{tr}(\varphi) \wedge \mathrm{tr}(\psi) \\ \mathrm{tr}(\bigcirc\varphi) = \bigcirc\mathrm{tr}(\varphi) & \end{array}$$

Note that such a translation exists as there is a bijective mapping between the game signatures. The following result holds that if two paths are structure-bisimilar, then they preserve the corresponding GDL-formulas at each stage.

Lemma 4. *Let* $M_{\mathcal{S}}$, $M'_{\mathcal{S}'}$ *be two ST-models. For every* $\delta \in \mathcal{P}(M_{\mathcal{S}})$ *and* $\delta' \in \mathcal{P}(M'_{\mathcal{S}'})$, *if* $M_{\mathcal{S}}, \delta \leftrightarrows_s M'_{\mathcal{S}'}, \delta'$, *then* $M_{\mathcal{S}}, \delta, j \models \varphi$ *iff* $M'_{\mathcal{S}'}, \delta', j \models \mathrm{tr}(\varphi)$ *for any* $\varphi \in \mathcal{L}_{\mathcal{S}}$ *and* $j \geq 0$.

Note that the converse to this proposition does not hold. Please refer to Figure 2 for a counter-example. We now provide the following logical characterization result that structure equivalence and the invariance of the corresponding GDL-formulas coincide on ST-models.

Proposition 3. *Let* $M_{\mathcal{S}}$, $M'_{\mathcal{S}'}$ *be two ST-models. The following are equivalent.*

1. $M_{\mathcal{S}} \sim M'_{\mathcal{S}'}$
2. *for every* $\varphi \in \mathcal{L}_{\mathcal{S}}$, φ *is satisfied in* $M_{\mathcal{S}}$ *iff* $\mathrm{tr}(\varphi)$ *is satisfied in* $M'_{\mathcal{S}'}$.

Proof. The direction from Clause 1 to Clause 2 follows from Lemma 3 and Lemma 4. To prove the other direction, we need the following notion.

Consider two ST-models $M_{\mathcal{S}} = (W, w_0, T, L, U, g, \pi)$ with $\mathcal{S} = (N, \mathcal{A}, \Phi)$ and $M'_{\mathcal{S}'} = (W', w'_0, T', L', U', g', \pi')$ with $\mathcal{S}' = (N', \mathcal{A}', \Phi')$. Given bijections $f_1 : N \mapsto N'$, $f_2 : \mathcal{A} \mapsto \mathcal{A}'$ and $f_3 : \Phi \mapsto \Phi'$, let tr be a translation defined in Definition 10. A translation Tr is a bijection from $trace(M)$ to $trace(M')$. For every $V(\delta) \in trace(M)$, $\mathrm{Tr}(V(\delta)) = \mathrm{tr}(V(w_0)) \cdot \mathrm{tr}(does(d_1)) \cdot \mathrm{tr}(V(w_1)) \cdots \mathrm{tr}(does(d_{e-1})) \cdot \mathrm{tr}(V(w_e))$ where $\mathrm{tr}(V(w_j)) = \{\mathrm{tr}(\varphi) \in \mathcal{L}_{\mathcal{S}'} : \varphi \in V(w_j)\}$ for any $0 \leq j \leq e$, and $\mathrm{tr}(does(d_j)) = \bigwedge_{r \in N} \mathrm{tr}(does(r, d_j(r)))$ for any $1 \leq j \leq e$. Let $\mathrm{Tr}(trace(M)) = \{\mathrm{Tr}(V(\delta)) \mid V(\delta) \in trace(M)\}$. Then we have that the fact holds that $M_{\mathcal{S}} \sim M'_{\mathcal{S}'}$ iff $\mathrm{Tr}(trace(M)) = trace(M')$. With this, the proof of the direction from Clause 2 to Clause 1 is similar to Theorem 1. (□)

We end this section with the interesting observation that the GDL-descriptions of Tic-Tac-Toe and Number Scrabble are logically equivalent in terms of the translation.

Observation 1 *Let Σ_{TT} and Σ_{NS} denote the GDL-descriptions of Tic-Tac-Toe (Figure 6) and Number Scrabble (Figure 1), respectively. Then $\models \bigwedge \mathrm{tr}(\Sigma_{TT}) \leftrightarrow \bigwedge \Sigma_{NS}$, where $\mathrm{tr}(\Sigma_{TT}) = \{\mathrm{tr}(\varphi) \in \mathcal{L}_{NS} \mid \varphi \in \Sigma_{TT}\}$.*

5 Conclusion

We have defined the notion of bisimulation for GDL and showed that it coincides with the invariance of GDL-formulas. We have also introduced the quotient model to improve the efficiency of model checking for GDL. Moreover, we have generalized the notion of bisimulation to capture more general game equivalence.

Although various game equivalence have been proposed in economics, mathematics and logic [1, 5, 8, 22], few work has been done in the domain of GGP. To the best of our knowledge, Zhang *et al.* investigated game equivalence for knowledge transfer in GGP. They consider two games are equivalent if their state transition models are isomorphic [25]. While our notion of game equivalence is more general as it is based on bisimulation relation.

Directions of future research are manifold. We intend to explore the van Benthem Characterization Theorem for GDL [4]. More recently, GDL has been extended to GDL-II and epistemic GDL for representing and reasoning about imperfect information games [13, 21]. We plan to study the expressiveness of these extended languages. Besides structure equivalence, it would be also interesting to investigate different types of game equivalence in GGP, such as strategic equivalence, subgame equivalence [6, 10, 17].

Acknowledgments. We are grateful to three anonymous referees for their insightful comments. 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., Vardi, M.Y.: Alternating refinement relations. In: International Conference on Concurrency Theory. pp. 163–178. Springer (1998)
2. Baier, C., Katoen, J.P., Larsen, K.G.: Principles of model checking. MIT press (2008)
3. van Benthem, J.: Modal Correspondence Theory. Ph.D. thesis, University of Amsterdam (1977)
4. van Benthem, J.: Correspondence theory. In: Handbook of philosophical logic, pp. 167–247. Springer (1984)

5. van Benthem, J.: Logic in games. MIT Press (2014)
6. 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)
7. Blackburn, P., van Benthem, J., Wolter, F.: Handbook of modal logic, vol. 3. Elsevier (2006)
8. Elmes, S., Reny, P.J.: On the strategic equivalence of extensive form games. Journal of Economic Theory **62**(1), 1–23 (1994)
9. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. AI magazine **26**(2), 62–72 (2005)
10. Goranko, V.: The basic algebra of game equivalences. Studia Logica **75**(2), 221–238 (2003)
11. Grädel, E., Otto, M.: The freedoms of (guarded) bisimulation. In: Johan van Benthem on Logic and Information Dynamics, pp. 3–31. Springer (2014)
12. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. Journal of the ACM (JACM) **32**(1), 137–161 (1985)
13. 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)
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. Michon, J.A.: The game of jam: An isomorph of tic-tac-toe. The American Journal of Psychology **80**(1), 137–140 (1967), `http://www.jstor.org/stable/1420555`
16. Park, D.: Concurrency and automata on infinite sequences. In: Theoretical computer science, pp. 167–183. Springer (1981)
17. Pauly, M.: Logic for Social Software. PhD thesis, University of Amsterdam (2001), `ILLCDissertationSeries2001-10`
18. Pell, B.: Strategy generation and evaluation for meta-game playing. Ph.D. thesis, University of Cambridage (1993)
19. 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)
20. 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)
21. 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)
22. Thompson, F.: Equivalence of games in extensive form. Classics in game theory **36** (1997)
23. Zhang, D., Thielscher, M.: A logic for reasoning about game strategies. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15). pp. 1671–1677 (2015)
24. Zhang, D., Thielscher, M.: Representing and reasoning about game strategies. Journal of Philosophical Logic **44**(2), 203–236 (2015)
25. Zhang, H., Liu, D., Li, W.: Space-consistent game equivalence detection in general game playing. In: Workshop on Computer Games. pp. 165–177. Springer (2015)