Game Characterizations of Process Equivalences

Xin Chen⋆ and Yuxin Deng⋆⋆

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China

Abstract. In this paper we propose a hierarchy of games that allows us to make a systematic comparison of process equivalences by characterizing process equivalences as games. The well-known linear/branching time hierarchy of process equivalences can be embedded into the game hierarchy, which not only provides us with a refined analysis of process equivalences, but also offers a guidance to defining interesting new process equivalences.

1 Introduction

A great amount of work in process algebra has centered around process equivalences as a basis for establishing system correctness. Usually both specifications and implementations are written as process terms in the same algebra, where a specification describes the expected high-level behaviour of the system under consideration and an implementation gives the detailed procedure of achieving the behaviour. An appropriate equivalence is then chosen to verify that the implementation conforms to the specification. In the last three decades, a lot of process equivalences have been developed to capture various aspects of system behaviour. They usually fit in the linear/branching time hierarchy [10]; see Figure 1 for some typical process equivalences.

Process equivalences can often be understood from different perspectives such as logics and games. For example, bisimulation equivalence can be characterized by Hennessy-Milner logic [1] and the modal mu-calculus [2]. Equivalences which are weaker than bisimulation equivalence in the linear/branching time hierarchy can be characterized by some sub-logics of Hennessy-Milner logic [3]. It is also well-known that bisimulation equivalence can be characterized by bisimulation games [6] between an attacker and a defender in an elegant way; two processes are bisimilar if and only if the defender of a bisimulation game played on the processes has a history free winning strategy. Bisimulation games came from Ehrenfeucht-Fraïssé games that were originally introduced to determine expressive power of logics [9]. To some extent games can be considered as descriptive languages like logics. In many cases we can design a game directly from the semantics of a particular logic such that the game captures the logic. For example, the bisimulation game with infinite duration is an Ehrenfeucht-Fraïssé game

⋆ Supported by the National 973 Project (2003CB317005) and the National Natural Science Foundation of China (60573002).

⋆⋆ Supported by the National Natural Science Foundation of China (60703033).

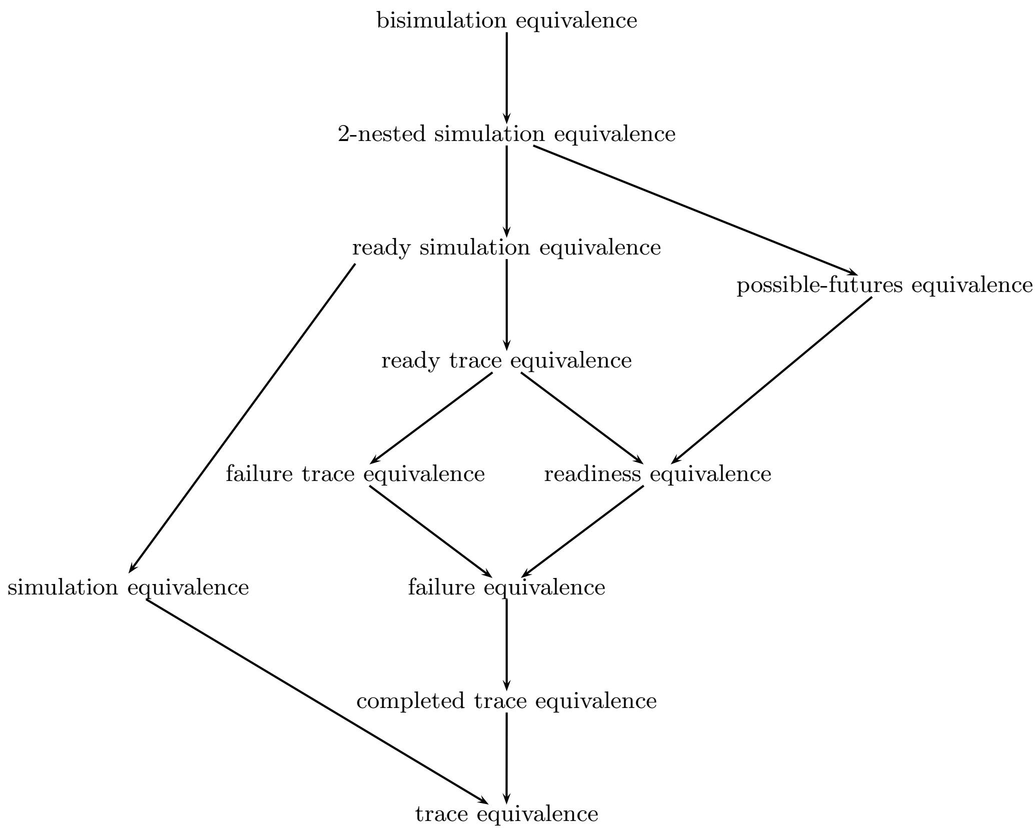

Fig. 1. The linear/branching time hierarchy [10]

that captures Hennessy-Milner logic [6], and the fixed point game that allows infinite fixed point and modal moves captures the modal mu-calculus [8]. Games indeed offer new sights into old problems, and sometimes let us understand these problems easier than before.

In this paper we provide a systematic comparison of different process equivalences from a game-theoretic point of view. More precisely, we present a game hierarchy (cf. Figure 4) which has a more refined structure than the process equivalence hierarchy in Figure 1. Viewing the hierarchies as partial orders, we can embed the process equivalence hierarchy into the game hierarchy because each process equivalence can be characterized by a corresponding class of games. Moreover, there are games that do not correspond to any existing process equivalences. This kind of games would be useful for guiding us to define interesting new process equivalences.

To define games, we make use of a game template that is basically an abstract two-player game leaving concrete moves unspecified. Then we define a few types of moves. Instantiating the game template by different combinations of moves generates different games. We compare the games using a preorder which says that $\mathcal{G}_1 \succeq \mathcal{G}_2$ if player II has a winning strategy in $\mathcal{G}_1$ implies she has a winning strategy in $\mathcal{G}_2$. The preorder provides us with a neat means to compare process equivalences. Suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ characterize process equivalences $\sim_1$ and $\sim_2$, respectively. Then we have that $\mathcal{G}_1 \succeq \mathcal{G}_2$ if and only if $\sim_1 \subseteq \sim_2$, i.e. $\sim_2$ is a coarser relation than $\sim_1$.

The rest of the paper is organized as follows. Section 2 briefly recalls the definitions of labelled transition systems and bisimulations. In Section 3, we design several kinds of moves and a game template in order to define games. In Section 4, we present two game hierarchies, with or without considering alternations of moves, and we combine them into a final hierarchy. In Section 5, we show that the linear/branching time hierarchy can be embedded into our game hierarchy. Section 6 concludes and discusses some future work.

2 Preliminaries

We presuppose a countable set of actions $Act = \{a, b, \dots\}$.

Definition 1. *A labelled transition systems (LTS) is a triple* $(\mathcal{P}, A, \rightarrow)$*, where*

- $\mathcal{P}$ *is a set of states,*
- $A \subseteq Act$ *is a set of actions,*
- $\rightarrow \subseteq \mathcal{P} \times A \times \mathcal{P}$ *is a transition relation.*

As usual, we write $P \xrightarrow{a} Q$ for $(P, a, Q) \in \rightarrow$ and we extend the transition relation to traces in the standard way, e.g. $P_0 \xrightarrow{t} P_n$ if $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} P_2 \dots P_{n-1} \xrightarrow{a_n} P_n$, where $t = a_1 a_2 \dots a_n$. An LTS $(\mathcal{P}, A, \rightarrow)$ is *image-finite* if for all $P \in \mathcal{P}$ the set $\{P' \mid P \xrightarrow{a} P',$ for some $a \in A\}$ is finite. In this paper we only consider image-finite LTSs. Instead of drawing LTSs as graphs, we use CCS processes to represent the LTSs generated by their operational semantics [4]. We say two processes are isomorphic if their LTSs are isomorphic.

Definition 2. *A binary relation $\mathcal{R}$ is a bisimulation if for all $(P, Q) \in \mathcal{R}$ and $a \in Act$,*

(1) whenever $P \xrightarrow{a} P'$, there exists $Q \xrightarrow{a} Q'$ such that $(P', Q') \in \mathcal{R}$, and
(2) whenever $Q \xrightarrow{a} Q'$, there exists $P \xrightarrow{a} P'$ such that $(P', Q') \in \mathcal{R}$.

We define the union of all bisimulations as bisimilarity, written $\sim$.

Bisimilarity can be approximated by a sequence of inductively defined relations. The following definition is taken from [4], except that $\sim_k$ is replaced by $\sim_k^r$. The meaning of the superscript r will be clear in Section 5.

Definition 3. *Let $\mathcal{P}$ be the set of all processes, we define*

- $\sim_0^r = \mathcal{P} \times \mathcal{P}$,
- $P \sim_{n+1}^r Q$, *for* $n \geq 0$, *if for all* $t \in Act^*$,
  (1) whenever $P \xrightarrow{t} P'$, there exists $Q \xrightarrow{t} Q'$ such that $P' \sim_n^r Q'$,
  (2) whenever $Q \xrightarrow{t} Q'$, there exists $P \xrightarrow{t} P'$ such that $P' \sim_n^r Q'$.

The definition of $\sim_k^a$ for $k \geq 0$ is similar to the previous one, except that we replace $\xrightarrow{t}$ with $\xrightarrow{a}$ where $a \in Act$. For image-finite LTSs, it holds that $\sim \; = \; \bigcap_{n \geq 0} \sim_n^r \; = \; \bigcap_{n \geq 0} \sim_n^a$.

3 Game Template

We briefly review the bisimulation games [8]. A bisimulation game $\mathcal{G}_k(P, Q)$ starting from the pair of processes (P, Q) is a round-based game with two players. Player I, viewed as an attacker, attempts to show that the initial states are different whereas player II, viewed as a defender, wishes to establish that they are equivalent. A configuration is a pair of processes of the form (P_i, Q_i) examined in the i-th round, and (P, Q) is the configuration for the first round. Suppose we are in the i-th round. The next configuration (P_{i+1}, Q_{i+1}) is determined by one of the following two moves:

$\langle a \rangle$: Player I chooses a transition $P_i \xrightarrow{a} P_{i+1}$ and then player II chooses a transition with the same label $Q_i \xrightarrow{a} Q_{i+1}$.
$[a]$: Player I chooses a transition $Q_i \xrightarrow{a} Q_{i+1}$ and then player II chooses a transition with the same label $P_i \xrightarrow{a} P_{i+1}$.

Player I wins if she can choose a transition and player II is unable to match it within k rounds. Otherwise, Player II wins. If $k = \infty$ then there is no limitation on the number of rounds.

Below we define four other moves that will give rise to various games later on.

Definition 4 (Moves). *Suppose the current configuration is (P, Q), we define the following kinds (or sets, more precisely) of moves.*

$\langle t \rangle$: *Player I performs a nonempty action sequence* $t = a_1 \cdots a_l \in Act^*$ *from* P, $P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$ *and then player II performs the same action sequence from* Q, $Q \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$. *Player I selects some* $1 \leq j \leq l$ *and sets the configuration for the next round to be* (P_j, Q_j).

$[t]$: *Player I performs a nonempty action sequence* $t = a_1 \cdots a_l \in Act^*$ *from* Q, $Q \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$ *and then player II performs the same action sequence from* P, $P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$. *Player I selects some* $1 \leq j \leq l$ *and sets the configuration for the next round to be* (P_j, Q_j).

r-$\langle t \rangle$: *Player I performs a nonempty action sequence* $t = a_1 \cdots a_l \in Act^*$ *from* P, $P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$ *and then player II performs the same action sequence from* Q, $Q \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$. *The configuration for the next round is* (P_l, Q_l).

r-$[t]$: *Player I performs a nonempty action sequence* $t = a_1 \cdots a_l \in Act^*$ *from* Q, $Q \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$ *and then player II performs the same action sequence from* P, $P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$. *The configuration for the next round is* (P_l, Q_l).

For the sake of convenience, we define some unions of the moves above:

- $t := \langle t \rangle \cup [t]$.
- $r := r\text{-}\langle t \rangle \cup r\text{-}[t]$.
- $a := \langle a \rangle \cup [a]$.
- $\mathfrak{M}$ *is the set of all moves.*

Clearly, $\langle a \rangle$ moves are special r-$\langle t \rangle$ moves and r-$\langle t \rangle$ moves are special $\langle t \rangle$ moves. We have similar observation for box modalities.

- $\langle a \rangle \subsetneq r\text{-}\langle t \rangle \subsetneq \langle t \rangle$.
- $[a] \subsetneq r\text{-}[t] \subsetneq [t]$.

We now introduce the concept of alternation for games; it has an intimate relation with quantifier alternation in logics.

Definition 5 (Alternation). *An* alternation *consists of two successive moves such that one of them is in* $\langle t \rangle$ *and the other is in* $[t]$. *The number of alternations in a game is the number of occurrences of such successive moves in the game.*

Note that bisimulation games have no restriction on their alternation numbers.

Definition 6 (Extra conditions). *Given a round-based game and a set* α *which is the set of moves player I can make in the game, an extra condition can be one of the following, for some* $m \subseteq \mathfrak{M}$,

m: *The game is extended with one more round, where player I can only make a move in* m. *Moreover, player I can make a move in* $m - \alpha$ *in each round, but the game has to be finished regardless of the remaining rounds, which implies that if player I fails to make player II stuck by this move, she loses.*

$-m$: *Similar to the case for* m*, except that if player I makes a move in* $m - \alpha$ *to end the game, the last two moves must be an alternation. Therefore, this condition could not be applied to a* 0*-round game.*
c_0: *In the beginning of the game, all deadlock processes reachable from* P_0 *and* Q_0 *are colored* $\mathcal{C}_0$*. In each round, the two processes in the related configuration should be in the same color (or neither of them is colored), otherwise player II loses.*

We now define a *game template* which is intuitively an abstract game in the sense that concrete games can be obtained from it by instantiating its moves.

Definition 7 (Game template). *The game template* $n\text{-}\Gamma_k^{\alpha,\beta}(P, Q)$ *with* $n \geq 0$ *denotes a* k*-round game between player I and player II with the starting configuration* (P, Q) *such that the following conditions are satisfied.*

1. *The number of alternations in the game is at most* n*; it is omitted when there is no restriction on the number of alternations.*
2. β *is an extra condition; it is omitted when there is no extra condition.*
3. *Player I can only make a move in* $\alpha \subseteq \mathfrak{M}$ *in each round if* β *is neither* m *nor* $-m$*. Otherwise, player I can also make a move in* $m - \alpha$ *in each round, but if she cannot make player II stuck by this move, she loses.*
4. *The players' winning conditions are similar to those in bisimulation games.*

Notice that k-round bisimulation games can be defined by Γ_k^a. Although a lot of games can be defined by various combinations of n, α and β; this paper mainly focuses on some typical ones. Given a game $\Gamma_k^{\alpha,\beta}(P, Q)$, we say player I (resp. player II) wins $\Gamma_k^{\alpha,\beta}(P, Q)$ if player I (resp. player II) has a winning strategy in it, and we abbreviate the game to $\Gamma_k^{\alpha,\beta}$ if the starting configuration is insignificant.

4 Game Hierarchy

To facilitate the presentation, we classify our games into two hierarchies with respect to a preorder relation between games; one hierarchy counts alternations of moves and the other does not count. We show that all the relations in the hierarchies are correct. Then we combine the two hierarchies into one, by introducing some new relations. At last, we prove that no more non-trivial relations can be added into the final hierarchy. We shall see in Section 5 that the hierarchy of process equivalences in Figure 1 can be embedded into this hierarchy of games.

The preorder relation between games is defined as follows.

Definition 8. *Given two games* $\mathcal{G}_1$ *and* $\mathcal{G}_2$*, we write* $\mathcal{G}_1 \succeq \mathcal{G}_2$*, if for any processes* P *and* Q*,*

$$\text{player II wins } \mathcal{G}_1(P, Q) \Longrightarrow \text{player II wins } \mathcal{G}_2(P, Q).$$

Here $\succeq$ *is indeed a preorder as this is inherited from logical implication. We write* $\mathcal{G}_1 \succ \mathcal{G}_2$ *if* $\mathcal{G}_1 \succeq \mathcal{G}_2$ *and* $\mathcal{G}_2 \not\succeq \mathcal{G}_1$*.*

4.1 Game Hierarchy I

We propose the game hierarchy I in Figure 2. Its correctness is stated by the next theorem.

Theorem 1. *In Figure 2, if $\mathcal{G}_1 \to \mathcal{G}_2$ then $\mathcal{G}_1 \succeq \mathcal{G}_2$.*

The rest of this subsection is devoted to proving Theorem 1.

Let $\alpha, \alpha' \subseteq \mathfrak{M}$ and β be an extra condition. The following statements can be derived from Definition 7 immediately:

(1) $\Gamma_0^{\alpha} = \Gamma_0^{\alpha'}$.
(2) $\Gamma_0^{\alpha,\beta} = \Gamma_0^{\alpha',\beta}$.
(3) For $k \geq 0$, $\Gamma_k^{\alpha,\alpha} = \Gamma_{k+1}^{\alpha}$.
(4) For $k \geq 0$, $\Gamma_k^{\alpha,t} = \Gamma_k^{\alpha,r}$.

Since t contains r, r contains a, and if two processes P, Q do not have the same color, they can be distinguished in a round by a move in a, we get the following statement:

$$\Gamma_\infty^{\alpha} = \Gamma_\infty^{\alpha,c_0} = \Gamma_\infty^{\alpha,-a} = \Gamma_\infty^{\alpha,a}, \text{ for } \alpha \in \{a, r, t\}.$$

Lemma 1. *For any processes P and Q, the following statements are equivalent:*

(1) $P \sim Q$.
(2) player II wins $\Gamma_\infty^a(P,Q)$.
(3) player II wins $\Gamma_\infty^r(P,Q)$.
(4) player II wins $\Gamma_\infty^t(P,Q)$.

Proof. It is trivial that $\Gamma_\infty^a \preceq \Gamma_\infty^r \preceq \Gamma_\infty^t$, so we have (4)⇒(3)⇒(2). Observe that Γ_k^a is exactly the k-round bisimulation game, which means (1)⇔(2) (cf. [6]). We now show (1)⇒(4). Assume $P \sim Q$, we construct a winning strategy for player II for the game $\Gamma_\infty^t(P,Q)$: in any round, suppose the configuration is (P_i, Q_i). If player I performs $P_i \xrightarrow{a_1} P_{i1} \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_{il}$, then player II can respond with $Q_i \xrightarrow{a_1} Q_{i1} \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_{il}$, such that $P_{ij} \sim Q_{ij}$ for all $1 \leq j \leq l$. Clearly, whatever configuration for the next round player I selects, she cannot win the game. □

Lemma 1 yields the immediate corollary that $\Gamma_\infty^a = \Gamma_\infty^r = \Gamma_\infty^t$.

Lemma 2. *(1) $\Gamma_{k+1}^a \preceq \Gamma_k^{r,a} \preceq \Gamma_k^{t,a}$ for all $k \geq 1$.*
(2) $\Gamma_k^r \preceq \Gamma_k^t$ for all $k \geq 1$.

Proof. Since $a \subsetneq r \subsetneq t$, both (1) and (2) can be easily derived. □

Lemma 3. $\Gamma_k^r \preceq \Gamma_k^{r,c_0} \preceq \Gamma_k^{r,-a} \preceq \Gamma_k^{r,a} \preceq \Gamma_{k+1}^r$ *for all $k \geq 1$.*

Proof. It is easy to see that $\Gamma_k^r \preceq \Gamma_k^{r,c_0} \preceq \Gamma_k^{r,a} \preceq \Gamma_{k+1}^r$ and $\Gamma_k^r \preceq \Gamma_k^{r,-a} \preceq \Gamma_k^{r,a} \preceq \Gamma_{k+1}^r$. We now prove $\Gamma_k^{r,c_0} \preceq \Gamma_k^{r,-a}$ by induction on k. Given two processes P, Q, suppose player II wins $\Gamma_k^{r,-a}(P,Q)$. We show that player II wins $\Gamma_k^{r,c_0}(P,Q)$ as well.

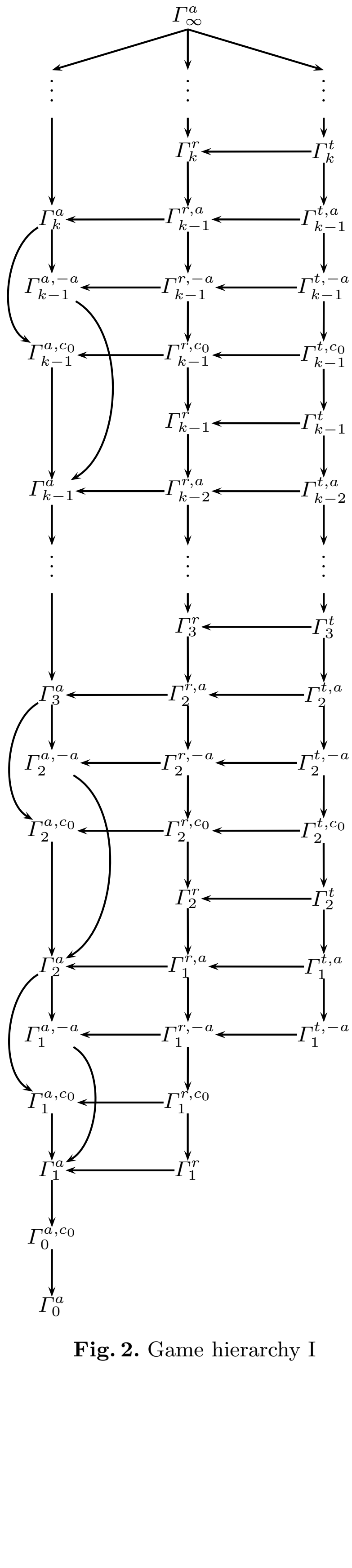

Fig. 2. Game hierarchy I

– $k = 1$. From the assumption, player II wins $\Gamma_1^{r,-a}(P,Q)$. The game $\Gamma_1^{r,c_0}(P,Q)$ has just one round and all deadlock processes reachable from P and Q are colored $\mathcal{C}_0$, and the other processes are uncolored. (Clearly both P and Q are colored $\mathcal{C}_0$ or neither of them is colored.) We distinguish four cases.

Case 1: Player I performs $P \xrightarrow{t} P'$, where $t \in Act^*$ is a nonempty action sequence and P' is colored. Player II can perform $Q \xrightarrow{t} Q'$ such that Q' is colored. Otherwise there is some $a \in Act$ and player I can make player II stuck by performing $P \xrightarrow{ta} P''$ for some P'' in the first round of $\Gamma_1^{r,-a}(P,Q)$, which contradicts the assumption.

Case 2: Player I performs $P \xrightarrow{t} P'$, where $t \in Act^*$ is a nonempty action sequence and P' is uncolored $\mathcal{C}_0$. Player II can perform $Q \xrightarrow{t} Q'$ such that Q' is uncolored $\mathcal{C}_0$. Otherwise, in $\Gamma_1^{r,-a}(P,Q)$, player I can make player II stuck by making a move in $[a]$ in the second round, contradicting the assumption.

Case 3: Player I performs $Q \xrightarrow{t} Q'$, where $t \in Act^*$ is a nonempty action sequence and Q' is colored. This case is similar to Case 1.

Case 4: player I performs $Q \xrightarrow{t} Q'$, where $t \in Act^*$ is a nonempty action sequence and Q' is uncolored $\mathcal{C}_0$. This case is similar to Case 2.

– $k > 1$. We know player II wins $\Gamma_k^{r,-a}(P,Q)$. In the first round of $\Gamma_k^{r,c_0}(P,Q)$, whenever player I performs some action sequence from P (resp. Q) to P' (resp. Q'), player II can always perform the same action sequence from Q (resp. P) to Q' (resp. P') such that both P' and Q' are colored $\mathcal{C}_0$, or neither of them is colored. Otherwise, in $\Gamma_k^{r,-a}(P,Q)$, player I can make player II stuck in the second round. In the second round of $\Gamma_k^{r,c_0}(P,Q)$, the game becomes $\Gamma_{k-1}^{r,c_0}(P',Q')$ and by induction player II wins the $\Gamma_{k-1}^{r,c_0}(P',Q')$. □

Similar to Lemmas 2 and 3, all the other relations illustrated in Figure 2 can be proven, thus Theorem 1 is established.

4.2 Game Hierarchy II

The games in Section 4.1 do not count alternations of moves, which are taken into account in this section. For simplicity, we are not going to discuss all the games defined from those in Figure 2 by restricting the number of alternations. Instead, we focus on the games in which the players can only make moves in a. To further simplify the exposition, Figure 3 only illustrates a game hierarchy where the number of alternations n is restricted to 0 and 1. However, in the rest of the paper the lemmas cover all $n \geq 0$. From Definitions 6 and 7, the relations illustrated in Figure 3 are apparent, so we omit the proof of the theorem below.

Theorem 2. *In Figure 3, if* $\mathcal{G}_1 \rightarrow \mathcal{G}_2$ *then* $\mathcal{G}_1 \succeq \mathcal{G}_2$*.* □

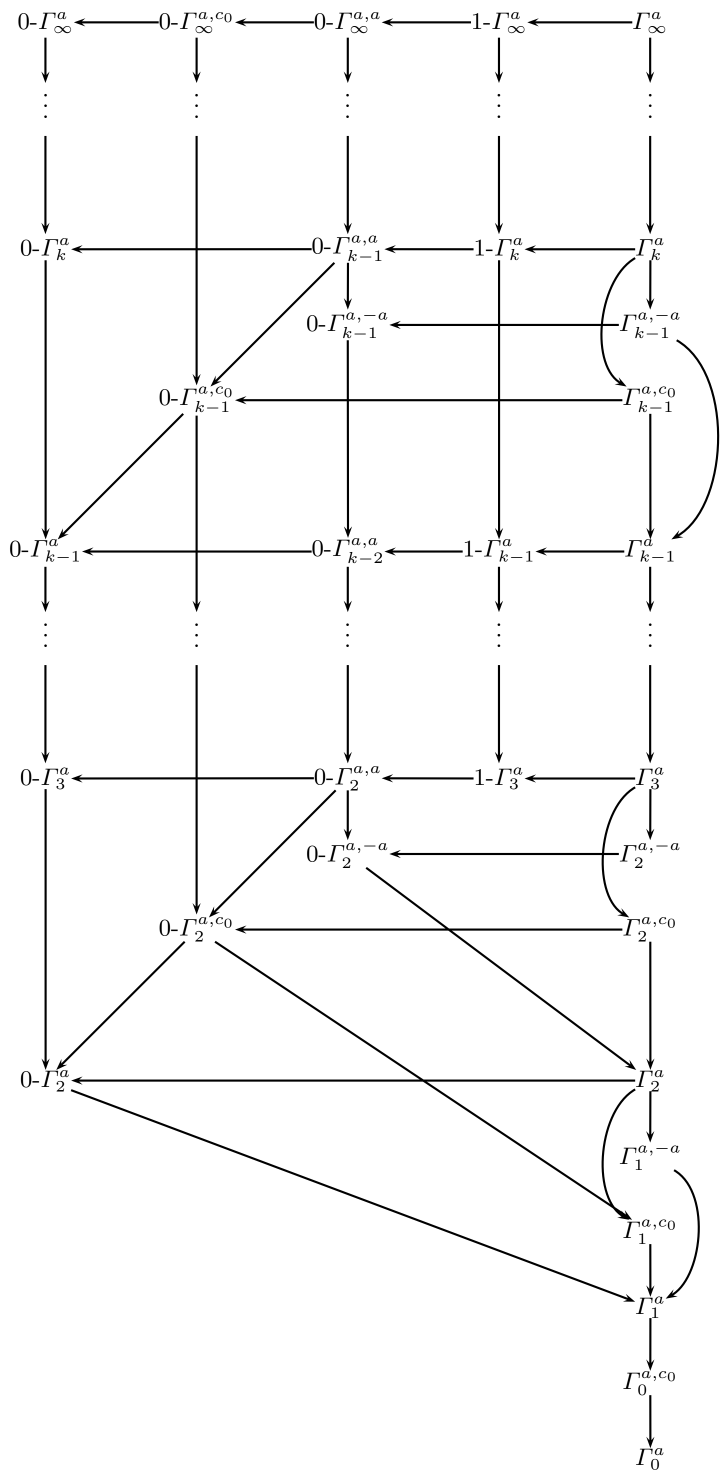

Fig. 3. Game hierarchy II

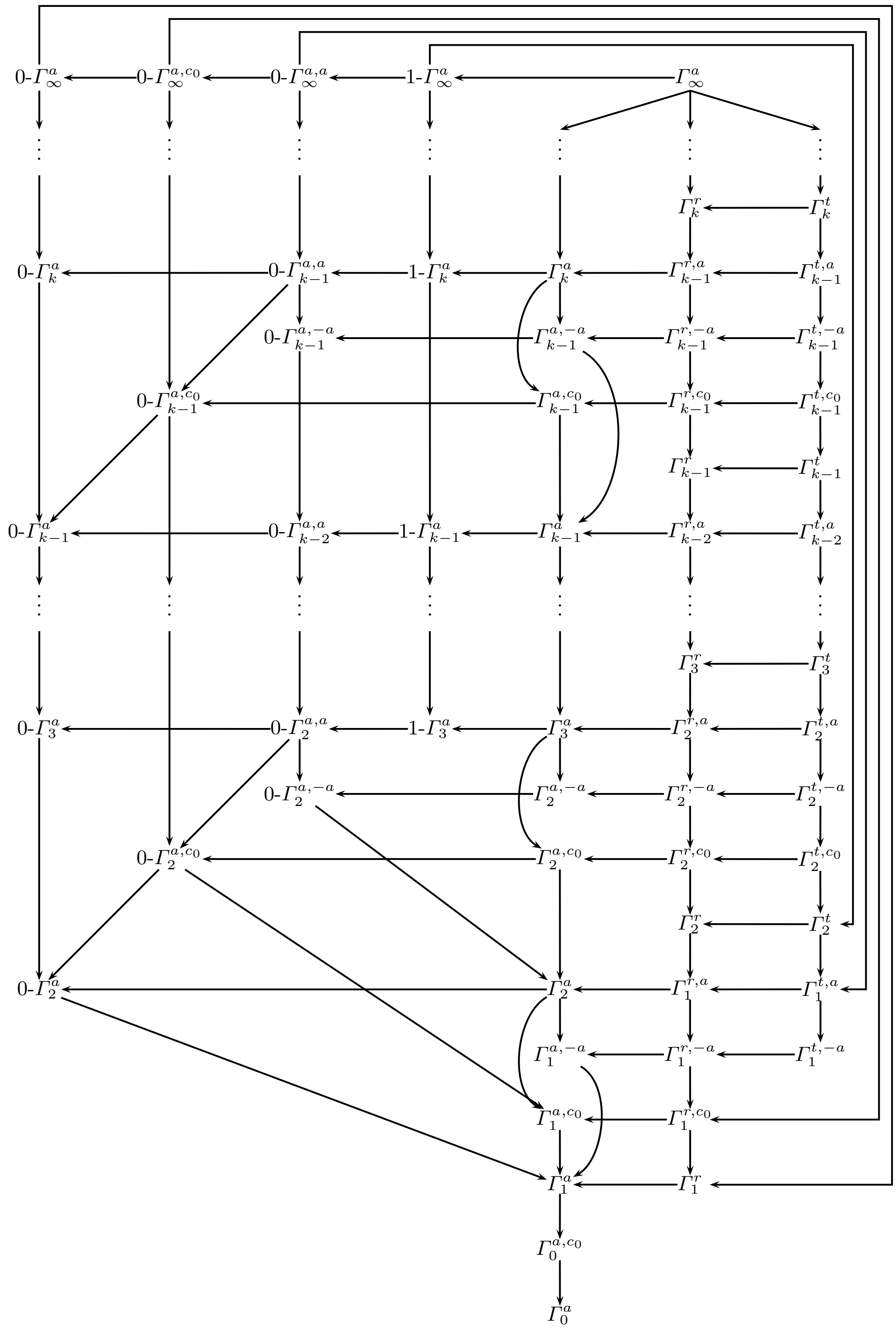

Fig. 4. The whole game hierarchy

4.3 The Whole Game Hierarchy

We now combine game hierarchies I and II into a single hierarchy, as described in Figure 4. Similar to Figure 3, we have not drawn the games with alternations exceeding 1, but our lemmas below cover them.

In the combined game hierarchy, we have the new relations, $\Gamma^t_{n+1} \preceq n\text{-}\Gamma^a_\infty$, $\Gamma^{t,c_0}_{n+1} \preceq n\text{-}\Gamma^{a,c_0}_\infty$, $\Gamma^{t,a}_{n+1} \preceq n\text{-}\Gamma^{a,a}_\infty$ for $n \geq 0$. We give a proof of $\Gamma^t_{n+1} \preceq n\text{-}\Gamma^a_\infty$ in the lemma below; the others can be proven analogously.

Lemma 4. *$\Gamma^t_{n+1} \preceq n\text{-}\Gamma^a_\infty$ for all $n \geq 0$.*

Proof. We prove the statement by induction on n. Assume player II wins n-$\Gamma^a_\infty(P_0, Q_0)$ for some processes P_0 and Q_0.

- $n = 0$. Suppose player I performs $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \dots \xrightarrow{a_l} P_l$ (resp. $Q_0 \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$). Since player II wins 0-$\Gamma^a_\infty(P_0, Q_0)$, she can respond with $Q_0 \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \dots \xrightarrow{a_l} Q_l$ (resp. $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$). Hence, player II wins $\Gamma^t_1(P_0, Q_0)$.
- $n > 0$. From the assumption, player II wins n-$\Gamma^a_\infty(P_0, Q_0)$. In the first round of $\Gamma^t_{n+1}(P_0, Q_0)$, if player I performs $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \dots \xrightarrow{a_l} P_l$ (resp. $Q_0 \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} Q_l$), player II can respond with $Q_0 \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \dots \xrightarrow{a_l} Q_l$ (resp. $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_l} P_l$), such that for each P_i and Q_i, where $1 \leq i \leq l$, player II wins $(n-1)$-$\Gamma^a_\infty(P_i, Q_i)$. By induction, $\Gamma^t_n \preceq (n-1)$-Γ^a_∞, player II wins $\Gamma^t_n(P_i, Q_i)$ for any $1 \leq i \leq l$. Hence, player II wins $\Gamma^t_{n+1}(P_0, Q_0)$. □

We are in a position to state the main result of the paper.

Theorem 3. *(1) In Figure 4, if $\mathcal{G}_1 \to \mathcal{G}_2$ then $\mathcal{G}_1 \succ \mathcal{G}_2$.*
(2) No more relations can be added to the game hierarchy in Figure 4, except for those derived from the transitivity of $\succ$. □

The first statement follows from Theorems 1, 2 and Lemma 4 provided we could show that

$$(*) \qquad \text{In Figure 4, if } \mathcal{G}_1 \to \mathcal{G}_2 \text{ then } \mathcal{G}_2 \not\succeq \mathcal{G}_1.$$

The rest of this section is devoted to proving (*) and the second statement of Theorem 3 by providing counterexamples to prove the invalidities of some relations. For that purpose, it suffices to establish Lemmas 5 to 7 below.

Lemma 5. *For all $k \geq 1$,*

(1) $\Gamma^r_1 \not\preceq \Gamma^a_k$.
(2) $\Gamma^{t,-a}_1 \not\preceq \Gamma^r_k$.

Proof. (1) We define the processes below:

Example 1. $A \stackrel{def}{=} a.A$ and $A_i \stackrel{def}{=} \begin{cases} 0 & \text{if } i = 0 \\ a.A_{i-1} & \text{if } i > 0 \end{cases}$

Consider $\mathit{\Gamma}_k^a(A, A_k)$, in each round player I can only perform action a from one process, and player II can always respond properly, since both A and A_k can perform action a for k times. Then player II wins $\mathit{\Gamma}_k^a(A, A_k)$. But player I wins $\mathit{\Gamma}_1^r(A, A_k)$, she performs an action sequence $t = a^{k+1}$ from A in the first round, player II fails to respond to such sequence from A_k, since the process can only perform action a for k times.

(2) Consider the following processes.

Example 2.

$$P_0 \stackrel{def}{=} b.0, \quad Q_0 \stackrel{def}{=} c.0,$$
$$P_{i+1} \stackrel{def}{=} a.(P_i + d.0) + a.(Q_i + e.0),$$
$$Q_{i+1} \stackrel{def}{=} a.(P_i + e.0) + a.(Q_i + d.0).$$

It is not difficult to prove that player II wins $\mathit{\Gamma}_k^r(P_{k+1}, Q_{k+1})$ by induction on k.

- $k = 1$. This case is easy.
- $k > 1$. We distinguish five sub-cases.

Case 1: Player I performs $P_{k+1} \xrightarrow{a} (P_k + d.0)$. Then player II can perform $Q_{k+1} \xrightarrow{a} (Q_k + d.0)$. By induction player II wins $\mathit{\Gamma}_{k-1}^r(P_k, Q_k)$ and thus she also wins $\mathit{\Gamma}_{k-1}^r(P_k + d.0, Q_k + d.0)$.

Case 2: Player I performs $P_{k+1} \xrightarrow{a} (Q_k + e.0)$. Then player II can perform $Q_{k+1} \xrightarrow{a} (P_k + e.0)$. Similar to the previous case, player II wins $\mathit{\Gamma}_{k-1}^r(Q_k + e.0, P_k + e.0)$.

Case 3: Player I performs $Q_{k+1} \xrightarrow{a} (Q_k + d.0)$. Then player II can perform $P_{k+1} \xrightarrow{a} (P_k + d.0)$. The rest is similar to Case 1.

Case 4: Player I performs $Q_{k+1} \xrightarrow{a} (P_k + e.0)$. Then player II can perform $P_{k+1} \xrightarrow{a} (Q_k + e.0)$. The rest is similar to Case 2.

Case 5: If player I performs $P_{k+1} \xrightarrow{t} P'$ (resp. $Q_{k+1} \xrightarrow{t} Q'$) for some $t \in Act^*$ and $|t| > 1$, player II can always respond with $Q_{k+1} \xrightarrow{t} Q'$ (resp. $P_{k+1} \xrightarrow{t} P'$) such that P' and Q' are isomorphic.

On the other hand, player I wins $\mathit{\Gamma}_1^{t,-a}(P_{k+1}, Q_{k+1})$. A winning strategy is to perform $P_{k+1} \xrightarrow{a} (P_k + d.0) \xrightarrow{a} (P_{k-1} + d.0) \xrightarrow{a} \cdots \xrightarrow{a} (b.0 + d.0)$, where each process passed in the sequence can perform action d and the last process can perform action b. But player II fails to perform such an action sequence from Q_{k+1} and will become stuck in the second round. □

Similar to Lemma 5, the next two lemmas can be proven by providing appropriate counterexamples. See Appendix A for their detailed proofs.

Lemma 6. *For all $k \geq 1$,*

(1) $0\text{-}\mathit{\Gamma}_k^{a,c_0} \not\preceq \mathit{\Gamma}_k^t$.
(2) $0\text{-}\mathit{\Gamma}_k^{a,-a} \not\preceq \mathit{\Gamma}_k^{t,c_0}$.
(3) $0\text{-}\mathit{\Gamma}_{k+1}^a \not\preceq \mathit{\Gamma}_k^{t,-a}$.

(4) $0\text{-}\Gamma_k^{a,c_0} \not\preceq \Gamma_k^{a,-a}$. □

Lemma 7. *For all $n \geq 0$,*

(1) $\Gamma_{n+1}^{a,c_0} \not\preceq n\text{-}\Gamma_\infty^{a}$.
(2) $\Gamma_{n+1}^{a,-a} \not\preceq n\text{-}\Gamma_\infty^{a,c_0}$.
(3) $(n+1)\text{-}\Gamma_{n+3}^{a} \not\preceq n\text{-}\Gamma_\infty^{a,a}$. □

5 Characterizing Process Equivalences

In this section we revisit some important process equivalences[1] in the linear/branching time hierarchy showed in Figure 1.

Definition 9. *Given a game $\mathcal{G}$ and a process equivalence $\approx$, we say $\approx$ is characterized by $\mathcal{G}$ if for any processes P, Q, it holds that $P \approx Q$ iff player II wins $\mathcal{G}(P, Q)$.*

Theorem 4. *(1) Trace equivalence is characterized by Γ_1^{r}.*
(2) Completed trace equivalence is characterized by Γ_1^{r,c_0}.
(3) Failures equivalence is characterized by $\Gamma_1^{r,-a}$.
(4) Failure trace equivalence is characterized by $\Gamma_1^{t,-a}$.
(5) Ready trace equivalence is characterized by $\Gamma_1^{t,a}$.
(6) Readiness equivalence is characterized by $\Gamma_1^{r,a}$.
(7) Possible-futures equivalence is characterized by Γ_2^{r}.

Proof. We only prove (5) and the others can be proven analogously. Suppose P and Q are ready trace equivalent, written $P \sim_{RT} Q$, we prove that player II wins $\Gamma_1^{t,a}(P, Q)$. In the first round, if player I performs some trace t from P or Q, then player II considers t as a ready trace, since she has full knowledge of player I's move. Clearly, in the second round player I cannot make player II stuck. Conversely, suppose player II wins $\Gamma_1^{t,a}(P, Q)$. It is apparent that P, Q have the same ready traces, and then $P \sim_{RT} Q$. □

Similar to the approximation of bisimilarity (cf. Definition 3), we can define similarity $\sim_S$, completed similarity $\sim_{CS}$, ready similarity $\sim_{RS}$, 2-nested similarity $\sim_{2S}$, and their approximants. We write $\sim_k^{*}$, where $k \geq 0$, for the approximants of $\sim^{*}$.

Lemma 8. *For all $k \geq 0$,*

(1) Γ_k^{a} *characterizes* $\sim_k^{a}$.
(2) Γ_k^{r} *characterizes* $\sim_k^{r}$.
(3) $0\text{-}\Gamma_k^{a}$ *characterizes* $\sim_k^{S}$.
(4) $0\text{-}\Gamma_k^{a,c_0}$ *characterizes* $\sim_k^{CS}$.
(5) $0\text{-}\Gamma_k^{a,a}$ *characterizes* $\sim_k^{RS}$.

[1] Due to lack of space we do not list the definitions of those process equivalences; they can be found in [10].

(6) $1\text{-}\Gamma_k^a$ *characterizes* $\sim_k^{2S}$.

Proof. All the statements can be easily proven by induction on k, so we omit them. □

Since we are dealing with image-finite LTSs, the next theorem follows from Lemma 8.

Theorem 5. *(1) Simulation equivalence is characterized by* $0\text{-}\Gamma_\infty^a$.
(2) Completed simulation equivalence is characterized by $0\text{-}\Gamma_\infty^{a;c_0}$.
(3) Ready simulation equivalence is characterized by $0\text{-}\Gamma_\infty^{a;a}$.
(4) 2-nested simulation equivalence is characterized by $1\text{-}\Gamma_\infty^a$. □

Furthermore, new equivalences can be defined using the games in Figure 4. For example, we can define a new equivalence using game Γ_2^t which is stronger than possible-futures equivalence and ready trace equivalence, but weaker than 2-nested simulation equivalence. In addition, from the game hierarchy, we learn the relationship between the approximants of bisimilarity, similarity, completed similarity etc. For example, possible-futures equivalence is stronger than $\sim_2^S$, but is incomparable with $\sim_3^S$. Hence, the game hierarchy is interesting in that it offers an intuitive way of comparing various process equivalences.

6 Concluding Remarks

We have presented a hierarchy of games that allows us to compare process equivalences systematically in a game-theoretic way by characterizing process equivalences as games. The hierarchy not only provides us with a refined analysis of process equivalences, but also offers a guidance to defining interesting new process equivalences.

The work closely related to ours is [5] which provides a Stirling class of games to characterize various process equivalences. The methodology adopted in the current work is different because we examine in a systematic way the theory of games that could characterize typical equivalences in the process equivalence hierarchy.

Paying our attention to the analysis of process equivalences is for the purpose of studying the complexity of equivalence checking. We know that model checking can be considered in a game-theoretic way [7], but the complexity depends on particular models. Similar phenomena exist for equivalence checking. However, equivalence checking is much harder than model checking, and sometimes it cannot be done in similar ways. Further investigation in this respect would be interesting.

Acknowledgement We thank Enshao Shen, Yunfeng Tao and Chaodong He for interesting discussions. We also thank the anonymous referees for their constructive comments.

References

[1] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[2] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[3] A. Kucera and J. Esparza. A logical viewpoint on process-algebraic quotients. In *Proceedings of the 8th Annual Conference of the European Association for Computer Science Logic*, volume 1683 of *Lecture Notes in Computer Science*, pages 499–514. Springer, 1999.

[4] R. Milner. *Communication and concurrency*. Prentice-Hall, Inc., 1989.

[5] S. K. Shukla, H. B. H. III, and D. J. Rosenkrantz. HORNSAT, model checking, verification and games (extended abstract). In *Proceedings of the 8th International Conference on Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 99–110. Springer, 1996.

[6] C. Stirling. Modal and temporal logics for processes. *Notes for Summer School in Logic Methods in Concurrency*, 1993.

[7] C. Stirling. Local model checking games. In *Proceedings of the 6th International Conference on Concurrency Theory*, volume 962 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 1995.

[8] C. Stirling. Games and modal mu-calculus. In *Proceedings of the 2nd International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, volume 1055 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 1996.

[9] W. Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science. In *Proceedings of the 3rd International Joint Conference CAAP/FASE on Theory and Practice of Software Development*, volume 668 of *Lecture Notes in Computer Science*, pages 559–568. Springer, 1993.

[10] R. J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *Proceedings of the 1st International Conference on Concurrency Theory*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer, 1990.

A Proofs from Section 4.3

Proof of Lemma 6.

Proof. (1) We define the following processes:

Example 3.

$$\begin{aligned} &P_1 \stackrel{def}{=} a.b.0 + a.0, \quad Q_1 \stackrel{def}{=} a.b.0, \\ &P_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i) + r(P_i) + Q_i), \\ &Q_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i)) + a.(r(P_i) + Q_i). \end{aligned}$$

where r is an injective renaming function that maps actions in P_k, Q_k to some fresh actions.

We prove that player II wins $\Gamma_k^t(P_k, Q_k)$.

- $k = 1$. This case is trivial.

- $k > 1$. In the first round of $\mathit{\Gamma}_k^t(P_k, Q_k)$, we have the following sub-cases:

  Case 1: Player I performs $P_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}) + r(P_{k-1}) + Q_{k-1})$. By induction, whenever player II responds with $Q_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}))$ or $Q_k \xrightarrow{a} (r(P_{k-1}) + Q_{k-1})$, she can always continue the game for at least $(k-1)$ rounds.

  Case 2: Player I performs $Q_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}))$ and player II performs $P_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}) + r(P_{k-1}) + Q_{k-1})$. The rest is similar to Case 1.

  Case 3: Player I performs $Q_k \xrightarrow{a} (r(P_{k-1}) + Q_{k-1})$ and player II performs $P_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}) + r(P_{k-1}) + Q_{k-1})$. The rest is similar to Case 1.

  Case 4: Player I performs $P_k \xrightarrow{t} P'$ (resp. $Q_k \xrightarrow{t} Q'$), where $t \in \mathit{Act}^*$, player II can respond with $Q_k \xrightarrow{t} Q'$ (resp. $P_k \xrightarrow{t} P'$) such that P' and Q' are isomorphic. By induction hypothesis, whenever player I chooses some configuration for the next round, player II can always continue the game for at least $(k-1)$ rounds.

The fact that player I wins $0\text{-}\mathit{\Gamma}_k^{a,c_0}(P_k, Q_k)$ can be proven similarly.

- $k = 1$. Player I performs $P_1 \xrightarrow{a} 0$ and 0 is colored $\mathcal{C}_0$, but player II fails to make a proper response.
- $k > 1$. Player I performs $P_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}) + r(P_{k-1}) + Q_{k-1})$. Then player II has two ways to respond.

  Case 1: Player II responds with $Q_k \xrightarrow{a} (P_{k-1} + r(Q_{k-1}))$. Then in the second round the configuration is $(P_{k-1} + r(Q_{k-1}) + r(P_{k-1}) + Q_{k-1}, P_{k-1} + r(Q_{k-1}))$. If player I performs some action from $r(P_{k-1})$, player II can respond with the same action from $r(Q_{k-1})$. By induction, player I wins $0\text{-}\mathit{\Gamma}_{k-1}^{a,c_0}(P_{k-1}, Q_{k-1})$ by performing some action from P_{k-1} in the first round, and obviously it is also the case for $0\text{-}\mathit{\Gamma}_k^{a,c_0}(r(P_{k-1}), r(Q_{k-1}))$. Hence, player I can make player II stuck in the k-th round and does not need any alternation.

  Case 2: Player II responds with $Q_k \xrightarrow{a} (r(P_{k-1}) + Q_{k-1})$. The rest is similar to Case 1.

(2) Define the following processes:

Example 4.

$$P_1 \stackrel{def}{=} a.b.0 + a.c.0, \quad Q_1 \stackrel{def}{=} a.(b.0 + c.0),$$
$$P_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i) + r(P_i) + Q_i),$$
$$Q_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i)) + a.(r(P_i) + Q_i).$$

Player II wins $\mathit{\Gamma}_k^{t,c_0}(P_k, Q_k)$, but player I wins $0\text{-}\mathit{\Gamma}_k^{a,-a}(P_k, Q_k)$. The proofs are similar to part (1).

(3) Define the following processes:

Example 5.

$$P_1 \stackrel{def}{=} a.(b.0 + c.0) + a.b.0 + a.c.0, \quad Q_1 \stackrel{def}{=} a.b.0 + a.c.0,$$
$$P_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i) + r(P_i) + Q_i),$$
$$Q_{i+1} \stackrel{def}{=} a.(P_i + r(Q_i)) + a.(r(P_i) + Q_i).$$

Player II wins $\Gamma_k^{t,-a}(P_k, Q_k)$, but player I wins 0-$\Gamma_{k+1}^{a}(P_k, Q_k)$. The proofs are similar to part (1).

(4) We give the following example.

Example 6.

$$P_0 \stackrel{def}{=} b.0,$$
$$P_{i+1} \stackrel{def}{=} a.P_i + A_{i+1},$$

where A_{i+1} is defined in Example 1.

The statement that player II wins $\Gamma_k^{a,-a}(P_k, A_k)$ can be proven by induction on k, as we did in the proofs of previous lemmas. We observe that player I wins 0-$\Gamma_k^{a,c_0}(P_k, A_k)$, since she can perform $P_k \xrightarrow{a} P_{k-1} \xrightarrow{a} P_{k-2} \ldots P_1 \xrightarrow{a} P_0$ in the first k rounds, and player II has to respond with $A_k \xrightarrow{a} A_{k-1} \xrightarrow{a} A_{k-2} \ldots A_1 \xrightarrow{a} A_0$. Since P_0 is not a deadlock process, it is not colored $\mathcal{C}_0$, but A_0 is colored $\mathcal{C}_0$. □

Proof of Lemma 7.

Proof. (1) Consider the following processes:

Example 7.

$$P_0 \stackrel{def}{=} b.0, \quad Q_0 \stackrel{def}{=} 0,$$
$$P_{i+1} \stackrel{def}{=} a.(P_i + Q_i),$$
$$Q_{i+1} \stackrel{def}{=} a.P_i + a.Q_i.$$

Player II wins n-$\Gamma_k^{a}(P_{n+1}, Q_{n+1})$ for all $k \geq 0$, since player I needs at least $(n{+}1)$ alternations to distinguish P_{n+1} and Q_{n+1} without coloring. It follows that player I wins n-$\Gamma_\infty^{a}(P_{n+1}, Q_{n+1})$. But player I wins $\Gamma_{n+1}^{a,c_0}(P_{n+1}, Q_{n+1})$ because she can make player II stuck in the last round.

(2) We define the following processes:

Example 8.

$$P_1 \stackrel{def}{=} a.(b.0 + c.0) + a.c.0,$$
$$Q_1 \stackrel{def}{=} a.(b.0 + c.0),$$
$$P_{i+1} \stackrel{def}{=} a.(P_i + Q_i),$$
$$P_{i+1} \stackrel{def}{=} a.P_i + a.Q_i.$$

Player II wins n-$\Gamma_\infty^{a,c_0}(P_{n+1}, Q_{n+1})$, since $(n+1)$ alternations are needed for player I in the first $(n+2)$ rounds to distinguish the two processes. Player I wins $\Gamma_{n+1}^{a,-a}(P_{n+1}, Q_{n+1})$ because she can make $(n+1)$ alternations in the $(n+2)$ rounds of the game.

(3) Consider following processes:

Example 9.

$$P_0 \stackrel{def}{=} a.(a.b.0 + a.0),$$
$$Q_0 \stackrel{def}{=} a.(a.b.0 + a.0) + a.a.0,$$
$$P_{i+1} \stackrel{def}{=} a.Q_i,$$
$$Q_{i+1} \stackrel{def}{=} a.P_i + a.Q_i.$$

Player II wins n-$\Gamma_\infty^{a,a}(P_n, Q_n)$, since in the first $(n+2)$ rounds, player I need $(n+1)$ alternations in order to prevent player II from making two processes in the configuration for the next round isomorphic, and she also needs one more round, but no more alternation, to make player II stuck. We also showed a winning strategy for player I in $(n+1)$-$\Gamma_{n+3}^{a}(P_n, Q_n)$. □