

On Parity Game Preorders and the Logic of Matching Plays

M.W. Gazda and T.A.C. Willemse^(✉)

Eindhoven University of Technology, Eindhoven, The Netherlands
t.a.c.willemse@tue.nl

Abstract. Parity games can be used to solve satisfiability, verification and controller synthesis problems. As part of an effort to better understand their nature, or the nature of the problems they solve, preorders on parity games have been studied. Defining these relations, and in particular proving their transitivity, has proven quite difficult on occasion. We propose a uniform way of lifting certain preorders on Kripke structures to parity games and study the resulting preorders. We explore their relation with parity game preorders from the literature and we study new relations. Finally, we investigate whether these preorders can also be obtained via modal characterisations of the preorders.

1 Introduction

Parity games [6, 15] are two player games played on a directed graph. These games are interesting as they underpin, *e.g.* solutions to verification, satisfiability and synthesis problems, see [2, 7] and they appear as solution to fundamental problems such as complementing tree automata [6]. The problem of solving a parity game (computing the set of vertices won by each player) is one of those rare problems that are in $\text{NP} \cap \text{coNP}$, and for which no polynomial time algorithm has yet been found.

In an effort to increase the general understanding of the parity game solving problem or of those problems mapped to parity game solving, preorders on parity games have been studied on various occasions and for different purposes. For instance, in [13], Namjoshi investigated simulation in the context of abstraction using a variant of parity games called *model checking games*, leading to a framework that was *complete* (in the sense of having finite abstract objects) for the μ -calculus; in [8], Fritz and Wilke [8] defined and studied *delayed simulation*, an adaptation of simulation; Cranen *et al.* [3, 4] studied variations of stuttering bisimulation for parity games; Kissig and Venema [12] defined *basic game bisimulation* for studying complementation. Dawar and Grädel [5] defined yet two other forms of bisimulation on parity games by viewing these as relational structures; their purpose is to analyse the descriptive complexity of parity games.

For the most part, the preorders on parity games are inspired by similar relations on computational models such as Kripke structures or Labelled Transition

Systems. However, there seems to be no systematic method by which the aforementioned parity game relation have been obtained from a relation on a computational model, as testified by the existence of several variations of bisimulation on parity games. Moreover, defining new relations on parity games, showing transitivity and proving that they approximate the winning set of some player can be quite cumbersome; for instance, proving transitivity of delayed simulation required analysing 24 different cases, see [8], and in [4], the proof that governed stuttering bisimilarity is an equivalence relation is technically involved, requiring a step-wise rephrasing of the definition and intricate arguments.

The contributions of this paper are as follows. We propose a novel, more generic method for obtaining a parity game relation from a relation on a computational model. It is based on the notion of matching paths, which we lift naturally to matching plays. In this approach, we can lift any preorder on a computational model that can be specified using the matching paths to a corresponding preorder in the parity game setting. Moreover, we identify conditions that guarantee that the resulting relation is a preorder and that it approximates the winning set of a particular player.

We exemplify our approach using a number of well-known and some less-known relations on Kripke structures and show that some of the thus obtained relations coincide with existing parity game relations. Finally, for all the relations we study in detail, we provide logical characterisations, by identifying sound and complete fragments of an alternating-time temporal logic for the respective relations. The logical characterisations reveal interesting differences between the behavioural relations and the induced parity game relations.

Structure. In Sect. 2, we recall the basics of parity games. Then, in Sect. 3, we introduce our generic scheme for obtaining parity game relations. In Sect. 4, we show how this theory can be applied to recover existing and define new parity game preorders; in Sect. 5, we provide modal characterisations of these relations. We finish with conclusions in Sect. 6. Proofs for all results can be found in [10].

2 Preliminaries

A parity game is an infinite duration game, played by players *odd*, denoted by \square and *even*, denoted by \diamond , on a directed, finite graph.

Definition 1. *A parity game is a tuple $\langle V, E, \Omega, (V_\diamond, V_\square) \rangle$, where*

- V is a set of vertices, partitioned in a set V_\diamond of vertices owned by player \diamond , and a set of vertices V_\square owned by player \square ,
- $E \subseteq V \times V$ is a total edge relation, i.e. for all v , $(v, w) \in E$ for some w ,
- $\Omega: V \rightarrow \mathbb{N}$ is a priority function that assigns priorities to vertices.

We depict parity games as graphs in which diamond-shaped vertices represent vertices owned by player \diamond , box-shaped vertices represent vertices owned by player \square and priorities, associated with vertices, are written inside vertices; see Fig. 1(a) and (b) for examples.

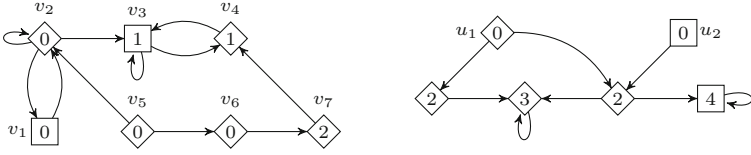


Fig. 1. (a) Left: A parity game with three different priorities. (b) Right: A parity game with four different priorities.

We use the following notational conventions: we write $v \rightarrow w$ instead of $(v, w) \in E$, and we write v^\bullet for the set $\{w \in V \mid v \rightarrow w\}$. Henceforth, \circ denotes an arbitrary player. We write $\bar{\circ}$ for \circ 's opponent; *i.e.* $\bar{\diamond} = \square$ and $\bar{\square} = \diamond$. Finally, given the set of vertices V , the subset of vertices of V with priority n is denoted V_n : we have $V_n = \{v \in V \mid \Omega(v) = n\}$.

A play starts by placing a token on some vertex $v \in V$. Players \diamond and \square move the token indefinitely according to a single simple rule: if the token is on some vertex that belongs to player \circ , that player gets to move the token to an adjacent vertex. The *parity* of the highest priority that occurs infinitely often on a play defines the *winner* of the play: player \diamond wins if, and only if this priority is even. This is known as the *parity condition*.

A strategy for player \circ is a partial function $\sigma: V^* \rightarrow V$ satisfying that for all sequences of vertices $u_1 \dots u_n \in V^*$ on which σ is defined, both $u_n \in V_\circ$ and $\sigma(u_1 \dots u_n) \in u_n^\bullet$. The set of all strategies for player \circ is denoted \mathbb{S}_\circ^* .

Let $\pi = v_1 v_2 v_3 \dots$, with $v_1 = v$ be a play starting in v . We denote the i -th vertex on π by π_i ; that is, $\pi_i = v_i$. Play π is *consistent* with a strategy $\sigma \in \mathbb{S}_\circ^*$ if all prefixes $v_1 \dots v_n$ of π for which $\sigma(v_1 \dots v_n)$ is defined, we have $v_{n+1} = \sigma(v_1 \dots v_n)$. The set of plays consistent with strategy σ , starting in v is denoted $\text{Plays}(\sigma, v)$; we sometimes refer to this set as the set of σ -plays. A strategy σ is winning for player \circ from vertex v iff all plays consistent with σ are won by \circ . Vertex v is won by player \circ whenever she has a winning strategy for vertex v .

Example 1. In the parity game depicted in Fig. 1(a), v_1, v_2, v_5 are won by player \diamond , and player \square wins the remaining vertices.

Let $C, D \subseteq V$ be sets of vertices. We generalise the one-step reachability relation E to (forced) reachability, where reachability is confined to a set of vertices. Let $v \in V$.

$$v_{\circ} \rightarrow D = \begin{cases} v^\bullet \cap D \neq \emptyset & \text{if } v \in V_\circ \\ v^\bullet \subseteq D & \text{otherwise} \end{cases}$$

$$v_{\circ} \mapsto_C D = \exists \sigma \in \mathbb{S}_\circ^* : \forall \pi \in \text{Plays}(\sigma, v) : \exists k : \pi_k \in D \wedge \forall j < k : \pi_j \in C$$

$$v_{\circ} \mapsto_C = \exists \sigma \in \mathbb{S}_\circ^* : \forall \pi \in \text{Plays}(\sigma, v) : \forall k : \pi_k \in C$$

Finally, let $R \subseteq V \times V$ be a relation on V . The set of vertices *below* some vertex v , denoted Rv , is the set $\{w \in V \mid w R v\}$; the set of vertices *above* v , denoted

vR , is defined as $\{w \in V \mid v R w\}$. Note that in the special case that R is an equivalence relation, we have $Rv = vR$. For a set $U \subseteq V$, we write UR for the set $\bigcup_{v \in U} vR$; likewise for RU , and if R is an equivalence relation, we write V/R for the set of equivalence classes (quotient set) of R .

3 Inducing Parity Game Preorders and Equivalences

In the past, behavioural relations for transition systems, such as the simulation preorder and bisimulation, have been ported to parity games, see [3–5, 8, 9, 12]. These efforts, however, did not appear to have followed a general guiding principle. In this section, we propose a scheme for lifting particular types of behavioural preorders for Kripke structures to parity game preorders. More specifically, the Kripke structure preorders and equivalences we consider are those that can be phrased in terms of *matching paths*.

Let $K = \langle S, T, \text{AP}, \mathcal{L} \rangle$ be an arbitrary Kripke structure, where S is a (possibly infinite) set of states, $T \subseteq S \times S$ is a total transition relation, AP is a set of atomic propositions and $\mathcal{L}: S \rightarrow \mathcal{P}(\text{AP})$ is a state labelling function. A *path* through K , starting in some state $s_1 \in S$, is an infinite sequence of states $s_1 s_2 s_3 \dots$ for which $(s_i, s_{i+1}) \in T$ for all i , and $\text{Paths}(s)$ denotes the set of paths starting in s .

Let, within the context of some Kripke structure K , $\text{Rel}(R)$ denote a predicate on relations $R \subseteq S \times S$ on K 's states. Think, for instance, of the predicate that a relation R is a simulation relation. For given predicate Rel and relation R , a *matching predicate* is a predicate $\text{Rel-match}_R^{\mathcal{L}}(\pi_t, \pi_s)$ on *paths* π_t and π_s in K , and K 's labelling \mathcal{L} and the relation R on K 's states satisfying:

$$\text{Rel}(R) \text{ iff } \forall (s, t) \in R : \forall \pi_s \in \text{Paths}(s) : \exists \pi_t \in \text{Paths}(t) : \text{Rel-match}_R^{\mathcal{L}}(\pi_t, \pi_s)$$

Note that matching predicates do *not* use K 's transition relation T .

Example 2. A typical instance of Rel is the simulation predicate Sim : for $R \subseteq S \times S$, $\text{Sim}(R)$ holds iff for all $(s, t) \in R$ we have $\mathcal{L}(s) = \mathcal{L}(t)$ and for any $s' \in S$ with $(s, s') \in T$, there is a $t' \in S$ for which $(t, t') \in T$ and $(s', t') \in R$. An associated matching predicate $\text{Sim-match}_R^{\mathcal{L}}(\pi, \pi')$ is $\forall i : \mathcal{L}(\pi'_i) = \mathcal{L}(\pi_i) \wedge \pi'_i R \pi_i$.

In case a matching predicate is to be interpreted on a parity game, we use the priority function Ω as the state labelling function, writing $\text{Rel-match}_R^{\Omega}$. A matching predicate $\text{Rel-match}_R^{\mathcal{L}}$ is *monotonic* if $\text{Rel-match}_R^{\mathcal{L}}(\pi_t, \pi_s)$ implies $\text{Rel-match}_{R'}^{\mathcal{L}}(\pi_t, \pi_s)$ for all $R \subseteq R'$.

Definition 2. Let $G = \langle V, E, \Omega, (V_{\diamond}, V_{\square}) \rangle$ be a parity game and let $\text{Rel-match}_R^{\mathcal{L}}$ be a matching predicate for a predicate Rel on Kripke structure relations. A relation $R \subseteq V \times V$ is a parity game *Rel*-relation whenever $v R w$ implies that for all strategies $\sigma_v \in \mathbb{S}_{\diamond}^*$, there is a strategy $\sigma_w \in \mathbb{S}_{\square}^*$ such that

$$\forall \pi_w \in \text{Plays}(\sigma_w, w) : \exists \pi_v \in \text{Plays}(\sigma_v, v) : \text{Rel-match}_R^{\Omega}(\pi_w, \pi_v)$$

We write $v \sqsubseteq_{\text{Rel}} w$ iff for some parity game *Rel*-relation R we have $v R w$.

Example 3. A relation $R \subseteq V \times V$ is a *parity game Sim-relation* whenever $v R w$ implies that for all strategies $\sigma_v \in \mathbb{S}_\diamond^*$, there is a strategy $\sigma_w \in \mathbb{S}_\diamond^*$ such that

$$\forall \pi_w \in \text{Plays}(\sigma_w, w) : \exists \pi_v \in \text{Plays}(\sigma_v, v) : \forall i : \Omega(\pi_{v,i}) = \Omega(\pi_{w,i}) \wedge \pi_{v,i} R \pi_{w,i}$$

In Sects. 4.1 and 4.2, we study further instances of Definition 2, showing that the theory of this section can be used to recover existing parity game relations (Sect. 4.1) and how it can be used to obtain new parity game relations (Sect. 4.2). For the remainder of this section, we focus on establishing under which conditions one can prove the resulting parity game relations are preorders and when they can be used to approximate the winning partition for player \diamond . The theorem below shows that an induced parity game Rel-relation is a preorder whenever a simple monotonicity criterion for the matching predicate holds.

Theorem 1. *Assume that for all preorders R for which Rel holds, $\text{Rel-match}_R^{\mathcal{L}}$ is a preorder, too. If $\text{Rel-match}^{\mathcal{L}}$ is monotonic, then \sqsubseteq_{Rel} is a preorder.*

Under similar conditions, one can prove that \sqsubseteq_{Rel} is an equivalence relation. The next theorem states that we can conclude that the parity game relations *under-approximate* the winning partition for player \diamond from a simple condition on the matching predicate.

Theorem 2. *Let R be a parity game Rel-relation. Assume $v R w$ and suppose that for all π_v, π_w , if $\text{Rel-match}_R^{\mathcal{Q}}(\pi_w, \pi_v)$ and π_v is won by \diamond then so is π_w . If v is won by \diamond , then w is won by \diamond .*

4 Applications

We first illustrate how the theory we developed in the previous section can be put to use to recover preorders and equivalences already present in the literature. More specifically, we show that *governed simulation* [11], also known as *direct simulation* [8,9], *governed bisimulation* [11] and *governed stuttering bisimulation* [4] are all instances of our general theory. We then proceed to show that we can also obtain relations that did not appear in the literature.

4.1 Existing Parity Game Relations

Consider the Kripke structure Rel predicates for simulation, bisimulation and stuttering bisimulation (aka stuttering equivalence), listed in Table 1. For lack of space, we refrain from giving the standard definition of these predicates, given that these can be found in most standard textbooks, and since these predicates are essentially also defined via their matching predicates next to them. Note that also the latter can be found in the literature (although less commonly).

Using the Rel-predicates of Table 1 and Definition 2, we immediately obtain parity game simulation, parity game bisimulation and parity game stuttering bisimulation. One can check with little effort that the matching predicates for simulation, bisimulation and stuttering bisimulation meet the conditions of Theorems 1 and 2. As a result, we can claim the following:

Table 1. Matching predicates for well-known behavioural relations; R is a relation on states, \mathcal{L} is a state labelling function, and π, π' are infinite sequences of states

Rel	$\text{Rel-match}_R^{\mathcal{L}}(\pi, \pi')$
Simulation	For all i , $\mathcal{L}(\pi'_i) = \mathcal{L}(\pi_i)$ and $\pi'_i R \pi_i$.
Bisimulation	For all i , $\mathcal{L}(\pi'_i) = \mathcal{L}(\pi_i)$, $\pi'_i R \pi_i$ and $\pi_i R \pi'_i$.
Stuttering bisimulation	There is a non-decreasing, unbounded function $f : \omega \rightarrow \omega$ with $f(1) = 1$ such that for all i and all $j \in [f(i), f(i + 1))$, $\mathcal{L}(\pi'_i) = \mathcal{L}(\pi_j)$, $\pi'_i R \pi_j$ and $\pi_j R \pi'_i$.

Proposition 1. Relation $\sqsubseteq_{\text{simulation}}$ is a preorder and relations $\sqsubseteq_{\text{bisimulation}}$ and $\sqsubseteq_{\text{stuttering bisimulation}}$ are equivalences. Moreover, all three relations under-approximate the winning set for player \diamond .

Definition 3. A relation R is a governed simulation iff for all $v R w$:

1. $\Omega(v) = \Omega(w)$,
2. if $v \in V_{\diamond}$, then for each $v \rightarrow v'$ we have $w_{\diamond} \rightarrow v'R$,
3. if $v \in V_{\square}$, then $w_{\diamond} \rightarrow v^{\bullet}R$.

We write $v \leq w$ iff there is a governed simulation R such that $v R w$. A relation $R \subseteq V \times V$ is a governed bisimulation if both R and R^{-1} are governed simulations. We write $v \Leftrightarrow w$ iff for some governed bisimulation R , $v R w$.

The example below illustrates governed simulation and governed bisimulation.

Example 4. Consider the parity game of Fig. 1(a). We have $v_3 \Leftrightarrow v_4$, because, even though both vertices belong to different players, neither player can force play to vertices of different priorities. On the other hand, $v_1 \Leftrightarrow v_2$ does not hold.

The theorem below confirms that governed similarity and governed bisimilarity coincide with the preorder and equivalence induced by Definition 2 using the simulation and bisimulation matching predicates of Table 1.

Theorem 3. We have $\leq = \sqsubseteq_{\text{simulation}}$ and $\Leftrightarrow = \sqsubseteq_{\text{bisimulation}}$.

Next, we focus on the notion of governed stuttering bisimulation [4].

Definition 4. An equivalence relation $R \subseteq V \times V$ is a governed stuttering bisimulation iff $v R w$ then

1. $\Omega(v) = \Omega(w)$,
2. for any $v \rightarrow \mathcal{C}$ with $\mathcal{C} \in V/R \setminus \{vR\}$ and $v \in V_{\diamond}$, then $w_{\diamond} \mapsto_v R \mathcal{C}$,
3. for any player \circ , we have $v_{\circ} \mapsto_v R$ iff $w_{\circ} \mapsto_v R$.

We write $v \Leftrightarrow_s w$ iff there is a governed stuttering bisimulation R such that $v R w$.

Example 5. Reconsider the parity game from Fig. 1(a). Observe that we did not have $v_1 \leftrightarrow v_2$. However, we do have $v_1 \leftrightarrow_s v_2$: player \diamond is capable of enforcing divergent plays (plays that only pass through vertices with the same priority), and it is capable of enforcing plays to reach vertices with priority 1. \square

Governed stuttering bisimulation coincides with parity game stuttering bisimulation; this confirms that governed stuttering bisimulation naturally generalises Kripke structure stuttering bisimulation to the parity game setting.

Theorem 4. *We have $\leftrightarrow_s = \sqsubseteq_{\text{stuttering bisimulation}}$.*

4.2 Two New Parity Game Relations

So far, we have illustrated that Definition 2 can be used to recover parity game relations from the literature, and that Theorems 1 and 2 are instrumental in establishing that the resulting parity game relations are preorders and that they approximate the winning set for player \diamond . In this section, we show that Definition 2 immediately gives us definitions for *parity game trace inclusion* and *parity game stuttering simulation*; these relations have so far not appeared in the literature. Consider the matching predicates listed in Table 2.

Table 2. Matching predicates for behavioural relations; R is a relation on states, \mathcal{L} is a state labelling function, and π, π' are infinite sequences of states

Rel	Rel-match $^{\mathcal{L}}_R(\pi, \pi')$
Trace inclusion	For all i , $\mathcal{L}(\pi'_i) = \mathcal{L}(\pi_i)$.
Stuttering simulation	There is a non-decreasing, unbounded function $f : \omega \rightarrow \omega$ with $f(1) = 1$ such that for all i and all $j \in [f(i), f(i + 1))$, $\mathcal{L}(\pi'_i) = \mathcal{L}(\pi_j)$ and $\pi'_i R \pi_j$

Stuttering simulation for Kripke structures is *coarser* than simulation: it allows for abstracting from finite computations through states with the same information and computational branching structure. Trace inclusion is even coarser than stuttering simulation. Both stuttering similarity and trace inclusion for Kripke structures are known to be preorders. Theorem 1 allows us to establish that the parity game relations they induce are preorders too. Theorem 2 again allows us to conclude that both preorders under-approximate the winning partition for player \diamond .

Proposition 2. *Relations $\sqsubseteq_{\text{stuttering simulation}}$ and $\sqsubseteq_{\text{trace inclusion}}$ are preorders. Both preorders under-approximate the winning partition for player \diamond .*

We next focus on giving a coinductive definition for parity game stuttering simulation. Apart from providing a deeper understanding of this relation, and understanding how it compares to the ones from the previous section, the coinductive definition gives rise to a polynomial time algorithm for deciding this relations.

Definition 5. *A preorder $R \subseteq V \times V$ is a governed stuttering simulation iff $v R w$ implies:*

1. $\Omega(v) = \Omega(w)$,
2. *left-to-right even transfer*:
 - (a) if $v \in V_\diamond$, then for all $v \rightarrow v_s$ we have $w \diamond \mapsto_{vR} v_s R$,
 - (b) if $v \in V_\square$, then $w \diamond \mapsto_{vR} v^\bullet R$,
3. *right-to-left odd transfer*:
 - (a) if $w \in V_\diamond$, then $v \square \mapsto_{Rw} R w^\bullet$
 - (b) if $w \in V_\square$, then for all $w \rightarrow w_s$, we have $v \square \mapsto_{Rw} R w_s$

We write $v \Leftarrow_s w$ iff there is a governed stuttering bisimulation R such that $v R w$.

For the above coinductive definition of our parity game stuttering simulation relation one can deduce that a symmetric relation that meets its properties is a governed stuttering bisimulation relation—a basic sanity check for the correctness of the definition. The link with governed stuttering bisimulation is, however, not obvious. We have the following theorem relating governed stuttering simulation to parity game stuttering simulation.

Theorem 5. *We have $\leq_s = \sqsubseteq_{\text{stuttering simulation}}$.*

Governed stuttering simulation can be computed in polynomial time; we sketch a naive algorithm based on fixpoint iteration. We start with a trivial relation R that relates all states with the same priorities. Upon every iteration, every pair $(s, t) \in R$ is checked as to whether the conditions of Definition 5 hold; if it is not the case, the pair is removed from R ; thus every iteration a monotonic transformer is applied that after at most a quadratic number of steps will reach a fixpoint. As for checking that s, t and R meet the conditions of Definition 5, the main source of the complexity is in computing the $\circ \mapsto$ relation. The latter can be done in $\mathcal{O}(|V| + |E|)$ time using a modified attractor computation [4] and such a computation is performed for $\mathcal{O}(|V|)$ successors. This means that deciding governed stuttering simulation can be done in at most $\mathcal{O}(|V|^5 \cdot (|V| + |E|))$.

We remark that we did not strive to have optimal running times for deciding the preorders in this section. We leave it for future research to tighten our bounds. For deciding governed stuttering simulation, it may be fruitful to incorporate ideas from [14], which describes, as far as we could trace, the first algorithm for stuttering simulation in the Kripke structure setting.

5 Logical Characterisations of Parity Game Relations

An alternative approach to defining a behavioural preorder is by identifying an appropriate fragment of a modal logic. A natural question is thus whether, given a fragment of a modal logic for Kripke structures that coincides with a given preorder, there is a uniform way of obtaining a fragment of a modal logic for parity games that coincides with the parity game relation. While a logical characterisation of a behavioural relation offers an alternative angle for understanding it, and, as such, is interesting to study in its own right, our results in this section suggest it is unlikely such a uniform method exists.

5.1 A Modal Logic for Parity Games

The logic we consider is called the Alternating-time Hennessy-Milner logic with Until. It is essentially based on the alternating-time temporal logic of [1], but its syntax is inspired by Hennessy-Milner logic for Labelled Transition Systems. Our syntax facilitates characterising all relations we study in this paper by imposing restrictions on our base grammar.

Definition 6. *The Alternating-time Hennessy-Milner logic with Until, henceforth referred to as the logic AHML, is defined as follows:*

$$\phi, \psi := \perp \mid \top \mid \neg\phi \mid \langle n \rangle_{\circ} \phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle n \rangle_{\circ} \psi \mid \phi \langle n \rangle_{\circ}^{\infty} \psi$$

where $n \in \mathbb{N}$ and $\circ \in \{\diamond, \square\}$. The semantics of AHML formulae is defined inductively in the context of a parity game $G = \langle V, E, \Omega, (V_{\diamond}, V_{\square}) \rangle$:

$$\begin{aligned} \llbracket \perp \rrbracket &= \emptyset \\ \llbracket \top \rrbracket &= V \\ \llbracket \neg\phi \rrbracket &= V \setminus \llbracket \phi \rrbracket \\ \llbracket \langle n \rangle_{\circ} \phi \rrbracket &= \langle n \rangle_{\circ} \llbracket \phi \rrbracket \\ \llbracket \phi \wedge \psi \rrbracket &= \llbracket \phi \rrbracket \cap \llbracket \psi \rrbracket \\ \llbracket \phi \vee \psi \rrbracket &= \llbracket \phi \rrbracket \cup \llbracket \psi \rrbracket \\ \llbracket \phi \langle n \rangle_{\circ} \psi \rrbracket &= (V_n \cap \llbracket \psi \rrbracket) \cup \mu V' \subseteq V. (\llbracket \phi \rrbracket \cap (\langle n \rangle_{\circ} \llbracket \psi \rrbracket \cup \langle n \rangle_{\circ} V')) \\ \llbracket \phi \langle n \rangle_{\circ}^{\infty} \psi \rrbracket &= (V_n \cap \llbracket \psi \rrbracket) \cup \nu V' \subseteq V. (\llbracket \phi \rrbracket \cap (\langle n \rangle_{\circ} \llbracket \psi \rrbracket \cup \langle n \rangle_{\circ} V')) \end{aligned}$$

where, for $W \subseteq V$ and $n \in \mathbb{N}$, operator $\langle n \rangle_{\circ} W$ yields the set $\{v \in V_n \mid v \circ \rightarrow W\}$. We write $v \models \phi$ iff $v \in \llbracket \phi \rrbracket$.

Intuitively, $\langle n \rangle_{\circ} \phi$ holds in vertices with priority n (i.e. those from the set V_n) for which \circ can force play to vertices satisfying ϕ . The strong until operator $\phi \langle n \rangle_{\circ} \psi$ holds in vertices with priority n for which \circ can govern the plays through ϕ vertices, ultimately reaching ψ vertices. The weak until operator $\phi \langle n \rangle_{\circ}^{\infty} \psi$ is more or less the same but also holds whenever \circ governs plays through ϕ -invariant vertices. Observe that our use of fixpoints in the semantics is permitted as the associated predicate transformers to which they are applied are monotonic and the set $(2^V, \subseteq)$ is a complete lattice. The example below illustrates typical properties one can express using AHML.

Example 6. Reconsider the parity game depicted in Fig. 1(a). Observe that $v_1 \models \langle 0 \rangle_{\diamond} \top$ and $v_1 \models \langle 0 \rangle_{\square} \top$. We have $v_3 \models \top \langle 1 \rangle_{\diamond}^{\infty} \perp$ and $v_3 \models \top \langle 1 \rangle_{\square}^{\infty} \perp$; we also have $v_2 \models \top \langle 0 \rangle_{\diamond}^{\infty} \perp$ but $v_2 \not\models \top \langle 0 \rangle_{\square}^{\infty} \perp$. Moreover, we have $v_7 \models \langle 1 \rangle_{\diamond} \top \langle 2 \rangle_{\diamond} \neg(\langle 0 \rangle_{\diamond} \top)$ because v_7 has (1) priority 2 as demanded by the until operator and (2) satisfies the goal formula $\neg(\langle 0 \rangle_{\diamond} \top)$. \square

In general, we are interested in comparing the “observations” that we can make in different vertices in a parity game; that is, we wish to compare the set of modal formulae satisfied by different vertices. We formalise observations as follows.

Definition 7. *Let L be a fragment of AHML. We write $\mathcal{O}^L(v)$ to denote the set of formulae $\phi \in L$ for which $v \models \phi$.*

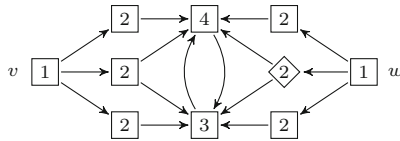
5.2 Characterising Preorders Using AHML

Throughout this section, we assume that $G = \langle V, E, \Omega, (V_\diamond, V_\square) \rangle$ is an arbitrary parity game. In Table 3, we list the sound and complete fragments of the modal logic of Sect. 5.1 and the relations of Sects. 4.1 and 4.2.

Table 3. Parity game preorders and equivalences and their corresponding sound and complete fragments of AHML with their grammars. In these grammars, $n \in \mathbb{N}$

Relation	Fragment	Grammar
\leq	AHML^{\leq}	$\phi, \psi := \top \mid \langle n \rangle_\diamond \phi \mid \phi \wedge \psi \mid \phi \vee \psi$
\leftrightarrow	$\text{AHML}^{\leftrightarrow}$	$\phi, \psi := \top \mid \neg \phi \mid \langle n \rangle_\diamond \phi \mid \phi \wedge \psi \mid \phi \vee \psi$
\leq_s	AHML^{\leq_s}	$\phi, \psi := \top \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle \langle n \rangle \rangle_\diamond \psi \mid \phi \langle \langle n \rangle \rangle_\infty \psi$
\leftrightarrow_s	$\text{AHML}^{\leftrightarrow_s}$	$\phi, \psi := \top \mid \neg \phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle \langle n \rangle \rangle_\diamond \psi \mid \phi \langle \langle n \rangle \rangle_\infty \psi$
\leq_t	AHML^{\leq_t}	$\phi := \top \mid \bigvee_{n \in N} \langle n \rangle_\diamond \phi_n \ (\emptyset \neq N \subset \mathbb{N} \text{ is a finite set of priorities})$

Before we address the soundness and completeness of the fragments of the modal logic listed in Table 3, we first point out that there are interesting and fundamental differences with the modal characterisations for the preorders on computational models such as Kripke structures. For instance, disjunction is a necessary part of the logic for the simulation relations in the parity game setting: without it, one cannot show that $w \leq v$ does not hold in the following game:



While disjunction *can* be part of the characteristic logic for the corresponding Kripke structure preorders, there, it is redundant: it does not add to the distinguishing power of the modal logic. This can be explained by the phenomenon that in the Kripke structure setting, it suffices to describe one *fixed* behaviour. In the parity game setting, one must be able to express that a player can *guarantee* (i.e. regardless of the strategies of her opponent) a certain *set* of behaviours; this requires disjunctions. As we will show in Example 9, it is also not the case that disjunction can be added harmlessly to the characteristic logic for a parity game preorder. This suggests there is no easy way to obtain a modal characterisation for a parity game preorder from a modal characterisation of a Kripke structure preorder.

We next state the relation between the fragments identified in Table 3 and the studied preorders, and we illustrate these correspondences using small examples.

Theorem 6. *Let v, w be arbitrary vertices in G . Then:*

1. $v \leq w$ iff $\mathcal{O}^{\text{AHML}^{\leq}}(v) \subseteq \mathcal{O}^{\text{AHML}^{\leq}}(w)$;

2. $v \leftrightarrow w$ iff $\mathcal{O}^{AHML^=} (v) = \mathcal{O}^{AHML^=} (w)$;

Example 7. Consider the parity game of Fig. 1(a). Recall that $v_1 \leq v_2$, see Example 4. We thus have $\mathcal{O}^{AHML^{\leq}}(v_1) \subseteq \mathcal{O}^{AHML^{\leq}}(v_2)$. For instance, both v_1 and v_2 satisfy $\langle 0 \rangle_{\diamond} \langle 0 \rangle_{\diamond} \top$. However, we do not have $v_2 \leq v_1$. This follows from the fact that $\langle 0 \rangle_{\diamond} \langle 1 \rangle_{\diamond} \top$ is a distinguishing formula that holds in v_2 , but fails for v_1 . \square

Theorem 7. *Let v, w be arbitrary vertices in G . Then*

1. $v \leq_s w$ iff $\mathcal{O}^{AHML^{\leq_s}}(v) \subseteq \mathcal{O}^{AHML^{\leq_s}}(w)$.
2. $v \leftrightarrow_s w$ iff $\mathcal{O}^{AHML^{\equiv_s}}(v) = \mathcal{O}^{AHML^{\equiv_s}}(w)$.

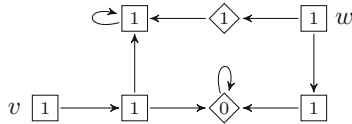
Example 8. One can check that in the parity game of Fig. 1(a), we do not have $v_5 \leq_s v_2$. This is confirmed by the formula $(\top \langle\langle 0 \rangle\rangle_{\diamond} \top) \langle\langle 0 \rangle\rangle_{\diamond} (\top \langle\langle 2 \rangle\rangle_{\diamond} \top)$ that expresses that through vertices with priority 0, a vertex with priority 2 can be reached. It holds in v_5 , but not in v_2 . \square

The relation between parity game trace inclusion and $AHML^{\leq_t}$ is given below.

Theorem 8. *For all $v, w \in V$, we have $v \sqsubseteq_{trace} w$ iff $\mathcal{O}^{AHML^{\leq_t}}(v) \subseteq \mathcal{O}^{AHML^{\leq_t}}(w)$.*

The fragment of AHML needed to characterise the parity game trace inclusion preorder is non-obvious. In particular, the restriction on $AHML^{\leq_t}$ to at all depths of the formulae only allow for $\langle n \rangle_{\diamond}$ for which the priorities are distinct is needed to reduce player \diamond 's powers. Omitting this constraint and allowing for arbitrary disjunctions will lead to a finer relation, as shown by the example below.

Example 9. Consider the parity game depicted below.



Clearly, we have $v \leq_t w$. By Theorem 8 we have $\mathcal{O}^{AHML^{\leq_t}}(v) \subseteq \mathcal{O}^{AHML^{\leq_t}}(w)$. Note that we also have $w \leq_t v$. However, we have $w \models \langle 1 \rangle_{\diamond} (\langle 1 \rangle_{\diamond} \langle 1 \rangle_{\diamond} \top \vee \langle 1 \rangle_{\diamond} \langle 0 \rangle_{\diamond} \top)$ but $v \not\models \langle 1 \rangle_{\diamond} (\langle 1 \rangle_{\diamond} \langle 1 \rangle_{\diamond} \top \vee \langle 1 \rangle_{\diamond} \langle 0 \rangle_{\diamond} \top)$. Omitting the constraint on the disjunctions would therefore lead to incorrect distinguishing formulae. \square

6 Conclusions

We proposed a scheme for lifting preorders on Kripke structures that can be defined through *matching paths* to preorders on parity games. We showed that our scheme can be used to recover preorders for parity games that have already been defined in the literature. Moreover, we demonstrated that we can easily

construct new ones, such as parity game trace inclusion and parity game stuttering simulation, and prove such new relations are preorders (or equivalences) and that they approximate the winning partition of player \diamond .

Our scheme for obtaining parity game relations from existing behavioural relations also extends to other relations by choosing different parameters for the matching predicate. For instance, the bisimulation of [5] on parity games with a finite number of priorities can be recovered using the matching predicate bisimulation–match $_R^c$, where $\mathcal{L}(v) = \{\Omega(v), \circ \mid v \in V_\circ\}$; that is, the vertex labelling is extended with information which player owns the vertex. Observe that the resulting relation is finer than the ones studied in this paper.

Lastly, we provided modal characterisations of all parity game relations studied. Given the fundamental differences between these modal characterisations and their Kripke structure counterparts, we deem it highly unlikely that a logical approach to a systematic way of obtaining parity game relations from Kripke structure relations will be successful.

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. *J. ACM* **49**(5), 672–713 (2002)
2. Arnold, A., Vincent, A., Walukiewicz, I.: Games for synthesis of controllers with partial observation. *TCS* **303**(1), 7–34 (2003)
3. Cranen, S., Keiren, J.J.A., Willemse, T.A.C.: Stuttering mostly speeds up solving parity games. In: Bobaru, M., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) *NFM 2011*. LNCS, vol. 6617, pp. 207–221. Springer, Heidelberg (2011)
4. Cranen, S., Keiren, J.J.A., Willemse, T.A.C.: A cure for stuttering parity games. In: Roychoudhury, A., D’Souza, M. (eds.) *ICTAC 2012*. LNCS, vol. 7521, pp. 198–212. Springer, Heidelberg (2012)
5. Dawar, A., Grädel, E.: The descriptive complexity of parity games. In: Kaminski, M., Martini, S. (eds.) *CSL 2008*. LNCS, vol. 5213, pp. 354–368. Springer, Heidelberg (2008)
6. Emerson, E.A., Jutla, C.S.: Tree automata, Mu-Calculus and determinacy. In: *FOCS 1991*, pp. 368–377. IEEE Computer Society (1991)
7. Friedmann, O., Lange, M.: The modal μ -calculus caught off guard. In: Brännler, K., Metcalfe, G. (eds.) *TABLEAUX 2011*. LNCS, vol. 6793, pp. 149–163. Springer, Heidelberg (2011)
8. Fritz, C., Wilke, T.: Simulation relations for alternating parity automata and parity games. In: Ibarra, O.H., Dang, Z. (eds.) *DLT 2006*. LNCS, vol. 4036, pp. 59–70. Springer, Heidelberg (2006)
9. Gazda, M.W., Willemse, T.A.C.: Consistent consequence for boolean equation systems. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) *SOFSEM 2012*. LNCS, vol. 7147, pp. 277–288. Springer, Heidelberg (2012)
10. Gazda, M.W.: Parity Games, Fixpoint Logic and Relations of Consequence. Eindhoven University of Technology, Forthcoming (2016)
11. Keiren, J.J.A.: Advanced Reduction Techniques for Model Checking. Eindhoven University of Technology (2013)
12. Kissig, C., Venema, Y.: Complementations of coalgebra automata. In: Kurz, A., Lenisa, M., Tarlecki, A. (eds.) *CALCO 2009*. LNCS, vol. 5728, pp. 81–96. Springer, Heidelberg (2009)

13. Namjoshi, K.S.: Abstraction for branching time properties. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 288–300. Springer, Heidelberg (2003)
14. Ranzato, F., Tapparo, F.: Computing stuttering simulations. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 542–556. Springer, Heidelberg (2009)
15. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. TCS **200**(1–2), 135–183 (1998)