Combinatorial Games

Achim Blumensath

last update: 7th March 2021

Contents

1 Positional games

Many games like go, chess, and checkers can be modelled as a directed graph where the vertices represent the different *states*, or *positions*, of the game and

the edges correspond to possible *moves*. Starting in a given initial state such a game consists of a sequence of moves which describe a path in the game graph. We call such a path a *partial play* of the game. The path is a *(complete) play* if it is either infinite or if it ends in some vertex without outgoing edges. For simplicity, we will only consider games with two players, *Player* $\diamond$ and *Player* $\square$, and we assume that the *outcome* of each play is either a *win* for one of the two players, or a *draw*.

How exactly the actions of the players determine the next move depends on the specific kind of game we are considering. The three main options are: (i) *positional games* where the current position determines which player may choose where to move next; (ii) *alternating games* where the players alternate making this choice; and (iii) *simultaneous games* where both players act simultaneously and the resulting move is then determined by combining their choices in some way.

In this section we consider the first variant. A general *positional game* can thus be specified by a tuple

$$\mathfrak{G} = \langle V_\diamond, V_\square, E, W_\diamond, W_\square \rangle ,$$

where $V_\diamond$ is the set of positions belonging to Player $\diamond$, $V_\square$ are the positions for Player $\square$, $V := V_\diamond \cup V_\square$ is the set of all positions, $E \subseteq V \times V$ is the (directed) edge relation, and $W_\diamond$ and $W_\square$ are disjoint sets of infinite plays that determine the winning condition of the game as follows. A player that has to make a move, but cannot, loses. If this does not happen, the game results in an infinite play and the player σ wins whose winning set W_σ contains this play. If the play does not belong to either set, the result is a draw.

The central problem associated with a positional game is which player wins when the game is started in a given initial position and to find a corresponding strategy. Here, a *strategy* for player σ is a function s that, given a partial play p ending in a position $v \in V_\sigma$ chooses one of the outgoing edges. If the value of $s(p)$ does not depend on all of p, but only on the final position, we say that s is *positional* or *memory-free*. We say that a partial play p *conforms* to such a strategy s if, for every proper prefix $p_\circ$ of p ending in a position for Player σ

the extension of $p_\circ$ by the edge $s(p_\circ)$ is again a prefix of p. Finally, a strategy s for Player σ is *winning* from a position $v \in V$ if he wins every (complete) play that starts in v and conforms to s. The *winning region* $\mathcal{W}_\sigma$ of Player σ is the set of all positions $v \in V$ from which he has a winning strategy. Informally, we will say that a player *wins* a game if the given starting position belongs to his winning region. These notions of course only make sense if the game is *determined*, that is, if either one of the players has a winning strategy, or both players have a strategy that guarantees at least a draw. While most of the games one encounters in the wild are in fact determined, we will see below that, although hard to find, there indeed exist games which are not. Some of the classes of games we will consider below are even more well-behaved: they are not only determined but the corresponding strategies can always be chosen to be positional. Such games are called *positional determined*.

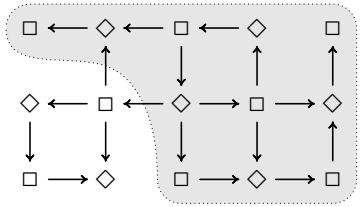
In the terminology we have just established we can rephrase the central problem thus as:

> Given a determined game $\mathfrak{G}$, find the winning regions and the corresponding winning strategies.

In the rest of this note we will consider this context for several classes of positional games.

2 Reachability games

We start with the simplest form of a game: a *positional reachability game* is a positional game where the sets $W_\diamond$ and $W_\square$ are both empty, that is, a game where infinite plays are considered draws. As an example, in the game



(where the label $\Diamond$ denotes positions for Player $\Diamond$ and $\Box$ positions for Player $\Box$) the shaded part constitutes the winning region for Player $\Diamond$, while the winning region for Player $\Box$ is empty.

The winning regions of a reachability game are easy to compute recursively: a player wins from some position v if either v belongs to him and at least one outgoing edge leads to a winning position, or it belongs to his opponent and all the outgoing edges lead to winning positions. To define this formally we introduce the following notation. For $X \subseteq V$, we set

$$\Diamond X := \{\, v \in V \mid \langle v, w\rangle \in E \text{ for some } w \in X \,\},$$
$$\Box X := \{\, v \in V \mid \langle v, w\rangle \in E \text{ implies } w \in X \,\}.$$

We denote the opponent of Player σ by $\overline{\sigma}$, i.e., $\overline{\Diamond} := \Box$ and $\overline{\Box} := \Diamond$. Then we can define the winning region for Player σ as the least set $\mathcal{W}_\sigma$ such that

$$\mathcal{W}_\sigma = (V_\sigma \cap \Diamond \mathcal{W}_\sigma) \cup (V_{\overline{\sigma}} \cap \Box \mathcal{W}_\sigma).$$

Hence, $\mathcal{W}_\sigma$ is the least fixed point of the following *step function*

$$\mathrm{Step}_\sigma(X) := (V_\sigma \cap \Diamond X) \cup (V_{\overline{\sigma}} \cap \Box X).$$

It is easy to see that $\mathrm{Step}_\sigma(X)$ contains all the positions from which Player σ can ensure that in the next step the game either reaches some position in X or the game ends with a win for him. Thus, by iterating the step function we obtain the set of all positions from which the player either wins or eventually reaches a position in X. This iteration of Step_σ is call the *σ-attractor* of the set X. The formal definition is

$$\mathrm{Attr}_\sigma(X) := \bigcup_\alpha \mathrm{Step}^\alpha_\sigma(X),$$

where α ranges over all ordinals (actually, it is sufficient to take the union for all $\alpha < |V|^+$) and $\mathrm{Step}^\alpha_\sigma$, the α-th iteration of the step function, is defined as

follows

$$\mathrm{Step}^0_\sigma(X) := X,$$
$$\mathrm{Step}^{\alpha+1}_\sigma(X) := \mathrm{Step}_\sigma(\mathrm{Step}^\alpha_\sigma(X)),$$
$$\mathrm{Step}^\delta_\sigma(X) := \bigcup_{\alpha<\delta} \mathrm{Step}^\alpha_\sigma(X), \qquad \text{for limit ordinals } \delta.$$

Lemma 2.1. *Player σ has a positional strategy s such that, every play p that starts in some position $v \in \mathrm{Attr}_\sigma(X)$ and that conforms to s is winning or contains some position from X.*

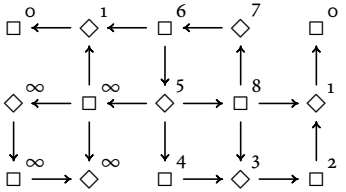
Proof. Let $v \in \mathrm{Attr}_\sigma(X)$. Then $v \in \mathrm{Step}^\alpha_\sigma(X)$, for some ordinal α. We prove the claim by induction on α.

If $\alpha = 0$, then $v \in X$ and the claim is trivial. For the successor step, suppose that $\alpha = \beta + 1$. Then $v \in \mathrm{Step}_\sigma(\mathrm{Step}^\beta_\sigma(X))$ and Player σ has a strategy to either win in one step, or to reach some vertex of $\mathrm{Step}^\beta_\sigma(X)$. In the first case we are done. In the second one, we can continue with the strategy from the inductive hypothesis. Finally, if α is a limit ordinal, then $v \in \mathrm{Step}^\beta_\sigma(X)$, for some $\beta < \alpha$, and the claim follows immediately from the inductive hypothesis. □

Using the notion of an attractor we can define a measure for how long it takes from a given position to win. The *rank* of a position v is the least ordinal α such that

$$v \in \mathrm{Step}^{\alpha+1}_\sigma(\varnothing).$$

If there is no such ordinal, we set the rank to ∞. In the above example the ranks for Player $\Diamond$ are



Next let us take a look at complements of attractors. We call a subset $U \subseteq V$ a *σ-trap* if $\mathrm{Step}_\sigma(V \setminus U) \subseteq V \setminus U$, that is, if the opponent can ensure that Player σ never leaves the set U once the game has entered it. An easy way to find traps is by computing attractors.

Lemma 2.2. *A set U is a σ-trap if, and only if, it is of the form*

$$U = V \setminus \mathrm{Attr}_\sigma(X), \quad \text{for some } X.$$

Proof. ($\Leftarrow$) is obvious since $\mathrm{Step}_\sigma(\mathrm{Attr}_\sigma(X)) \subseteq \mathrm{Attr}_\sigma(X)$. For ($\Rightarrow$), note that $\mathrm{Step}_\sigma(V \setminus U) \subseteq V \setminus U$ implies $U = V \setminus \mathrm{Attr}_\sigma(V \setminus U)$. □

Lemma 2.3. *for every σ-trap U, Player $\overline{\sigma}$ has a positional strategy s ensuring that, starting from any vertex $v \in U$, the game never leaves U.*

Proof. Setting $A := V \setminus U$, we know that $A = \mathrm{Attr}_\sigma(X)$, for some set X. Hence,

$$V_\sigma \setminus A = V_\sigma \setminus \mathrm{Step}_\sigma(A) = V_\sigma \setminus \Diamond A$$
$$\text{and} \quad V_{\overline{\sigma}} \setminus A = V_{\overline{\sigma}} \setminus \mathrm{Step}_\sigma(A) = V_{\overline{\sigma}} \setminus \Box A.$$

In particular, (i) no vertex $v \in V_\sigma \cap U$ has an outgoing edge leading to a vertex in A and (ii) every vertex $v \in V_{\overline{\sigma}} \cap U$ has at least one outgoing edge $\langle v, u \rangle \in E$ with $u \in U$. Consequently, if the game starts in some vertex $v \in U$ Player σ can never move into A, while Player $\overline{\sigma}$ always has the option to stay in U. □

Theorem 2.4. *Positional reachability games are positionally determined with winning regions $\mathcal{W}_\sigma := \mathrm{Attr}_\sigma(\varnothing)$.*

Proof. Let s be the strategy for Player σ from Lemma 2.1 for $\mathcal{W}_\sigma = \mathrm{Attr}_\sigma(\varnothing)$, and let t be the strategy from Lemma 2.3 for $V \setminus \mathcal{W}_\sigma$. As no play can ever reach a position in $\varnothing$ it follows that s is winning for Player σ from every position $v \in \mathcal{W}_\sigma$. While, t ensures that Player $\overline{\sigma}$ does not lose when starting from a position in $V \setminus \mathcal{W}_\sigma$. This implies that $\mathcal{W}_\sigma$ is the winning region for Player σ and that, for $V \setminus (\mathcal{W}_\Diamond \cup \mathcal{W}_\Box)$, each player has a positional strategy ensuring that he does not lose. □

Having shown that winning regions exist, we next take a look at how to efficiently compute them.

Theorem 2.5. *The winning regions of a finite positional reachability game can be computed in linear time.*

Before presenting the linear time algorithm we start with a simpler, non-linear version that is just a direct translation of the definition of an attractor.

```
function Win(v, σ)        // Does Player σ win from position v?
  if v ∈ V_σ then
    if there is an edge v → u with Win(u, σ) then
      return true
    else
      return false
  if v ∈ V_σ̄ then
    if for every edge v → u we have Win(u, σ) then
      return true
    else
      return false
end
```

There are several obvious problems with this algorithm. First of all, it might not terminate if the game graph contains a cycle. And secondly, it is very inefficient (exponential time) as it does not remember if it has already computed the winner of a position and recomputes this information every time. There is a rather straightforward fix for both of these issues: we introduce an array where we store whether we have already visited a position and who the winner is. (Thus each entry can have one of four values: (i) not visited yet, (ii) already visited, but we do not know the winner yet, (iii) Player $\Diamond$ wins, and (iv) Player $\Box$ wins.) With this modification, the algorithm will run in quadratic time.

To improve the runtime to linear, we need to be more clever. We introduce two more array with auxiliary data that helps us to avoid unnecessary work.

Input: game $\langle V_\Diamond, V_\Box, E\rangle$
Output: array containing the winner for every position

```
// initialise auxiliary arrays

forall v ∈ V do
   win[v] := ⊥        // the winner of the position
   P[v] := ∅          // the set of predecessors of v
   n[v] := 0          // the number of (not yet processed)
                      // successors of v
end

forall ⟨u, v⟩ ∈ E do
   P[v] := P[v] ∪ {u}
   n[u] := n[u] + 1
end

// compute the winning regions

forall v ∈ V◇ do
   if n[v] = 0 then Propagate(v, □)
forall v ∈ V□ do
   if n[v] = 0 then Propagate(v, ◇)
return win

procedure Propagate(v, σ) =
   if win[v] ≠ ⊥ then return
   win[v] := σ
   forall u ∈ P[v] do
      n[u] := n[u] − 1
      if u ∈ Vσ or n[u] = 0 then Propagate(u, σ)
   end
end
```

To see that this algorithm works in linear time (in the number of positions plus the number of edges), note that the body of the procedure Propagate

(except for the first line) is executed exactly once for each vertex v. Furthermore, the loop in Propagate is executed once for each incoming edge which means that, in total, it is executed at most as many times as there are edges in the game. Since the precomputation steps are also linear in the number of vertices or edges, it follows that so is the total runtime of the algorithm.

It remains to show that the algorithm really computes the winning regions $\mathcal{W}_\Diamond$ and $\mathcal{W}_\Box$. To see this it is sufficient to note that, every time the procedure Propagate(v, σ) is called and we still have $w[v] = \bot$, then the vertex v really belongs to the winning region for Player σ. This is clear for the two calls of Propagate in the main part of the algorithm, where only vertices v without successors are considered. For the recursive call inside the body of Propagate we need to distinguish two cases. If $u \in V_\sigma$ and v already belongs to $\mathcal{W}_\sigma$ then Player σ can take the edge from u to v to win. Hence, $u \in \mathcal{W}_\sigma$. Otherwise, we have $u \in V_{\overline{\sigma}}$ and $n[u] = 0$, which means that we already know for all successors $w_0, \dots, w_n$ of u to which region they belong. If there is some $w_i \in \mathcal{W}_{\overline{\sigma}}$ then we have already called Propagate$(u, \overline{\sigma})$ when processing w_i and win$[u]$ is already set. Otherwise, all successors belong to $\mathcal{W}_\sigma$, which means that u belongs to it as well.

Horn formulae

As an application of reachability games let us take a look at the satisfiability problem for propositional Horn formulae. Such a formula is an implication of the form

$$A_1 \wedge \cdots \wedge A_n \to B\,,$$

where we allow both the left-hand side and the right-hand side to be empty, i.e., we allow implications of the form $1 \to B$ and $A_1 \wedge \cdots \wedge A_n \to 0$. We are interested in deciding in whether a given set of such formulae is satisfiable. Note that such a set is always satisfiable if there are no implications where the right-hand side is 0. We call such implications *purely negative*. It is not difficult to prove that every set of Horn formulae with no purely negative implications has a *minimal model*, that is, there exists a unique variable assignment that satisfies all the formulae and that only assigns 1 to those

variables that are true in *every* satisfying variable assignment. We will use games to show that this minimal model can be computed in linear time. Then we can check for satisfiability of a given set Φ of Horn formulae by

(1) removing all the purely negative implications from Φ,

(2) computing the minimal model, and

(3) checking that every of the removed implications is true in this model.

As an example, let us consider the following set of Horn formulae.

$$\begin{array}{ccc} 1 \to A & A \wedge C \wedge F \to D & E \to G \\ A \wedge D \to B & B \wedge E \wedge G \to D & 1 \to E \\ F \to C & G \to D & \end{array}$$

The minimal model assigns the value 1 to A, B, D, E, and G, and the value 0 to C and F.

The game corresponding to a set Φ of Horn formulae looks as follows. The positions for Player $\diamond$ are of the form $\langle A \rangle$, where A is a variable appearing in Φ, the positions for Player $\square$ are of the form $[\varphi]$ with $\varphi \in \Phi$. For each formula $A_0 \wedge \cdots \wedge A_{n-1} \to B \in \Phi$, we have edges

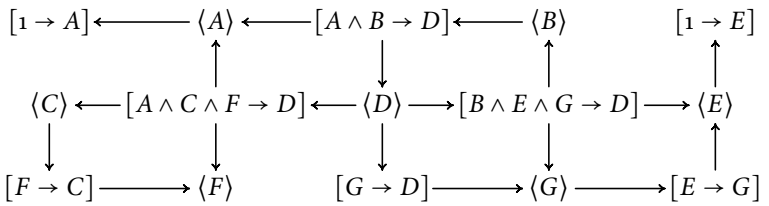
$$\langle B \rangle \quad \longrightarrow \quad [A_0 \wedge \cdots \wedge A_{n-1} \to B]$$

and $\quad [A_0 \wedge \cdots \wedge A_{n-1} \to B] \quad \longrightarrow \quad \langle A_i \rangle\,, \qquad \text{for } i < n\,.$

Intuitively, in the resulting game Player $\diamond$ tries to prove that a variable A must have value 1 by choosing an implication $B_0 \wedge \cdots \wedge B_{n-1} \to A$ that forces it to be true, while Player $\square$ tries to prove that such an implication is not applicable by finding some condition B_i that is not met. With this intuition it is straightforward to show that our game has the desired properties.

Lemma 2.6. *Let Φ be a set of Horn formulae that does not contain any purely negative implications. A position of the form $\langle A \rangle$ belongs to the winning region of Player $\diamond$ if, and only if, the variable A is true in the minimal model of Φ.*

The game corresponding to the above set of formulae is the following one, whose winning regions we have already computed above.

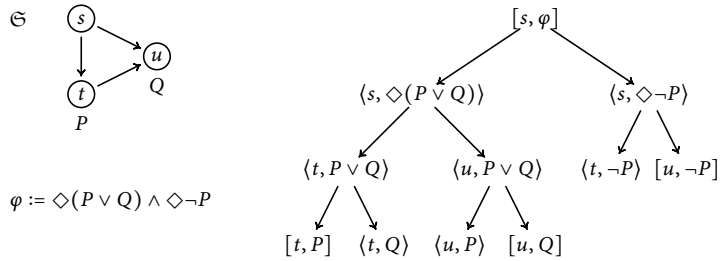


Modal logic

As a second application let us take a look at the model-checking problem for propositional modal logic. Given a transition system $\mathfrak{S}$ with starting state s and a modal formula φ in *negation normal form*, we can construct a game that is won by Player $\diamond$ if, and only if, $\mathfrak{S}, s \vDash \varphi$. The positions are of the form $\langle t, \psi \rangle$ where t is a state of $\mathfrak{S}$ and ψ is a subformula of φ. Intuitively, in such a position Player $\diamond$ tries to prove that $\mathfrak{S}, t \vDash \psi$, while Player $\square$ tries to show that $\mathfrak{S}, t \nvDash \psi$. The moves are as follows.

$$\begin{array}{rll} \langle t, \psi_0 \vee \psi_1 \rangle \to \langle t, \psi_i \rangle\,, & & \text{for } i = 0, 1\,, \\ \langle t, \psi_0 \wedge \psi_1 \rangle \to \langle t, \psi_i \rangle\,, & & \text{for } i = 0, 1\,, \\ \langle t, \diamond \vartheta \rangle \to \langle u, \vartheta \rangle\,, & & \text{if } t \to u \text{ is an edge of } \mathfrak{S}\,, \\ \langle t, \square \vartheta \rangle \to \langle u, \vartheta \rangle\,, & & \text{if } t \to u \text{ is an edge of } \mathfrak{S}\,. \end{array}$$

Finally, positions of the form $\langle t, \psi_0 \vee \psi_1 \rangle$ and $\langle t, \diamond \vartheta \rangle$ belong to Player $\diamond$; those of the form $\langle t, \psi_0 \wedge \psi_1 \rangle$ and $\langle t, \square \vartheta \rangle$ belong to Player $\square$; and a position of the form $\langle t, P \rangle$ belongs to Player $\diamond$ if $t \notin P$; otherwise, it belongs to Player $\square$. Similarly, a position of the form $\langle t, \neg P \rangle$ belongs to Player $\diamond$ if $t \in P$. For instance, for the transition system $\mathfrak{S}$ and the formula φ on the left below, the resulting game (or at least its reachable part) is depicted on the right.



It is straightforward to check by induction on ψ that Player $\Diamond$ has a winning strategy from a position $\langle t, \psi \rangle$ if, and only if, $\mathfrak{S}, t \vDash \psi$.

3 Conway games

For many actual games like chess, go, etc., looking at a given position does not tell you which player's turn it is. To model such games it can be advantageous to assign the players not to the positions but to the moves. A *Conway game* is a graph

$$\mathfrak{G} = \langle V, E_\Diamond, E_\Box \rangle$$

with two edge relations $E_\Diamond, E_\Box \subseteq V \times V$, one for Player $\Diamond$ and one for Player $\Box$. During a play of such a game starting in a given position and with a given starting player, the players alternatingly choose moves from their respective edge relation, that is, a play of such a game consists of a path through $\mathfrak{G}$ where the edges alternate between $E_\Diamond$-edges and $E_\Box$-edges. As usual, we consider a player whose turn it is but who cannot make a move to lose the game. For simplicity, we will only consider Conway games where infinite plays are considered draws. Thus, to win a player has to manoeuvre his opponent into a position where he cannot make a move.

The question of determinacy for Conway games becomes more involved as, given a starting position, we must distinguish several more cases: it can be that one of the players has a winning strategy no matter who goes first, or it can be that always the player that make the first move wins, or it is the

player making the second move. We therefore say that a Conway game is *determined* if, for every position v, one of the following cases occurs.

- When taking the first turn Player $\Diamond$ has a winning strategy from v.
- When taking the first turn Player $\Box$ has a winning strategy from v.
- The player that moves first has a winning strategy from v.
- The player that moves second has a winning strategy from v.
- Both players have strategies from v that ensure at least a draw.

To compute the winning regions we proceed similar to positional games. We start by defining a step function. Because the game is alternating, it is convenient to use a function that takes *two* steps at a time. We set

$$\mathrm{DStep}_\sigma(X) := \langle \sigma \rangle [\overline{\sigma}] X \,,$$

where, for $X \subseteq V$ and $\sigma \in \{\Diamond, \Box\}$,

$$\langle \sigma \rangle X := \{\, v \in V \mid \langle v, w \rangle \in E_\sigma \text{ for some } w \in X \,\} \,,$$
$$[\sigma] X := \{\, v \in V \mid \langle v, w \rangle \in E_\sigma \text{ implies } w \in X \,\} \,.$$

Then the winning regions are

$$\mathcal{W}_{\sigma,1} := \bigcup_\alpha \mathrm{DStep}_\sigma^\alpha(\varnothing) \,,$$
$$\mathcal{W}_{\sigma,2} := [\overline{\sigma}] \mathcal{W}_{\sigma,1} \,.$$

Where $\mathcal{W}_{\sigma,1}$ denotes the set of vertices from which Player σ wins if he goes first and $\mathcal{W}_{\sigma,2}$ is the set of vertices from which he wins if the opponent goes first.

Lemma 3.1. *For every $v \in \mathcal{W}_{\sigma,1} \cup \mathcal{W}_{\sigma,2}$, Player σ has a positional winning strategy.*

Proof. It is sufficient to provide winning strategies for every $v \in \mathcal{W}_{\sigma,1}$ since, in a game starting with Player $\overline{\sigma}$ from a position $\mathcal{W}_{\sigma,2}$, the first move necessarily

leads to a position in $\mathcal{W}_{\sigma,1}$. (Or Player $\overline{\sigma}$ has no move and loses immediately.) Hence, suppose that $v \in \mathcal{W}_{\sigma,1}$ and let α be the least ordinal such that $v \in \mathrm{DStep}_\sigma^{\alpha+1}(\varnothing)$. Then

$$v \in \langle\sigma\rangle[\overline{\sigma}]\mathrm{DStep}_\sigma^\alpha(\varnothing)$$

implies that Player σ can make a move to some position

$$w \in [\overline{\sigma}]\mathrm{DStep}_\sigma^\alpha(\varnothing)\,,$$

and the response of the opponent leads to some position

$$v' \in \mathrm{DStep}_\sigma^\alpha(\varnothing)\,.$$

Consequently, we can construct the desired strategy by induction on α. □

Theorem 3.2. *Conway games are positionally determined.*

Proof. Let $\mathcal{W} := \mathcal{W}_{\diamond,1} \cup \mathcal{W}_{\diamond,2} \cup \mathcal{W}_{\square,1} \cup \mathcal{W}_{\square,2}$. By Lemma 3.1, it is sufficient to prove that, for every position $v \in U := V \setminus \mathcal{W}$, both players have strategies that ensure a draw. Similar to the proof of Lemma 2.3, it is straightforward to check that, in every position $v \in V$ both players have the ability to choose an edge $\langle v, u\rangle$ with $u \in U$ and no player σ can choose an edge $\langle v, u\rangle$ with $u \in \mathcal{W}_{\sigma,2}$. □

4 Gale–Steward games

Reachability games are very simple since we can ignore infinite plays. Let us now take a look at what happens for games where every infinite play is winning for one player or the other. *Gale-Steward games* are simple games of this kind. There are played on a tree with set of vertices $V := A^*$. Every position $v \in V$ has outgoing edges to all vertices of the form va with $a \in A$. The players strictly alternate, that is, positions $v \in (A^2)^*$ of even length belong to Player $\diamond$ and those $v \in A(A^2)^*$ of odd length to Player $\square$. The winning condition is given by a set $W \subseteq A^\omega$, which contains the infinite plays won by Player $\diamond$. All other infinite player are won by Player $\square$.

Proposition 4.1. *There exist Gale-Steward games that are not determined.*

Proof. We play on the complete binary tree with vertices $V := \{0,1\}^*$. As the game graph is acyclic, every strategy is automatically positional. Thus, a strategy for Player σ is a function $V_\sigma \to \{0,1\}$ where V_σ is either the set of all $v \in \{0,1\}^*$ of even length, or the set of all v of odd length. There are $\kappa := 2^{\aleph_0}$ such functions. We fix enumerations $(s_\alpha)_{\alpha<\kappa}$ and $(t_\alpha)_{\alpha<\kappa}$ of all strategies for, respectively, Player $\diamond$ and Player $\square$. To construct a non-determined game we have to find two disjoint sets $W_\diamond, W_\square \subseteq \{0,1\}^\omega$ of infinite paths such that none of the s_α and t_α are winning strategies in the game with winning condition $W_\diamond, W_\square$.

We start with a bit of notation. Given a strategy s, we denote by $[s]$ the set of all infinite plays $p \in A^\omega$ that conform to s. With this notation we can say that a strategy s for Player σ is a winning if, and only if, $[s] \subseteq W_\sigma$.

By induction on $i < \kappa$, we construct two sequences $(\xi_i)_{i<\kappa}$ and $(\zeta_i)_{i<\kappa}$ of elements of A^ω as follows. Suppose that we have already defined ξ_i and ζ_i, for all $i < \alpha$. Then we pick some element $\xi_\alpha \in [s_\alpha]$ that is different from ξ_i and ζ_i, for all $i < \alpha$, and we pick some element $\zeta_\alpha \in [t_\alpha]$ that is different from ξ_i and ζ_i, for all $i \leq \alpha$, and also from the just chosen ξ_α. Note that we can do so since $[s_\alpha]$ has size κ, while the set $\{\,\xi_i \mid i < \alpha\,\} \cup \{\,\zeta_i \mid i < \alpha\,\}$ has size $|\alpha| < \kappa$. The same holds for $[t_\alpha]$.

We claim that the game with winning conditions

$$W_\diamond := \{\,\zeta_i \mid i < \kappa\,\} \quad \text{and} \quad W_\square := A^\omega \setminus W_\diamond$$

is not determined. For the proof, consider a strategy s for Player $\diamond$. Then $s = s_\alpha$, for some $\alpha < \kappa$. Since $\xi_\alpha \in [s_\alpha] \cap W_\square$, we have $[s] \not\subseteq W_\diamond$. Hence, s is not a winning strategy. In the same way it follows that no strategy t for Player $\square$ is winning. □

As Gale-Steward games can be non-determined in general, we have to put restrictions on the allowed winning conditions to get positive results. One handy way to do so is by equipping the set of all infinite plays with a topology. We call a set O of infinite plays *open* if there exists a set P of finite partial plays such that contains all infinite plays starting with some $p \in P$.

The complement of an open set is called *closed*. Note that the open sets are closed under arbitrary unions and finite intersections. Hence, they form a topology. A set is *Borel* if it is contained in the smallest class of sets that contains the open ones and that is closed under complement and countable unions.

Theorem 4.2 (Martin [8]). *If $W \subseteq A^\omega$ is Borel, then $\langle A^*, W\rangle$ is determined.*

The proof is a bit involved. Instead of proving the result in its full generality, we will only consider the much simpler case of open and closed winning conditions, i.e., where $W_\Diamond$ is open and $W_\Box$ closed, or vice versa. By symmetry, we may assume that the winning condition for Player $\Diamond$ is open. Then $W_\Diamond$ is determined by some set $P \subseteq A^*$ of prefixes and every play containing a position from P is winning. Thus, open winning condition corresponds to a reachability game: Player $\Diamond$ has to reach a position in P. As we have seen in Section 2, the winning region for Player $\Diamond$ in such a game is given by $\mathrm{Attr}_\Diamond(P)$. The complement of $\mathrm{Attr}_\Diamond(P)$ is a $\Diamond$-trap. Since every infinite play inside this complement cannot contain positions from P, all such plays are winning for Player $\Box$. Consequently, the game is determined with winning regions

$$\mathrm{Attr}_\Diamond(P) \quad \text{and} \quad A^* \smallsetminus \mathrm{Attr}_\Diamond(P)\,.$$

5 Regular games and parity games

In general, Gale-Steward games have no finite representation as the winning set W can be an arbitrary set of infinite sequences. In this section we take a look at a simple simple way to represent certain Gale-Steward games in a finite way: if the winning set W is a *regular* set of infinite plays, we can use an automaton to represent it. Or we can use ω-semigroups instead of automata. This leads to the following definition.

A *regular game* over an ω-semigroup $\mathfrak{S} = \langle S, S_\omega\rangle$ is a game

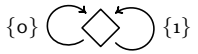
$$\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \lambda, W\rangle$$

where $\lambda : E \to S$ is an edge-labelling and the winning set is given by a subset $W \subseteq S_\omega$. Player $\Diamond$ wins an infinite play p if the product of the corresponding edge labels evaluates to an element of W.

As an example consider the ω-semigroup $\mathfrak{S} = \langle S, S_\omega\rangle$ where $S = \wp\{0,1\}$ and $S_\omega = \wp\{0,1\}$. We define the product by

$$\begin{aligned} a \cdot b &:= a \cup b\,, && \text{for } a, b \in S\,,\\ a \cdot u &:= u\,, && \text{for } a \in S \text{ and } u \in S_\omega\,,\\ \pi(a_0, a_1, \dots) &:= \bigcap_{i<\omega} \bigcup_{i\le k<\omega} a_i\,, && \text{for } a_0, a_1, \dots \in S\,. \end{aligned}$$

In the game



with winning set $W = \{\{0,1\}\}$ Player $\Diamond$ has a winning strategy by alternating between the two edges. But note that he does not have a positional winning strategy as using only one of the edges will result in a loss.

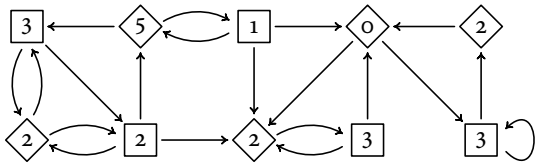
Computing the winning regions of a regular game is more complicated than for open games. The difference is that, instead of reaching a certain set once, we have to be able to reach it over and over again. To simplify our task, let us start by considering a special case of regular games of the following form. A *parity game* is a game of the form

$$\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \Omega\rangle$$

where $\Omega : V \to D$ for some finite set $D \subseteq \omega$. We call Ω the *priority function* and $\Omega(v)$ the *priority* of the position v. Player $\Diamond$ wins an infinite play $p = (v_i)_{i<\omega}$ if it satisfies the *parity condition*:

$$\liminf_{i<\omega} \Omega(v_i) \text{ is even.}$$

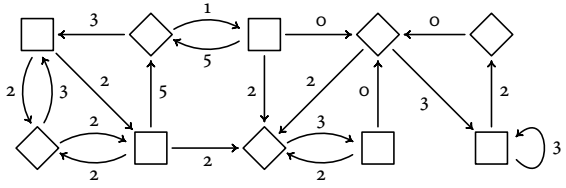
For instance, in the following parity game Player $\Diamond$ wins from every position except for the one in the lower right. (The numbers denote the priorities.)



As we will show below, parity games are simpler than general regular games. In particular, they are positionally determined. In fact, a regular game is positionally determined if, and only if, it is a parity game. Let us start our investigation of parity games by noting that they are regular. We can turn the set D of priorities into an ω-semigroup $\mathfrak{D} = \langle D, D_\omega \rangle$ with two infinite elements $D_\omega := \{0, 1\}$ and the product

$$\begin{aligned} k \cdot l &:= \min\{k, l\}\,, && \text{for } k, l \in D\,, \\ k \cdot b &:= b\,, && \text{for } k \in D \text{ and } b \in D_\omega\,, \\ \pi(k_0, k_1, \dots) &:= (\liminf_{i<\omega} k_i) \bmod 2\,, && \text{for } k_0, k_1, \dots, \in D\,. \end{aligned}$$

Then a parity game $\mathfrak{G}$ can be turned into a regular game by labelling every edge $\langle u, v \rangle$ by $\Omega(v)$. In the above example we get:



Positional determinacy

For the proof that parity games are positionally determined, we introduce some terminology. Let us say that a strategy s for Player σ is *winning on* some set $U \subseteq V$ if every play p that conforms to s and starts at a vertex in U is winning for σ and p never leaves the set U. We call a subset $U \subseteq V$ a *σ-domain* if it is a $\overline{\sigma}$-trap and Player σ has a positional strategy that is winning on U.

Lemma 5.1. *The union of a set of σ-domains is again a σ-domain.*

Proof. Let $W = \bigcup_{i \in I} U_i$, where each U_i is a σ-domain. We start by proving that W is a $\overline{\sigma}$-trap. Let $v \in W$. Then $v \in U_i$, for some $i \in I$. We distinguish two cases. If $v \in V_\sigma$ it follows that there is an edge $\langle v, u \rangle$ with $u \in U_i \subseteq W$. If $v \in V_{\overline{\sigma}}$ then every edge $\langle v, u \rangle$ leads to a vertex $u \in U_i \subseteq W$. This implies that W is a $\overline{\sigma}$-trap.

It remains to construct a positional strategy s for Player σ on W. By assumption, there are positional strategies t_i that are winning on U_i. Fix a well-order $\leq$ on I and define

$$s(v) := t_i(v)\,, \quad \text{for the least } i \in I \text{ with } v \in U_i\,.$$

We claim that every play p starting at a vertex in W and conforming to s is winning for Player σ and that p never leaves W. The second part is clear since the opponent cannot leave W and the choices of Player σ satisfy

$$s(v) = t_i(v) \in U_i \subseteq W\,, \quad \text{for every vertex } v \in V_\sigma \cap W\,.$$

For the first part of the above claim, note that, if p contains a vertex of U_i then the rest of p will be contained in $\bigcup_{j \leq i} U_j$. As I was well-ordered, it follows that there is some index $i \in I$ such that, after finitely many steps, p will remain in U_i and the corresponding suffix q of p will conform to t_i. Since t_i is winning, it follows that q satisfies the parity condition. Hence, so does p. □

Theorem 5.2. *Parity games are positionally determined.*

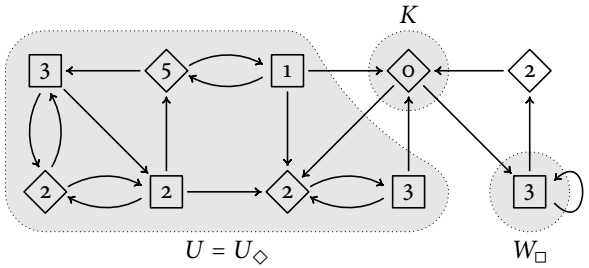
Proof. Let $\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \Omega \rangle$ be a parity game. We prove the claim by induction on the number of priorities used. Let k be be the minimal priority of $\mathfrak{G}$ and σ the player it belongs to. Let $W_{\overline{\sigma}}$ be the union of all $\overline{\sigma}$-domains. By Lemma 5.1, $W_{\overline{\sigma}}$ is also a $\overline{\sigma}$-domain. In particular, Player $\overline{\sigma}$ has a positional strategy that is winning from every vertex in $W_{\overline{\sigma}}$. It is therefore sufficient to prove that $W_\sigma := V \setminus W_{\overline{\sigma}}$ is a σ-domain.

Since $\mathrm{Attr}_{\overline{\sigma}}(W_{\overline{\sigma}})$ is a $\overline{\sigma}$-domain, it follows by definition of $W_{\overline{\sigma}}$ that $\mathrm{Attr}_{\overline{\sigma}}(W_{\overline{\sigma}}) = W_{\overline{\sigma}}$. Consequently, Lemma 2.2 implies that the complement

$W_\sigma = V \setminus \mathrm{Attr}_{\overline{\sigma}}(W_{\overline{\sigma}})$ is a $\overline{\sigma}$-trap. For a set $X \subseteq V$, we denote by $\mathfrak{G}[X]$ the subgame of $\mathfrak{G}$ consisting of all positions in X. Define

$$K := W_\sigma \cap \Omega^{-1}(k) \quad \text{and} \quad U := W_\sigma \setminus \mathrm{Attr}_\sigma(K/W_\sigma)\,,$$

where $\mathrm{Attr}_\sigma(K/W_\sigma)$ is the σ-attractor of K computed in $\mathfrak{G}[W_\sigma]$. Since no position in U uses the priority k, we can apply the inductive hypothesis to the game $\mathfrak{G}[U]$ and obtain a partition $U = U_\sigma \uplus U_{\overline{\sigma}}$ of U into a σ-domain U_σ and a $\overline{\sigma}$-domain $U_{\overline{\sigma}}$.



We will show next that $W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$ is a $\overline{\sigma}$-domain. By definition of $W_{\overline{\sigma}}$, it then follows that $U_{\overline{\sigma}} = \emptyset$. To see that $W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$ is a σ-trap, we distinguish four cases.

(i) Let $v \in V_{\overline{\sigma}} \cap W_{\overline{\sigma}}$. As $W_{\overline{\sigma}}$ is a $\overline{\sigma}$-domain, we can find a successor u of v that belongs to $W_{\overline{\sigma}} \subseteq W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$.

(ii) Let $v \in V_\sigma \cap W_{\overline{\sigma}}$. As $W_{\overline{\sigma}}$ is a $\overline{\sigma}$-domain, every successor u of v belongs to $W_{\overline{\sigma}} \subseteq W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$.

(iii) Let $v \in V_{\overline{\sigma}} \cap U_{\overline{\sigma}}$. As $U_{\overline{\sigma}}$ is a $\overline{\sigma}$-domain in $\mathfrak{G}[U]$, we can find a successor u of v that belongs to $U_{\overline{\sigma}} \subseteq W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$.

(iv) Let $v \in V_\sigma \cap U_{\overline{\sigma}}$. As $U_{\overline{\sigma}}$ is a $\overline{\sigma}$-domain in $\mathfrak{G}[U]$, every successor u of v either belongs to $U_{\overline{\sigma}}$ or to $V \setminus U$. Since U is a σ-trap in the subgame $\mathfrak{G}[W_\sigma]$, the latter is only possible if $u \notin W_\sigma$, i.e., $u \in W_{\overline{\sigma}}$. Consequently, all successors belong to $W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$.

It remains to find a positional strategy for Player $\overline{\sigma}$ on $W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$. As $W_{\overline{\sigma}}$ and $U_{\overline{\sigma}}$ are $\overline{\sigma}$-domains in, respectively, $\mathfrak{G}$ and $\mathfrak{G}[U]$, there are positional

strategies t_W and t_U for Player $\overline{\sigma}$ on these to sets. We define a strategy s by

$$s(v) := \begin{cases} t_U(v) & \text{if } v \in U\,, \\ t_W(v) & \text{otherwise}\,. \end{cases}$$

To show that s is winning, consider a play p conforming to s and starting in some position in $W_{\overline{\sigma}} \cup U_{\overline{\sigma}}$. If p enters $W_{\overline{\sigma}}$, it never leaves this set. Consequently, the rest of p conforms to t_W and is therefore winning. Otherwise, the play stays the whole time in $U_{\overline{\sigma}}$ and conforms to t_U. Thus, it is also winning.

We have shown that $U = U_\sigma$ is a σ-domain in $\mathfrak{G}[U]$. Let t be the corresponding strategy. To conclude the proof it is sufficient to show that W_σ is a σ-domain in $\mathfrak{G}$. We have already seen above that it is a $\overline{\sigma}$-trap. Hence, it remains to construct a positional strategy for Player σ on W_σ. Note that $W_\sigma = U_\sigma \cup \mathrm{Attr}_\sigma(K/W_\sigma)$. For positions $v \in U$, we use the strategy t for $\mathfrak{G}[U]$. For positions $v \in K$, we choose an arbitrary successor in W_σ. For the remaining positions, we use the attractor strategy that ensures that we visit K. Let s be the resulting strategy. To show that it is winning, consider a play p conforming to s and starting in some position in W_σ. If p enters U_σ it will stay in this set and the remainder of the play conforms to t. Hence, p is winning. Otherwise, p is entirely contained in $\mathrm{Attr}_\sigma(K/W_\sigma)$. This implies that p either ends in a terminal vertex belonging to Player $\overline{\sigma}$, or it visits the set K infinitely often. In both cases Player σ wins. □

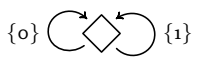
Finite-memory strategies

One can show that all regular games are Borel. Hence, determinacy follows from the Theorem of Martin. But we can prove a stronger statement: regular games admit what is called *finite-memory* strategies. A *finite-memory strategy* for Player σ is given by a finite set M (the *memory*) and two functions $s : M \times V_\sigma \to E$ and $\alpha : M \times E \to M$ such that, given a state $m \in M$ and a vertex $v \in V_\sigma$, $s(m, v)$ returns the outgoing edge e to be chosen by Player σ after which $\alpha(m, e)$ will be the new memory state. (If there is no outgoing edge, we let s remain undefined.) Formally, we say that a play $p = (e_i)_i$

(which, for regular games where the edge labelling matters, we consider as a sequence of edges) *conforms* to such a strategy if there exists a sequence $(m_i)_i$ of memory states such that, for every step i,

- if $e_i = \langle v_i, v_{i+1} \rangle$ with $v_i \in V_\sigma$, then $e_i = s(m_i, v_i)$, and
- $m_{i+1} := \alpha(m_i, e_i)$.

In the game



from above, Player $\Diamond$ has a finite-memory strategy with $M = \{0, 1\}$. If $m = 0$, he takes the left edge and sets the memory state to 1. Otherwise, he takes the right edge and sets the state to 0. This results in alternatingly taking the two edges, which is winning for him.

We can reformulate the definition of a finite-memory strategy as follows. For a regular game $\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \lambda, W \rangle$ and a function $\alpha : M \times E \to M$, we define the *product game*

$$\mathfrak{G} \times_\alpha M := \langle V'_\Diamond, V'_\Box, E', \lambda', W' \rangle$$

with positions

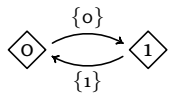
$$V'_\Diamond := V_\Diamond \times M \quad \text{and} \quad V'_\Box := V_\Box \times M\,,$$

edge relations

$$E' := \left\{ \langle \langle u, m \rangle, \langle v, n \rangle \rangle \;\middle|\; \langle u, v \rangle \in E \text{ and } n = \alpha(u, \langle u, v \rangle) \right\},$$

edge labelling $\lambda'(\langle \langle u, m \rangle, \langle v, n \rangle \rangle) := \lambda(\langle u, v \rangle)$, and the same winning condition $W' := W$. Then a strategy is finite-memory for $\mathfrak{G}$ if, and only if, there exists a finite set M and a function $\alpha : M \times E \to M$ such that s is a positional strategy in the game $\mathfrak{G} \times_\alpha M$.

In the above example, the product $\mathfrak{G} \times_\alpha M$ is the game



which clearly has a positional winning strategy.

Note that this operation of equipping a game with memory does not change the game much. There exist one-to-one correspondences between

- plays of $\mathfrak{G}$ and of $\mathfrak{G} \times_\alpha M$;
- strategies of $\mathfrak{G}$ and of $\mathfrak{G} \times_\alpha M$;
- winning strategies of $\mathfrak{G}$ and of $\mathfrak{G} \times_\alpha M$.

The only difference between these two games is the amount of memory a strategy needs. In particular, some positional strategies of $\mathfrak{G} \times_\alpha M$ might correspond to strategies of $\mathfrak{G}$ which are not positional. This is exactly what we need to prove the following result.

Theorem 5.3 (Büchi, Landweber). *In every regular game both players have a finite-memory winning strategy on their respective winning regions.*

Proof. Let $\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \lambda, W \rangle$ be a regular game. We fix a deterministic parity automaton $\mathcal{A} = \langle Q, S, \delta, q_0, \Omega \rangle$ recognising the set W of winning plays and construct the product game $\mathfrak{G} \times_\delta Q$. In this game a play p is winning for Player $\Diamond$ if, and only if, its projection to the second component produces an accepting run of $\mathcal{A}$. Consequently, we can turn $\mathfrak{G} \times_\delta Q$ into a parity game by using as priority function the function Ω from $\mathcal{A}$ applied to the second component. Since parity games are positionally determined, we obtain two positional winning strategies $s_\Diamond$ and $s_\Box$ for the two players in their respective winning regions. As we have seen in the remark before the theorem, these two strategies induce finite-memory strategies in the original game. □

Positionally determined games

We have seen that parity games admit positional strategies while arbitrary regular games only admit finite-memory ones. One might wonder whether

there exists a larger class of games with positional strategies. It turns out that is not the case. We will prove below that (nearly) every regular, positionally determined game is equivalent to a parity game – with one notable caveat: we will only be able to establish this statement for games with a winning condition of the following form. A winning condition $W \subseteq S_\omega$ is called *prefix-invariant* if

$$w \in W \Leftrightarrow aw \in W\,, \quad \text{for all } a \in S \text{ and } w \in S_\omega\,.$$

Note that this is not much of a restriction since most of the common winning conditions used in game theory or automata theory are of this form.

To analyse such conditions we start with an observation from semigroup theory. In a *finite* ω-semigroup $\mathfrak{S} = \langle S, S_\omega \rangle$ every infinite product $a_0 a_1 a_2 \cdots$ has a factorisation of the form be^ω. If W is prefix-invariant, we have $be^\omega \in W \Leftrightarrow e^\omega \in W$. Thus, the set W is completely determined by the powers e^ω it contains.

Given a regular game $\mathfrak{G} = \langle V_\Diamond, V_\Box, E, \lambda, W \rangle$ over an ω-semigroup $\mathfrak{S}$, let us write $W_\Diamond := W$ and $W_\Box := S_\omega \setminus W$. The *winning condition* of $\mathfrak{G}$ is the pair $\langle \mathfrak{S}, W \rangle$. We call the set

$$P_\sigma := \{\, e \in S \mid e^\omega \in W_\sigma \,\}$$

the *period set* for Player σ. Finally, let us say that $\langle \mathfrak{S}, W \rangle$ is *equivalent* to a parity condition if there exists a function $\Omega : S \to \omega$ such that

$$\pi(a_0, a_1, \dots) \in W \quad \text{iff} \quad \liminf_i \Omega(a_i) \text{ is even.}$$

In this case, we can turn every game $\mathfrak{G}$ with winning condition $\langle \mathfrak{S}, W \rangle$ into a parity game as follows. We first replace all edge labels by their image under Ω. In this way we obtain a kind of parity game where the priorities are attached to the edges instead of the vertices. We can turn the resulting game into an ordinary parity game by adding intermediate vertices to the edges where we can put the priorities.

Theorem 5.4 (Colcombet, Niwiński). *Let $\mathfrak{S}$ be an ω-semigroup (not necessarily finite) and $W \subseteq S_\omega$ prefix-invariant. If all games with winning condition $\langle \mathfrak{S}, W \rangle$ are positionally determined, then $\langle \mathfrak{S}, W \rangle$ is equivalent to a parity condition.*

We split the proof of this theorem into two lemmas. We start with listing some basic properties of the period sets.

Lemma 5.5. *Let $\mathfrak{S}$ be an ω-semigroup and $W \subseteq S_\omega$ prefix-invariant. If all games with winning condition $\langle \mathfrak{S}, W \rangle$ are positionally determined, then the following condition holds:*

(a) *$a, b \in P_\sigma$ implies $ab \in P_\sigma$.*

(b) *$ab \in P_\sigma$ implies $ba \in P_\sigma$.*

(c) *Every element $w \in S_\omega$ that can be written as an infinite product of elements of P_σ belongs to W_σ.*

(d) *For all $A, B \subseteq S$,*

$$(\exists a \in A)(\forall b \in B)[ab \in P_\sigma] \Leftrightarrow (\forall b \in B)(\exists a \in A)[ab \in P_\sigma]\,.$$

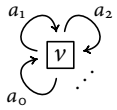
(e) *For every $a \in S$, there exists some $n > 0$ such that*

$$ab_0, \dots, ab_k \in P_\sigma \;\Rightarrow\; a^n b_0 \cdots b_k \in P_\sigma\,, \quad \text{for all } b_0, \dots, b_k \in S\,.$$

(f) *For $a \in P_\sigma$ and $B \subseteq S$,*

$$aB \subseteq P_\sigma \quad \text{implies} \quad aB^* \subseteq P_\sigma\,.$$

Proof. (c) Suppose that $w = \pi(a_0, a_1, \dots) \notin W_\sigma$. We have to show that there is some index k with $a_k^\omega \notin W_\sigma$. Consider the game $\mathfrak{G}$ with a single position v belonging to Player $\overline{\sigma}$ and one a_i-labelled edge $v \to v$, for every $i < \omega$.

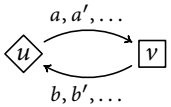


Since $w \notin w$, Player $\overline{\sigma}$ has a winning strategy in $\mathfrak{G}$ by choosing in turn i the edge with label a_i. By assumption, he also has a positional winning strategy s. Let a_k be the label of the edge chosen by s. As the resulting play is winning, it follows that $a_k^\omega \notin W_\sigma$. Hence, $a_k \notin P_\sigma$.

(a) If $a, b \in P_\sigma$, then $(ab)^\omega \in W_\sigma$ by (c).

(b) Let $ab \in P_\sigma$. Then $a(ba)^\omega = (ab)^\omega \in W_\sigma$ implies, by prefix-invariance, that $(ba)^\omega \in W_\sigma$.

(d) ($\Rightarrow$) is trivial. For ($\Leftarrow$), consider the game $\mathfrak{G}$ with positions $V_\sigma = \{u\}$ and $V_{\overline{\sigma}} = \{v\}$. For every $a \in A$, we add an a-labelled edge $u \to v$, and for every $b \in B$, a b-labelled edge $v \to u$.



Then Player σ can win this game by playing as follows. Every time Player $\overline{\sigma}$ chooses a b-labelled edge, Player σ responds with an a-labelled edge for some $ab \in P$. By assumption, Player σ also has a positional winning strategy s. Let a be the label of the edge chosen by s. For every $b \in B$, there exists a play with labelling $(ab)^\omega$ conforming to s. Since these plays must be winning, it follows that $(ab)^\omega \in W_\sigma$, i.e., $ab \in P_\sigma$, for all b.

(e) Fix $a \in S$ and set $B_0 := \{\, b \in S \mid ab \in P_\sigma \,\}$. Applying (d) to the sets

$$A := \{\, a^n \mid n > 0 \,\} \quad \text{and} \quad B := \{\, b_0 \cdots b_k \mid b_0, \ldots, b_k \in B_0 \,\},$$

we see that it is sufficient to prove that

$$a^{k+1} b_0 \cdots b_k \in P_\sigma, \quad \text{for } b_0, \ldots, b_k \in B_0.$$

We do so by induction on k. If $k = 0$, then $b_0 \in B_0$ implies that $a^1 b_0 \in P_\sigma$. Hence, suppose that $k > 0$. By inductive hypothesis and the fact that $b_k \in B_0$, we have $a^k b_0 \cdots b_{k-1} \in P_\sigma$ and $ab_k \in P_\sigma$. Hence, $(ab_k)^\omega \in W_\sigma$. Since $a(b_k a)^\omega = (ab_k)^\omega \in W_\sigma$, prefix-invariance implies that $(b_k a)^\omega \in W_\sigma$, i.e., $b_k a \in P_\sigma$. Consequently, it follows by (c) that $(a^k b_0 \cdots b_{k-1} b_k a)^\omega \in W_\sigma$ and

we can again use prefix-invariance to show that

$$(a^{k+1} b_0 \cdots b_k)^\omega = a(a^k b_0 \cdots b_{k-1} b_k a)^\omega \in W_\sigma.$$

Thus, $a^{k+1} b_0 \cdots b_k \in P_\sigma$.

(f) Fix $a \in P_\sigma$ and $B \subseteq S$ with $aB \subseteq P_\sigma$. By (e) the set $N := \{\, n \geq 1 \mid a^n B^+ \subseteq W_\Diamond \,\}$ is non-empty. Note that, if $n \in N$ then $a^n u \in P_\sigma$, for all $u \in B^+$. By (a) this implies that $aa^n u \in P_\sigma$. Consequently, $n + 1 \in N$. Thus, $N = \{k, k+1, k+2, \ldots\}$ for some $k < \omega$.

We claim that $k = 1$. Then $aB^+ \subseteq P_\sigma$. Since also $a \in P_\sigma$, it follows that $aB^* \subseteq P_\sigma$, as desired. Hence, it remains to prove the claim.

For a contradiction, suppose that $k > 1$ and set $m := k - 1$. Then $m \notin N$, but $2m \in N$. Hence, there is some $u \in B^+$ with $a^m u \in P_{\overline{\sigma}}$. By (a) and (b), it follows that $ua^m \in P_{\overline{\sigma}}$, $a^m u a^m \in P_{\overline{\sigma}}$, and $a^{2m} uu \in P_{\overline{\sigma}}$. Hence, $2m \notin N$. A contradiction. □

The next lemma now concludes the proof of Theorem 5.4.

Lemma 5.6. *Let $\mathfrak{S}$ be an ω-semigroup and $W \subseteq S_\omega$ a prefix-invariant set such that all games with winning condition $\langle \mathfrak{S}, W \rangle$ are positionally determined. There exists a function $\Omega : S \to [2n]$ such that*

(1) *Ω maps $P_\Diamond$ to even numbers and $P_\Box$ to odd ones,*

(2) $\Omega(a) \leq \Omega(b)$ *implies* $\Omega(ab) \equiv \Omega(a) \pmod 2$.

(3) $\pi(a_0, a_1, \ldots) \in W_\Diamond$ iff $\liminf_i \Omega(a_i)$ *is even.*

Proof. Consider the relation $\sqsubseteq \subseteq P_\Box \times P_\Box$ defined by

$$a \sqsubseteq b \quad :\text{iff} \quad ac \in P_\Box \Rightarrow bc \in P_\Box, \quad \text{for all } c \in S.$$

We start by proving that it is a linear preorder of finite index.

Reflexivity and transitivity of $\sqsubseteq$ follows immediately from the definition. For linearity, suppose that a and b are non-comparable. Then there are elements c and d such that

$$ac \in P_\Box, \quad bc \notin P_\Box, \quad ad \notin P_\Box, \quad bd \in P_\Box.$$

By Lemma 5.5 (a) and (b), it follows that

$$acbd \in P_\Box\,, \quad da \in P_\Diamond\,, \quad cb \in P_\Diamond\,, \quad dacb \in P_\Diamond\,, \quad acbd \in P_\Diamond\,.$$

A contradiction.

It remains to prove that $\sqsubseteq$ has finite index. For a contradiction, suppose otherwise. We distinguish two cases. If there exists an infinite strictly increasing chain $a_0 \sqsubset a_1 \sqsubset a_2 \sqsubset \cdots$, we can fix elements $c_i \in S$ with $a_i c_i \notin P_\Box$ and $a_{i+1} c_i \in P_\Box$. Then Lemma 5.5 (a) and (c) implies that $c_i a_{i+1} \in P_\Box$ and

$$c_0 a_1 c_1 a_2 c_2 a_3 \cdots \in W_\Box\,.$$

But by the same argument as above, $a_i c_i \in P_\Diamond$ implies that

$$a_0 c_0 a_1 c_1 a_2 c_2 a_3 \cdots \in W_\Diamond\,.$$

A contradiction to prefix-invariance.

Similarly, if there exists an infinite strictly decreasing chain $a_0 \sqsupset a_1 \sqsupset a_2 \sqsupset \cdots$, we can fix elements $c_i \in S$ with $a_i c_i \in P_\Box$ and $a_{i+1} c_i \notin P_\Box$. In the same way as above it follows that

$$c_0 a_1 c_1 a_2 c_2 a_3 \cdots \in W_\Diamond \quad \text{and} \quad a_0 c_0 a_1 c_1 a_2 c_2 a_3 \cdots \in W_\Box\,.$$

Again a contradiction.

To conclude the proof, let $B_0 \sqsupset \cdots \sqsupset B_{m-1}$ be a decreasing enumeration of all $\sqsubseteq$-classes and set

$$A_i := \{\, c \in P_\Diamond \mid ac \in P_\Box \text{ for some/all } a \in B_i \,\}\,, \quad \text{for } 0 \leq i < m\,.$$

In addition, we set $A_{-1} := P_\Diamond$ and $A_m := \varnothing$. Note that, by definition of $\sqsubseteq$, we have $P_\Diamond = A_{-1} \supseteq A_0 \supseteq \cdots \supseteq A_{m-1} = A_m = \varnothing$. We claim that the function $\Omega : S \to [2m+1]$ defined by

$$\Omega(a) := \begin{cases} 2k & \text{if } a \in A_{k-1} \smallsetminus A_k\,, \\ 2k+1 & \text{if } a \in B_k\,, \end{cases}$$

has the desired properties.

(1) Clearly, Ω maps each $A_k \subseteq P_\Diamond$ to an even number and each $B_k \subseteq P_\Box$ to an odd one.

(2) Suppose that $\Omega(a) \leq \Omega(b)$. We distinguish four cases. If both $\Omega(a)$ and $\Omega(b)$ are even, then $a, b \in P_\Diamond$, which implies by Lemma 5.5 (a) that $ab \in P_\Diamond$. Hence, $\Omega(ab)$ is also even. In the same way it follows that, if $\Omega(a)$ and $\Omega(b)$ are odd, then so is $\Omega(ab)$.

Suppose that $\Omega(a) = 2k+1$ and $\Omega(b) = 2i$. Then $k < i$, $a \in B_k$ and $b \in A_{i-1} \smallsetminus A_i \subseteq A_k$. By definition of A_k it follows that $ab \in P_\Box$. Hence, $\Omega(ab)$ is odd.

Finally, suppose that $\Omega(a) = 2k$ and $\Omega(b) = 2i+1$. Then $k \leq i$, $a \in A_{k-1} \smallsetminus A_k$ and $b \in B_i \subseteq B_k$. By definition of A_k and the fact that $a \notin A_k$, it follows that $ba \in P_\Diamond$. Hence, Lemma 5.5 (a) implies that $ab \in P_\Diamond$ and $\Omega(ab)$ is even.

(3) Fix $a_0, a_1, \cdots \in S$ and set $k := \liminf_i \Omega(a_i)$. Since W is prefix-invariant, we may assume w.l.o.g. that there is no i with $\Omega(a_i) < k$. Set

$$B := \Omega^{-1}(k) \quad \text{and} \quad C := \Omega^{-1}[\{k+1, \ldots, 2m\}]\,.$$

Then we can factorise the sequence $(a_i)_i$ into words $u_0, u_1, \cdots \in BC^*$. Let c_i be the product of u_i. If k is even, it follows by (b) that $bc \in P_\Diamond$, for all $b \in B$ and $c \in C$. Consequently, we can use Lemma 5.5 (f) and (c) to show that $c_i \in P_\Diamond$ and $\pi(a_0, a_1, \ldots) = \pi(c_0, c_1, \ldots) \in W_\Diamond$. If k is odd, it follows in the same way that $\pi(a_0, a_1, \ldots) \in W_\Box$. □

Solving parity games

To solve reachability games, we introduced the notion of a rank which, intuitively counts how far away from the goal we are. For parity games the situation is more complicated since we have to reach the goal not only once but repeatedly. It is possible to define ranks also for parity games if, instead of a single ordinal, one uses tuples of them, one for each priority k that counts how far away we are from a position of that priority. As this turns out to be a bit technical and not very enlightening, we will not do so.

Instead, we will present an algorithm for computing the winning regions that is similar to the construction in the Theorem of Büchi and Landweber. We will prove that, for every parity game $\mathfrak{G}$, there exists an action $\alpha : M \times E \to M$ that turns the product game $\mathfrak{G} \times_\alpha M$ into a reachability game. Then we can use the linear time algorithm from Section 2 to compute the winning regions. The memory M we will construct below has size $n^{\mathcal{O}(\log d)}$. As we can solve reachability games in linear time, we therefore obtain the following complexity bound.

Theorem 5.7. *The winning regions of a parity game $\mathfrak{G}$ with n positions and d priorities can be computed in time $n^{\mathcal{O}(\log d)}$.*

The precise complexity of computing the winning regions of a parity game are still unknown. One can show that the problem belongs to the complexity class U ∩ co-U, which means that it is probably not NP-complete. It might even belong to P, but no one has found a polynomial time algorithm so far.

Let us present the algorithm the above theorem is based on. Consider a parity game $\mathfrak{G}$ and let p be a play of $\mathfrak{G} \times_\alpha M$. We say that p contains an *even cycle* if $p = xyz$ where y is a non-empty path starting and ending at the same position of $\mathfrak{G}$ (the memory contents may differ) and such that the least priority seen along y is even. Note that, whether or not a given play p contains an even cycle is a reachability property: once we have found the end of the cycle, we do not need to look at the rest of p. We will design our memory M in such a way that detection of such cycles becomes easy. But first, let us show that the existence of even cycles is equivalent to winning.

Lemma 5.8. *Let $\mathfrak{G}$ be a finite parity game and $\alpha : M \times E \to M$ an action. The following statements are equivalent.*

(1) *Player $\Diamond$ has a winning strategy s in $\mathfrak{G}$.*

(2) *Player $\Diamond$ has a strategy s' for $\mathfrak{G} \times_\alpha M$ such that all cycles in every play conforming to s' are even.*

(3) *Player $\Diamond$ has a strategy s' for $\mathfrak{G} \times_\alpha M$ such that every play conforming to s' has an even cycle.*

Proof. (2) ⇒ (3) is trivial.

(3) ⇒ (1) Suppose that Player $\Diamond$ does not have a winning strategy for $\mathfrak{G}$. By determinacy, it then follows that Player $\Box$ has a positional winning strategy s in that game. This strategy induces a strategy s' for Player $\Box$ in the game $\mathfrak{G} \times_\alpha M$. Let p' be a play conforming to s' and let p be the corresponding play of $\mathfrak{G}$. Then p conforms to s. If p' contained an even cycle then, s being positional, it would follow that p contained infinite repetitions of this cycle. In particular, the least priority seen infinitely often in p would be even and p would be winning for Player $\Diamond$. A contradiction to our choice of s.

(1) ⇒ (2) Let s be a winning strategy for Player $\Diamond$ in $\mathfrak{G}$. W.l.o.g. we may assume that s is positional. Let s' be the strategy in $\mathfrak{G} \times_\alpha M$ induced by s. To show that it has the desired property, consider a play p' conforming to s' and let p be the corresponding play in $\mathfrak{G}$. Then p conforms to s and is, therefore, winning. Since s is positional, p consists of a path leading to a cycle which is repeated infinitely often. Let k be the minimal priority along this cycle. As p is winning, it follows that k is even. Hence, so is (every copy of) the cycle. □

How can we detect an even cycle? The easiest way would be to store all the positions of $\mathfrak{G}$ we have already seen. Once we see one of them for the second time, we have found a cycle. Unfortunately, storing that many positions requires too much memory. So instead, we resort to a counting trick.

Let $\mathfrak{G}$ be a parity game with n positions and priorities $\{0, \dots, d-1\}$. We use the ω-semigroup $\mathfrak{S}$ with domains $S := [d]$ and $S_\omega := [d]$ and product

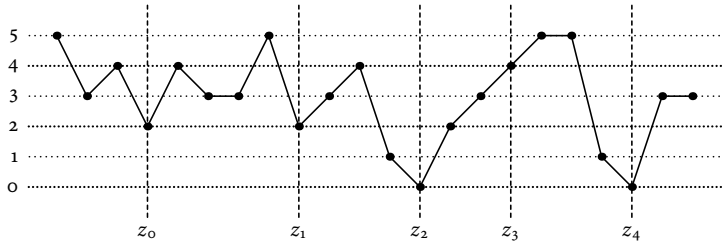
$$\begin{aligned} k \cdot k' &:= \min\{k, k'\}, && \text{for } k, k' \in S, \\ k \cdot l &:= l, && \text{for } k \in S \text{ and } l \in S_\omega, \\ \pi(k_0, k_1, k_2, \dots) &:= \liminf_{i<\omega} k_i, && \text{for } k_i \in S, \end{aligned}$$

and we label an edge $u \to v$ of $\mathfrak{G}$ by the semigroup element $\Omega(u) \in S$.

Given a finite word $w = k_0 \cdots k_{n-1} \in [d]^*$, we call a sequence $z_0 < \cdots < z_{m-1} < n$ an *even factorisation* of w if

- k_{z_i} is even, for all $i < m$,
- $k_j \geq \min\{k_{z_i}, k_{z_{i+1}}\}$, for all $z_i \leq j \leq z_{i+1}$, $i < m-1$,
- $k_j \geq k_{z_0}$, for $j \leq z_0$,
- $k_j \geq k_{z_{m-1}}$, for $j \geq z_{m-1}$.

We call m the *length* of the factorisation and the number $\min_i k_{z_i}$ its *value*.



Note that, if w has an even factorisation of length m and value k and w' has one of length m' and value k', the ww' has an even factorisation of length $m + m'$ and value $\min\{k, k'\}$.

For each $l < \omega$, we will define a deterministic automaton $\mathcal{A}_l$ over the alphabet $[d]$ that computes the length and value of an even factorisation of (some suffix of) its input. The precise definition is as follows. The set of states M_l consists of a special *accepting state* $*$ plus all triples

$$\langle s, \bar{k}, \bar{n}\rangle \in [l+1] \times [d]^* \times [l]^*,$$

where $s \leq l$ is the *size* of the state, $\bar{k}$ is a non-decreasing sequence of priorities of length s, and $\bar{n}$ a strictly decreasing sequence of *counters*, also of length s. To compute the cardinality of M_l, note that we can encode each state $\langle s, \bar{k}, \bar{n}\rangle$ as a word $c_0 \dots c_{l-1} \in ([d] + \square)^l$ where

$$c_i := \begin{cases} k_j & \text{if } n_j = i\,, \\ \square & \text{otherwise}\,. \end{cases}$$

$\langle 3, \langle 0, 3, 4\rangle, \langle 5, 2, 1\rangle\rangle$

	4	3			0

Below we will choose l such that $2^{l-1} \leq n < 2^l$. Then it follows

$$\begin{aligned} |M_l| &\leq (d+1)^l + 1 \leq (d+1)^{\log(n+1)} + 1 \\ &= (n+1)^{\log(d+1)} + 1 \in n^{\mathcal{O}(\log d)}, \end{aligned}$$

which is the right size for the theorem.

Before defining the transition relation of $\mathcal{A}_l$, let us state the intended behaviour of the automaton.

Lemma 5.9. *Suppose that after having read a word $w \in [d]^*$ the automaton $\mathcal{A}_l$ enters the state $\langle s, \bar{k}, \bar{n}\rangle$. Then*

$$w = w' w_0 \dots w_{s-1}\,,$$

where each w_i has an even factorisation of length at least 2^{n_i} with value k_i. In particular, the word w has a suffix with an even factorisation of length at least

$$\sum_{i<s} 2^{n_i}.$$

Before giving the proof, we have to finish the definition of $\mathcal{A}_l$. The initial state is the pair $\langle 0, \langle\rangle, \langle\rangle\rangle$ consisting of two empty sequences. In a state $\langle s, \bar{k}, \bar{n}\rangle$ when reading the letter c, the automaton can enter the state $\langle s', \bar{k}', \bar{n}'\rangle$ if one of the following three conditions is met.

(i) $s = 0$ or $k_{s-1} \leq c$,

$$s' = s\,, \quad \bar{k}' = \bar{k}\,, \quad \text{and} \quad \bar{n}' = \bar{n}\,.$$

(ii) There is some $0 < i < s$ such that $k_{i-1} < c < k_i$,

$$s' = i+1\,, \quad \bar{k}' = \langle k_0, \dots, k_{i-1}, c\rangle\,, \quad \bar{n}' = \langle n_0, \dots, n_{i-1}, n_i\rangle\,.$$

(iii) There is some $0 < i < s$ such that $k_{i-1} \leq c$, the priorities $k_i, \dots, k_{s-1}$ are even, $\langle n_i, \dots, n_{s-1}\rangle = \langle s-i-1, s-i-2, \dots, 1, 0\rangle$,

$$s' = i+1\,, \quad \bar{k}' = \langle k_0, \dots, k_{i-1}, c\rangle\,, \quad \bar{n}' = \langle n_0, \dots n_{i-1}, n_i + 1\rangle\,.$$

(iv) There is some $0 < i < s$ such that $k_{i-1} \leq c$, the priority k_i is odd, while $k_{i+1}, \ldots, k_{s-1}$ are even,

$$\langle n_i, \ldots, n_{s-1} \rangle = \langle s - i - 1, s - i - 2, \ldots, 1, 0 \rangle ,$$
$$s' = i + 1 , \quad \bar{k}' = \langle k_0, \ldots, k_{i-1}, c \rangle , \quad \bar{n}' = \langle n_0, \ldots, n_{i-1}, n_i + 1 \rangle .$$

(v) $s = l$, each priority k_i is even, c is even, and the next state is $*$.

(vi) Once $\mathcal{A}_l$ has reached the state $*$, it remains there.

If there are several possible transitions, we choose the one that leads to a state of minimal length.

Proof of Lemma 5.9. We prove the claim by induction on the length of w. For $w = \langle\rangle$, $\mathcal{A}_l$ is in the initial state $\langle 0, \langle\rangle, \langle\rangle\rangle$ and the claim is trivial.

For the inductive step, suppose that the input is wc with $w \in [d]^*$ and $c \in [d]$, and let $\langle s, \bar{k}, \bar{c} \rangle$ be the state after reading w. By inductive hypothesis, w has a suffix of the form $w_0 \cdots w_{s-1}$ where each w_i has an even factorisation of length at least 2^{n_i} with value k_i. We distinguish several cases, depending on which transition the automaton takes while reading the last letter c.

If the last transition is of the form (i), we obtain the desired suffix $w'_0 \ldots w'_{s-1}$ of wc by setting $w'_i := w_i$, for $i < s - 1$, and $w'_{s-1} := w_{s-1}c$.

If the last transition is of the form (ii), let i be the index such that $k_{i-1} < c < k_i$. We set $w'_j := w_j$, for $j < i$, and $w'_i := w_i \cdots w_{s-1}c$. Then w'_i has an even factorisation of length at least $2^{n_i} + \cdots + 2^{n_{s-1}} + 1 \geq 2^{n_i}$.

If the last transition is of the form (iii) or (iv), let i be the index from the above definition. We set $w'_j := w_j$, for $j < i$, and $w'_i := w_i \cdots w_{m-1}c$. Then w'_i has an even factorisation of length at least

$$2^{n_i} + \cdots + 2^{n_{s-1}} + 1 = 2^{s-i-1} + 2^{s-i-2} + \cdots + 2^1 + 2^0 + 1 = 2^{s-i}. \qquad \square$$

With the help of Lemma 5.9, we are able to show that the information contained in the states of $\mathcal{A}_l$ is sufficient to detect whether the input contains an even cycle.

Lemma 5.10. *Let p be an infinite play in a parity game $\mathfrak{G}$ with n positions, and let $w = (c_i)_{i<\omega}$ be the sequence of priorities along p. If $l > \log n$ and $\mathcal{A}_l$ accepts w, then p contains an even cycle.*

Proof. Let $\langle l, \bar{k}, \bar{n} \rangle \in M_l$ be the last state in the run of $\mathcal{A}_l$ before it enters the state $*$, let w' be the prefix of w leading to this state, and let c be the next input letter. By Lemma 5.9, the word w' has a suffix with an even factorisation of length at least

$$2^{l-1} + \cdots + 2^0 = 2^l - 1 \geq n ,$$

while $w'c$ has a suffix with an even factorisation $z_0 < \cdots < z_{m-1}$ of length $m \geq 2^l - 1 + 1 > n$. By the Pigeon Hole Principle it follows that, after reading the additional letter c, there must be two positions $z_i < z_j$ in the factorisation that correspond to the same vertex v of $\mathfrak{G}$. Let p_0 be the part of p corresponding to the path between these two positions. To see that p_0 is an even cycle, note that the minimal priority seen along the closed path is the minimal value of $c_{z_i}, \ldots, c_{z_j}$. In particular, it is even. Hence, the play is winning. $\square$

Lemma 5.11. *The automaton $\mathcal{A}_l$ accepts every word w satisfying the parity condition.*

Proof. By induction on l we will prove that, starting from an arbitrary state $\langle s, \bar{k}, \bar{n} \rangle$ the automaton $\mathcal{A}_l$ accepts every infinite word w that satisfies the parity condition. Hence, fix l and a run ρ of $\mathcal{A}_l$ on w. If ρ contains the accepting state $*$, we are done. Hence, suppose otherwise.

If every state of size l appears only finitely often in ρ, some suffix of ρ is a run of $\mathcal{A}_{l-1}$. By inductive hypothesis, this suffix is accepting. Hence, so is ρ.

Consequently, we may assume that some state $\langle l, \bar{k}', \bar{n}' \rangle$ of size l appears infinitely often in ρ. The way the transitions are defined it follows that, after the first appearance of $\langle l, \bar{k}', \bar{n}' \rangle$ every state $\langle s, \bar{k}'', \bar{n}'' \rangle$ in ρ satisfies $k''_0 = k'_0$. Let ρ' be the sequence of states obtained from ρ by (i) removing the part before the first appearance of $\langle l, \bar{k}', \bar{n}' \rangle$ and (ii) removing the first components of all remaining states, i.e., replacing $\langle s, \bar{k}'', \bar{n}'' \rangle$ by

$$\left\langle s - 1, \langle k''_1, \ldots, k''_{s-1} \rangle, \langle n''_1, \ldots, n''_{s-1} \rangle \right\rangle .$$

Then ρ' is a run of $\mathcal{A}_{l-1}$ on the corresponding suffix of w. Again it follows by inductive hypothesis that this run is accepting. Hence, so is ρ. □

We are finally able to prove Theorem 5.7. Set $l := \lfloor \log n \rfloor + 1$ and let $\alpha : M_l \times E \to M_l$ be the action induced by $\mathcal{A}_l$. We claim that a position v of $\mathfrak{G}$ belongs to the winning region of Player $\Diamond$ in $\mathfrak{G}$ if, and only if, Player $\Diamond$ has a strategy in the game $\mathfrak{G} \times_\alpha M_l$ from the position $\langle v, \langle 0, \langle\rangle, \langle\rangle\rangle\rangle$ to reach some position of the form $\langle u, * \rangle$. Since the latter game is a reachability game of size $n \times \mathcal{O}(n^{\log n}) = \mathcal{O}(n^{\log n})$ and we can compute its winning regions in linear time, the theorem follows. Hence, it remains to prove the claim.

($\Leftarrow$) follows immediately by Lemma 5.10 and the implication (2) $\Rightarrow$ (1) in Lemma 5.8, while ($\Rightarrow$) follows by Lemma 5.11 and the implication (1) $\Rightarrow$ (3) in Lemma 5.8.

Notes

One of the first articles on games is by Zermelo [9], who proved the determinacy of chess. Conway games were subsequently fully developed by Conway [5].

Gale and Steward [6] proved the existence of indetermined games and the determinacy for open games. The full proof of Borel determinacy is by Martin [8].

The section on parity games follows [10] and [7]. The Theorem of Büchi and Landweber was originally proved in [2], and Theorem 5.4 is taken from [4]. The algorithm to solve parity games in Theorem 5.7 is from [3]. Our presentation owes much to a set of lecture notes by Bojańczyk and Czerwiński [1].

References

[1] M. Bojańczyk and W. Czerwiński, *An Automata Toolbox*. lecture notes, 2018.

[2] J. R. Büchi and L. H. Landweber, *Solving sequential conditions by finite-state strategies*, Trans. Amer. Math. Soc., 139 (1969), pp. 367–378.

[3] C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan, *Deciding parity games in quasipolynomial time*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017, 2017, pp. 252–263.

[4] T. Colcombet and D. Niwinski, *On the positional determinacy of edge-labeled games*, Theoretical Computer Science, 352 (2006), pp. 190–196.

[5] J. H. Conway, *On Numbers and Games*, Academic Press, 1976.

[6] D. Gale and F. M. Steward, *Infinite Games with Perfect Information*, in Contributions to the Theory of Games, vol. 28 of Annals of Mathematical Studies, Princeton University Press, 1953, pp. 245–266.

[7] E. Grädel and I. Walukiewicz, *Positional Determinacy of Games with Infinitely Many Priorities*, Logical Methods in Computer Science, 2 (2006).

[8] D. A. Martin, *Borel Determinacy*, Annals of Mathematics, 102 (1975), pp. 363–371.

[9] E. Zermelo, *Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels*, in Proc. 5th Congress of Mathematicians, Cambridge University Press, 1913, pp. 501–504.

[10] W. Zielonka, *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*, Theoretical Computer Science, 200 (1998), pp. 135–183.

Symbol Index

Chapter 1

Index