Parity Games on Undirected Graphs

Dietmar Berwanger, Olivier Serre

Parity Games on Undirected Graphs

Dietmar Berwanger[a], Olivier Serre[b]

[a] *LSV, École Normale Supérieure de Cachan & CNRS, France*
[b] *LIAFA, Université Paris Diderot – Paris 7 & CNRS, France*

Abstract

We examine the complexity of solving parity games in the special case when the underlying game graph is undirected. For strictly alternating games, that is, when the game graph is bipartite between the nodes of the two players, we observe that the solution can be computed in linear time. In contrast, when the assumption of strict alternation is dropped, we show that the problem is as hard in the undirected case as it is in the general, directed, case.

Keywords: parity games, graph structure complexity

1. Introduction

Parity games are path-forming games on directed graphs with important applications to automata theory, logic, and verification. Notably, the problem of determining the winner of a parity game on a finite graph is polynomial-time equivalent to checking whether a μ-calculus formula holds in a finite model [3, 4].

The computational complexity of solving parity games is subject to an intriguing open question. The known upper bound is NP $\cap$ Co-NP, but it is unknown whether the problem can be solved in polynomial time; the currently best deterministic algorithm runs in time $n^{\mathcal{O}(\sqrt{n})}$ [7].

The problem received considerable attention over the last decade, and specialised algorithms were proposed for subclasses where certain structural parameters of the game graph are restricted. At the outset, Obdržálek [8] exhibited a polynomial-time algorithm for parity games on graphs of bounded tree-width. Yet, as he points out, tree-width measures the connectivity of the *undirected* graphs underlying the *directed* game graph, and the algorithm would not give good bounds, for instance, on directed acyclic graphs, even though solving the games on such graphs is easy. In line with the intuition that the complexity of parity games is sensitive to the direction of edges, several studies focused henceforth on connectivity measures designed for *directed* graphs. Thus, it was shown that parity games can be solved in polynomial time on directed graphs of bounded entanglement [2], dag-width [1], clique-width[9], or Kelly-width [6].

In this note, we argue that the directedness of the game graph may not be the main responsible for the computational complexity of parity games, and show that hard instances can already be found among games on undirected graphs, i.e., graphs where each edge comes with a back-edge. In support of the prevalent belief that games on undirected graphs are simple, we present a linear-time algorithm for the case where the two players strictly alternate their moves, that is, where the graph is bipartite between the nodes of the two players. However, for the case where this assumption is dropped, we show the following, somehow surprising, result: Solving parity games on undirected graphs is polynomial-time equivalent to solving parity games on arbitrary, directed graphs.

2. Parity games

A *finite graph* is a pair $G = (V, E)$ where V is a finite set of *vertices* and $E \subseteq V \times V$ is an *edge* relation. We say that the graph is *undirected*, if the edge relation E is symmetric, that is, $(v_1, v_2) \in E$ if, and only if, $(v_2, v_1) \in E$, for all $v_1, v_2 \in V$. A *dead-end* is a vertex v such that there is no vertex v' with $(v, v') \in E$. A *path* is a finite sequence $v_1, v_2, \ldots, v_\ell$ such that $(v_i, v_{i+1}) \in E$ for every $1 \leq i < \ell$; a *cycle* is a path $v_1, v_2, \ldots, v_\ell$ with $\ell > 1$ and $v_1 = v_\ell$. A graph that does not have any cycle is *acyclic*. The *size* of a graph is the number $|V|$ of its edges plus the number $|E|$ of its edges.

A *parity game* is a game for two players, we call them Éloïse and Abelard, described by a tuple $\mathcal{G} = (V, E, V_\mathrm{E}, V_\mathrm{A}, \Omega)$ where $G = (V, E)$ is a finite graph, $V = V_\mathrm{E} \uplus V_\mathrm{A}$ is a partition of the vertex set into positions V_E of Éloïse and V_A of Abelard, and $\Omega : V \to \mathbb{N}$ is a *priority* function. We say that the game is *bipartite* if $E \subseteq V_\mathrm{E} \times V_\mathrm{A} \cup V_\mathrm{A} \times V_\mathrm{E}$. An example of a parity game is depicted in Figure 1: circles and squares represent vertices of V_E and V_A, respectively, and the label indicates the priority.

To play a game $\mathcal{G}$, the two players move a token along the edges of the game graph. Starting from a designated initial vertex v_0, a *play* proceeds as follows: if $v_0 \in V_\mathrm{E}$ then Éloïse moves to a vertex v_1 such that $(v_0, v_1) \in E$; if $v_0 \in V_\mathrm{A}$, Abelard does the move. Then, the player who owns v_1 chooses

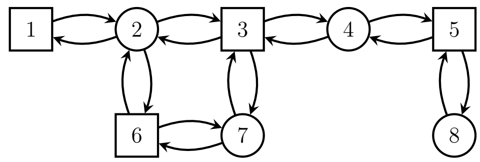

Figure 1: A bipartite parity game on an undirected graph.

a successor v_2 and so on. If, at some point, the player in turn cannot move, she or he loses the play. Otherwise, the play yields an infinite sequence $v_0 v_1 v_2 \cdots \in V^\omega$, and the winner is determined by looking at the least priority seen infinitely often along this sequence: Éloïse wins, if $\liminf(\Omega(v_i))_{i \geq 0}$ is even, otherwise Abelard wins. A *partial play* is a prefix of a play.

A *strategy* for Éloïse is a function σ assigning, to every partial play that ends in a vertex $v \in V_{\mathrm{E}}$, a vertex v' such that $(v, v') \in E$. A play $\pi = v_0 v_1 v_2 \ldots$ *follows* the strategy σ if $v_{i+1} = \sigma(v_0 \ldots v_i)$, for all $i \geq 0$ with $v_i \in V_{\mathrm{E}}$. A strategy σ for Éloïse is *winning* from a position $v \in V$ if she wins every play that starts from v and follows σ. In this case, we also say that the vertex $v \in V$ is winning for Éloïse; the *winning region* of Éloïse consists of all her winning vertices. The corresponding notions for Abelard are defined analogously. For example, in the parity game in Figure 1, the winning region for Éloïse is $\{1, 2, 6, 7\}$ and the one for Abelard is $\{3, 4, 5, 8\}$ (here, we identify vertices with their priority).

Of special interest are strategies that depend only on the current vertex. A strategy σ is *positional* if, for every partial play π and every vertex v, we have $\sigma(\pi \cdot v) = \sigma(\pi' \cdot v)$. In that case, the strategy can be represented as a function $\sigma : V \to V$. A crucial property of parity games is that positional winning strategies always exist.

Theorem 1 (Positional determinacy [5])**.** *For any vertex of a parity game, either Éloïse or Abelard has a positional winning strategy.*

3. Solving bipartite games on undirected graphs is easy

For this section, let us fix an undirected graph $G = (V, E)$, underlying a bipartite game $\mathcal{G} = (V, E, V_{\mathrm{E}}, V_{\mathrm{A}}, \Omega)$. We show that the winning regions in $\mathcal{G}$ can be computed in linear time in the size of G.

Towards this, we consider the directed graph $G' = (V, E')$ obtained from G by removing one of any two opposite edges (i, j), $(j, i) \in E$ as follows:

$$E' := \{(i,j) \in E \ : \ i \in V_{\mathrm{E}} \text{ and } \min\{\Omega(i), \Omega(j)\} \text{ even, or} \\ i \in V_{\mathrm{A}} \text{ and } \min\{\Omega(i), \Omega(j)\} \text{ odd } \}.$$

Let $\mathcal{G}' = (V, E', V_{\mathrm{E}}, V_{\mathrm{A}}, \Omega)$ be the parity game on the obtained orientation G' of G, with V_{E}, V_{A}, and Ω as in $\mathcal{G}$. (For an example, see Figure 2.) We argue that this transformation does not change the winning regions of the game. Intuitively, this is because the removed edges would not be profitable to a player using a positional strategy.

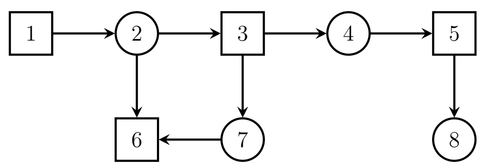

Figure 2: Orientation of the game in Figure 1.

Lemma 1. *A vertex $v \in V$ is winning for Éloïse in $\mathcal{G}$ if, and only if, v is winning for Éloïse in $\mathcal{G}'$.*

Proof. Let $\sigma : V \to V$ be a positional strategy for Éloïse in $\mathcal{G}$ that is winning from v_0, and let σ' be a positional strategy in $\mathcal{G}'$ that agrees with σ on any vertex $v \in V$ with $(v, \sigma(v)) \in E'$. We claim that σ and σ' agree along any play from v_0 in $\mathcal{G}$. Concretely, for every partial play $v_0 v_1 \ldots v_i$ in $\mathcal{G}$ that follows σ to a vertex $v_i \in V_{\mathrm{E}}$, the prescribed prolongation $v_{i+1} = \sigma(v_i)$ is along an edge (v_i, v_{i+1}) that belongs to E'. This is because $v_0 \cdots (v_i \cdot v_{i+1})^\omega$ is an infinite play in $\mathcal{G}$ that follows the winning strategy σ, hence the smallest priority $\min(\Omega(v_i), \Omega(v_{i+1}))$ seen infinitely often must be even, implying that the edge (v_i, v_{i+1}) is maintained in E'. Consequently, every play from v_0 that follows σ' in $\mathcal{G}'$ corresponds to a play in $\mathcal{G}$ that follows σ and is thus winning for Éloïse. The same reasoning applies for positions in the winning region of Abelard. $\square$

It turns out that the transformed game $\mathcal{G}'$ has a very simple structure.

Lemma 2. *The game graph of $\mathcal{G}'$ is acyclic.*

Proof. Towards a contradiction, assume that G' contains a cycle $v_1, v_2, \ldots, v_\ell$. Let us pick i such that $\Omega(v_i)$ is minimal; since the length ℓ must be even and greater than two, we can assume without loss that $1 < i < \ell$. Suppose that $v_i \in V_{\mathrm{E}}$. As $(v_{i-1}, v_i) \in E'$, it follows that $\Omega(v_i) = \min(\Omega(v_{i-1}), \Omega(v_i))$ must be odd. On the other hand, as $(v_i, v_{i+1}) \in E'$, it follows that $\Omega(v_i) = \min(\Omega(v_i), \Omega(v_{i+1}))$ must be even: a contradiction. The argument for the case when $v_i \in V_{\mathrm{A}}$ is analogous. □

Parity games over acyclic graphs admit only finite plays: a player wins if, and only if, he can ensure that every play reaches a dead-end belonging to the other player. Thus, we have a so-called *reachability* game, for which it is well known that winning regions can be computed in linear time in the size of the underlying graph (see, e.g., [4]). Together with the equivalence in Lemma 1, this implies the following.

Proposition 1. *For any bipartite parity game on an undirected graph, the winning regions can be computed in linear time in the size of the graph.*

4. Games on arbitrary, undirected graphs can be hard

In this section, we show that restricting parity games to undirected graphs does not make them computationally simpler. Towards this, we give a polynomial-time reduction for the problem of computing the winning regions in parity games on arbitrary directed graphs to the corresponding problem on general, not necessarily bipartite, undirected graphs.

For the following, let us fix be a parity game $\mathcal{G} = (V, E, V_{\mathrm{E}}, V_{\mathrm{A}}, \Omega)$ on an arbitrary, directed graph $G = (V, E)$. The idea is to encode the directions of edges in E in an undirected graph. Towards this, we first bring $\mathcal{G}$ into a normalised form (see Figure 4 for an example).

Lemma 3. *There exists a game $\mathcal{G}' = (V', E', V'_{\mathrm{E}}, V'_{\mathrm{A}}, \Omega)$, with $V \subseteq V'$ such that a vertex $v \in V$ is winning for Éloïse in $\mathcal{G}$ if, and only of, v is winning for Éloïse in $\mathcal{G}'$, which satisfies the following properties:*

(1) *$\mathcal{G}'$ is bipartite;*
(2) *for every $v' \in V'$, the priority $\Omega(v')$ is odd if, and only if $v' \in V'_{\mathrm{E}}$, and*
(3) *the size of $G' = (V', E')$ is linear in the size of G.*

Proof. To ensure properties (1) and (2), we describe a simple transformation of the game graph in two steps, illustrated in Figure 3.

(Step 1) Towards turning $\mathcal{G}$ into a bipartite game, we insert dummy vertices with an insignificant priority. Let k be the highest priority in the range of Ω. For every edge $(v_1, v_2) \in E$ with $v_1, v_2 \in V_{\mathrm{E}}$, we add a fresh vertex $x \in V_{\mathrm{A}}$ of priority k, and replace the edge (v_1, v_2) by the two edges (v_1, x) and (x, v_2). For edges $(v_1, v_2) \in E$ with $v_1, v_2 \in V_{\mathrm{A}}$ we proceed in the dual way (*i.e.* we let $x \in V_{\mathrm{E}}$).

(Step 2) To ensure that all vertices of odd priority belong to Éloïse and those of even priority to Abelard, we proceed as follows. Let k' be the least even number greater than any priority in the range of Ω. For any vertex $v \in V_{\mathrm{E}}$ of even priority, let v belong to V'_{A} and add two vertices v_{in} and v_{out} to V'_{E}, both of priority $k' + 1$. Then, add two edges (v_{in}, v) and (v, v_{out}), and replace any incoming edge $(x, v) \in E$ with an edge $(x, v_{in}) \in E'$ and any outgoing edge (v, x) with an edge $(v_{out}, x) \in E'$. For the vertices $v \in V_{\mathrm{A}}$ of odd priority, we perform the dual transformation. All other vertices and edges remain unchanged.

Clearly, the size of the resulting game graph $G' = (V', E')$ is linear in the size of G. It is an easy exercise to verify that the transformations preserve the membership of any position $v \in V$ in the winning region of Éloïse or Abelard. $\square$

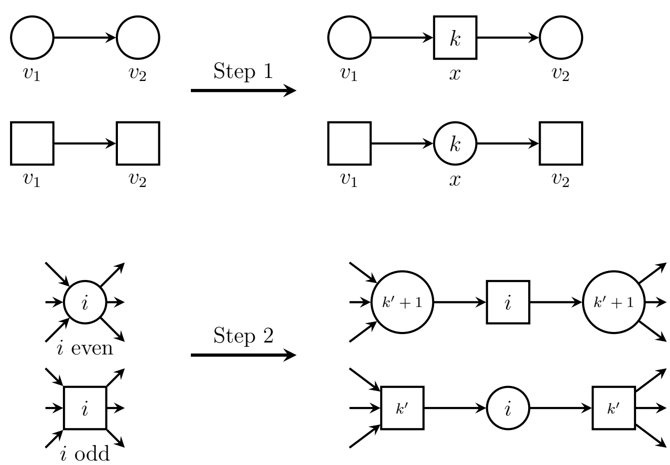

Figure 3: Normalisation rules (Lemma 3).

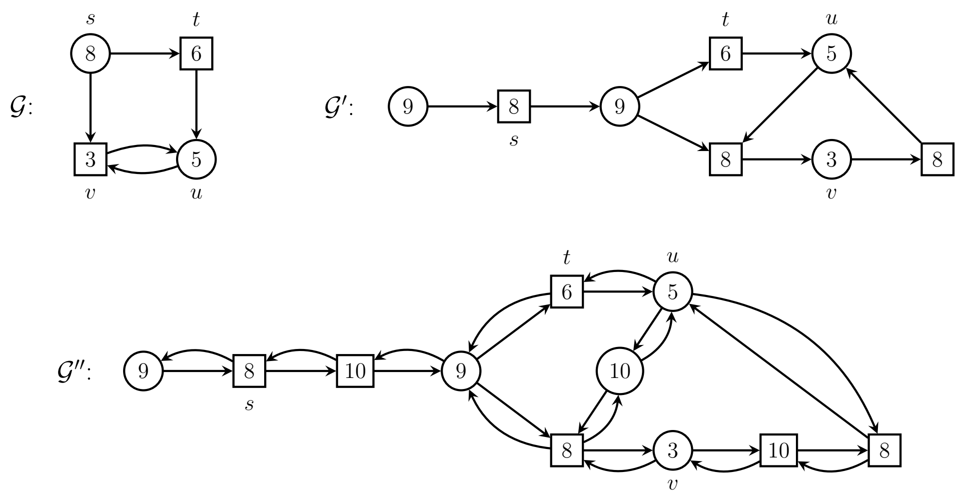

Figure 4: Normalisation ($\mathcal{G}'$) and direction encoding ($\mathcal{G}''$) for the game $\mathcal{G}$.

According to Lemma 3, we can assume without loss that the game $\mathcal{G}$ is bipartite, with vertices of odd priority belonging to V_{E} and those of even priority to V_{A}.

Next, we transform $\mathcal{G}$ into a game $\mathcal{G}' = (V', E', V'_{\mathrm{E}}, V'_{\mathrm{A}}, \Omega')$ on an undirected game graph $G' = (V', E')$ with $V' \supseteq V$ and prove that this transformation preserves the winning regions.

Let k be the maximal colour appearing in $\mathcal{G}$. The graph G' is obtained from G as follows (see Figure 5).

- For every $(v_1, v_2) \in E \cap V_{\mathrm{E}} \times V_{\mathrm{A}}$, let $i = \Omega(v_1)$ and $j = \Omega(v_2)$. If $i > j$, simply add the back-edge (v_2, v_1). If $i < j$, create a new vertex $x \in V'_{\mathrm{E}}$ of priority $k+1$, remove the edge (v_1, v_2), and add edges (v_1, x) and (x, v_2) together with their back-edges (x, v_1) and (v_2, x).

- For every $(v_1, v_2) \in E \cap V_{\mathrm{A}} \times V_{\mathrm{E}}$, let $i = \Omega(v_1)$ and $j = \Omega(v_2)$. If $i > j$, simply add the back-edge (v_2, v_1). If $i < j$, create a new vertex $x \in V'_{\mathrm{A}}$ of priority $k+1$, remove edge (v_1, v_2) and add the edges (v_1, x) and (x, v_2) together with their back-edges (x, v_1) and (v_2, x).

Then, the following holds.

Lemma 4. *A vertex $v \in V$ is winning for Éloïse in $\mathcal{G}$ if, and only if, v is winning for Éloïse in $\mathcal{G}'$.*

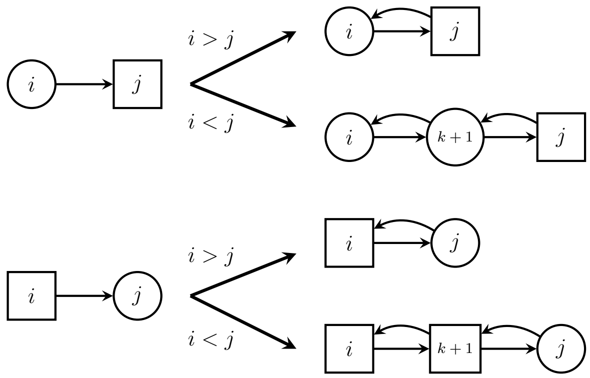

Figure 5: Transformation rules for encoding directions (Lemma 4)

Proof. Let σ be a positional winning strategy for Éloïse in $\mathcal{G}$ from v. We define a positional strategy σ' for Éloïse in $\mathcal{G}'$ as follows. For each $x \in V \cap V_{\mathrm{E}}$, let $y = \sigma(x)$. If $\Omega(x) > \Omega(y)$, the edge (x, y) exists in $\mathcal{G}'$, and we set $\sigma'(x) := y$. Otherwise, there is a unique vertex $z \in V' \cap V'_{\mathrm{E}}$ such that $\{(x, z), (z, y)\} \subseteq E'$, and we set $\sigma'(x) = z$ and $\sigma'(z) = y$. For any vertex in $V'_{\mathrm{E}} \setminus V_{\mathrm{E}}$ where σ' is not already defined by the above rules, choose an arbitrary successor along an edge in E'; such positions cannot be reached in a play starting from v that follows σ'. Note that the strategy σ' never uses a back-edge that was added when defining $\mathcal{G}'$.

We claim that σ' is winning from v. Assume otherwise that Abelard has a *positional* winning strategy τ from v, and let π be the unique play starting from v that follows σ' and τ. By the construction of G' and the assumption that τ is winning for Abelard, it follows that π never goes through any back-edge added in the construction of G': this would close a loop of length two and with even minimal priority, thus Abelard would never choose it. But this means that, if we remove from π all vertices from $V' \setminus V$, we obtain a valid play ρ in $\mathcal{G}$ from v that follows σ. As σ is winning for Éloïse in $\mathcal{G}$ from v, the play ρ would be also winning for her. But this contradicts the assumption that ρ is winning for Abelard, since, by construction of G', the least priority appearing infinitely often in ρ is the same as the one in π. Consequently, v is winning for Éloïse in $\mathcal{G}'$.

The dual argument shows that, if Abelard has a winning strategy in $\mathcal{G}$ from a vertex v, then he also has one in $\mathcal{G}'$. $\square$

As the size of G' is of polynomial in the size of G, Lemma 4 allows us to conclude.

Theorem 2. *The problem of computing the winning region in a parity game on a directed graph reduces in polynomial time to the corresponding problem for a parity game on an undirected graph. The reduction increases the number of priorities by two and and the size of the game graph by a constant factor.*

References

[1] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. Dag-width and parity games. In *Proceedings of Annual Symposium on Theoretical Aspects of Computer Science (STACS'06)*, volume 3884 of *LNCS*, pages 524–536. Springer, 2006.

[2] D. Berwanger and E. Grädel. Entanglement – a measure for the complexity of directed graphs with applications to logic and games. In *Proceedings of Logic for Programming and Automated Reasoning (LPAR'05)*, volume 3452 of *LNCS*, pages 209–223. Springer, 2005.

[3] E. A. Emerson and C. S. Jutla. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). In *Proceedings of Symposium on Foundations of Computer Science (FOCS'91)*, pages 368–377. IEEE, 1991.

[4] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.

[5] Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC'82)*, pages 60–65. ACM, 1982.

[6] P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoretical Computer Science*, 399(3):206–219, 2008.

[7] M. Jurdzinski, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM Journal on Computing*, 38(4):1519–1532, 2008.

[8] J. Obdrzalek. Fast Mu-calculus Model Checking when Tree-width is Bounded. In *Proceedings Computer Aided Verification (CAV'03)*, volume 2725 of *LNCS*, pages 80–92. Springer, 2003.

[9] J. Obdrzálek. Clique-width and parity games. In *Proceedings of Computer Science Logic (CSL'07)*, volume 4646 of *LNCS*, pages 54–68. Springer, 2007.