Graph Operations on Parity Games and Polynomial-Time Algorithms*

Christoph Dittmann, Stephan Kreutzer, Alexandru I. Tomescu**

Chair for Logic and Semantics, Technical University Berlin
christoph.dittmann@tu-berlin.de, stephan.kreutzer@tu-berlin.de,
alexandru.tomescu@mailbox.tu-berlin.de

1 Introduction

Parity games (see below) are a type of 2-player games that are studied in the area of formal verification of systems by model checking. Deciding the winner in a parity game is polynomial time equivalent to the model checking problem of the modal μ-calculus (e.g., [3]). Another strong motivation lies in the fact that the exact complexity of solving parity games is a long-standing open problem, the currently best known algorithm being subexponential [5]. It is known that the problem is in the complexity class UP $\cap$ coUP [4].

In this paper we identify restricted classes of digraphs where the problem is solvable in polynomial time, following an approach from structural graph theory. We consider three standard graph operations: the join of two graphs, repeated pasting along vertices, and the addition of a vertex. Given a class $\mathcal{C}$ of digraphs on which we can solve parity games in polynomial time, we show that the same holds for the class obtained from $\mathcal{C}$ by applying once any of these three operations to its elements.

These results provide, in particular, polynomial time algorithms for parity games whose underlying graph is a tournament (i.e., an orientation of a complete graph), a complete bipartite graph, a block graph, or a block-cactus graph. These are classes where the problem was not known to be efficiently solvable.

Previous results concerning restricted classes of parity games which are solvable in polynomial time include classes of bounded tree-width [7], bounded DAG-width [1], and bounded clique-width [8].

Notation and Preliminaries. A *parity game* $P = (V, V_\bigcirc, V_\Box, E, \Omega)$ is a finite directed graph (V, E) with a partitioning of the nodes $V = V_\bigcirc \cup V_\Box$ equipped with a priority map $\Omega : V \to \mathbb{N}$. A play on P starts with a token placed on some vertex $v \in V$. If $v \in V_\bigcirc$, Player $\circ$ moves the token to a successor of v, otherwise $V_\Box$ moves it to a successor. If there is no successor, the respective player loses. If the play continues forever, Player $\circ$ wins the game if and only if the maximum priority that appears infinitely often is even.

---

* We refer the interested reader to the full version [2] of this extended abstract.

** The third author gratefully acknowledges the support of the European Science Foundation, activity "Games for Design and Verification".

A *positional strategy* for Player $\circ$ is a map $\rho : V_{\bigcirc} \to V$ such that $\rho(v)$ is a successor of v for all v such that v has a successor. We only consider positional strategies in this paper. A play $v = v_0, v_1, v_2, \ldots$ *conforms* to ρ if $v_{i+1} = \rho(v_i)$ for all i such that $v_i \in V_{\bigcirc}$. A strategy ρ is a *winning strategy* for Player $\circ$ from vertex v if every play that starts at v and conforms to ρ is winning for Player $\circ$. We call the set of vertices $W_{\bigcirc}(P) \subseteq V$ from which Player $\circ$ has a positional winning strategy the *winning region* of Player $\circ$, similar for $W_{\Box}$ and Player $\Box$. We will write $W_{\bigcirc}$, $W_{\Box}$ if the game is clear from the context. Parity games are *positionally determined* in the sense that $W_{\bigcirc} \cup W_{\Box} = V$ and $W_{\bigcirc} \cap W_{\Box} = \emptyset$ [3].

Given $A \subseteq V$, we denote by $P \cap A$ the parity game restricted to the vertices in A, that is, $(V \cap A, V_{\bigcirc} \cap A, V_{\Box} \cap A, E \cap (A \times A), \Omega\restriction_A)$. Similarly, $P \setminus A$ stands for the game $P \cap (V \setminus A)$. Given a class of parity games $\mathcal{C}$, we say that $\mathcal{C}$ is *hereditary* if for all $P \in \mathcal{C}$ and all subsets A of vertices of P, we have $P \cap A \in \mathcal{C}$. If $i \in \{\circ, \Box\}$, we denote by $\bar{i}$ the element of $\{\circ, \Box\} \setminus \{i\}$.

2 Tournaments and Joins of Digraphs

We start by describing a polynomial-time algorithm for solving parity games on tournaments. In doing so, we observe that our algorithm can handle more general parity games. In particular, it can handle games with the sole requirement that between every vertex of Player $\circ$ and every vertex of Player $\Box$ there is an arc. This technique will then be generalized so that, as a very specific case, we obtain that parity games are solvable in polynomial-time on any biorientation of a complete bipartite graph. A *biorientation* of an undirected graph G is a directed graph G' with the same nodes as G such that for every edge $\{x, y\} \in E(G)$, the graph G' contains the arc (x, y), (y, x), or both.

We note that this result is not a special case of Obdržálek's polynomial time algorithm [8] for parity games of bounded directed clique-width because biorientations of complete graphs or complete bipartite graphs do not have bounded *directed* clique-width although their underlying undirected graphs have bounded clique-width.

We say that a digraph $D = (V, E)$, with a partition of its vertices $V = V_{\bigcirc} \cup V_{\Box}$, is a *weak tournament* if between every two vertices $v \in V_{\bigcirc}$, $w \in V_{\Box}$ we have that $(v, w) \in E$ or $(w, v) \in E$ (or both).

In Algorithm 1 on the next page, the function SOLVE-SINGLE-PLAYER-GAME solves single-player games in polynomial time (see [3]). We denote by $\mathrm{attr}_i(A)$ the set of vertices in V from which Player i has a strategy to enter A at least once and call it the *i-attractor set of A*. This notion is well-known [3] and stands at the basis of the exponential-time algorithms of McNaughton [6] and Zielonka [9].

Theorem 1. *Algorithm 1 correctly computes the winning regions of a parity game $P = (V, V_{\bigcirc}, V_{\Box}, E, \Omega)$ on a weak tournament and runs in time $O(|V|^4)$.*

Theorem 1 can be generalized to handle larger collections of digraphs, as long as the property that one of the winning regions induces a digraph on which we can efficiently solve parity games is maintained.

Algorithm 1: An algorithm for solving parity games on weak tournaments

Solve($P = (V, V_\bigcirc, V_\Box, E, \Omega)$)
 $(A_\bigcirc, A_\Box) \leftarrow$ Solve-Single-Player-Game($P \cap V_\bigcirc$)
 if $A_\bigcirc \neq \emptyset$ **then** $(W_\bigcirc, W_\Box) \leftarrow$ Solve($P \setminus A_\bigcirc$); **return** $(W_\bigcirc \cup A_\bigcirc, W_\Box)$
 $(B_\bigcirc, B_\Box) \leftarrow$ Solve-Single-Player-Game($P \cap V_\Box$)
 if $B_\Box \neq \emptyset$ **then** $(W_\bigcirc, W_\Box) \leftarrow$ Solve($P \setminus B_\Box$); **return** $(W_\bigcirc, W_\Box \cup B_\Box)$

 $d \leftarrow$ Maximum-Priority(Ω)
 $i \leftarrow \bigcirc$ **if** d is even, $\Box$ **otherwise**
 $(C_\bigcirc, C_\Box) \leftarrow$ Solve($P \setminus \mathrm{attr}_i(\Omega^{-1}(d))$)
 if $C_{\bar{i}} \neq \emptyset$ **then return** $(W_i \leftarrow \emptyset,\ W_{\bar{i}} \leftarrow V)$
 else return $(W_i \leftarrow V,\ W_{\bar{i}} \leftarrow \emptyset)$

If $P = (V, V_\bigcirc, V_\Box, E, \Omega)$ and $P' = (V', V'_\bigcirc, V'_\Box, E', \Omega')$ are two parity games with $V \cap V' = \emptyset$, we say that parity game $P'' = (V'', V''_\bigcirc, V''_\Box, E'', \Omega'')$ is a *join* of P and P' (see Figure 1) if

- $V'' = V \cup V'$, $V''_\bigcirc = V_\bigcirc \cup V'_\bigcirc$, $V''_\Box = V_\Box \cup V'_\Box$,
- $E'' = E \cup E' \cup E^*$, where $E^* \subseteq (V \times V') \cup (V' \times V)$ contains at least one arc (x, y) or (y, x) for all $x \in V$, $y \in V'$,
- and the vertices of P'' have the same priorities as they have in P and P'.

Given two classes of parity games $\mathcal{C}$ and $\mathcal{C}'$, we denote by $\mathsf{Join}(\mathcal{C}, \mathcal{C}')$ the class $\mathsf{Join}(\mathcal{C}, \mathcal{C}') := \{P'' \mid P'' \text{ is a join of } P \in \mathcal{C} \text{ and } P' \in \mathcal{C}'\}$.

Theorem 2. *If $\mathcal{C}$ and $\mathcal{C}'$ are hereditary classes of parity games that we can solve in polynomial time, then there is an algorithm for solving parity games in polynomial time on all games $P'' \in \mathsf{Join}(\mathcal{C}, \mathcal{C}')$, assuming a decomposition of P'' as a join of $P \in \mathcal{C}$ and $P' \in \mathcal{C}'$ is given.*

3 Pasting of Parity Games and Adding a Single Vertex

Let P, P' be two parity games on disjoint vertex sets and let v and v' be vertices of P and P', respectively. Assume that v, v' have the same priority and belong to the same player. We say that a game P'' is obtained by *pasting* P, P' at v, v' if P'' is the disjoint copy of P and P' with v, v' identified (see Figure 1). Given a class of parity games $\mathcal{C}$, we denote by $P(\mathcal{C})$ the class of games obtained by repeated pasting of a finite number of games from $\mathcal{C}$.

Theorem 3. *If $\mathcal{C}$ is a hereditary class of parity games that can be solved in polynomial time, then there is a polynomial time algorithm for solving parity games in $P(\mathcal{C})$.*

As a corollary of Theorems 1 and 3, we can solve parity games in polynomial time on any orientation of a *block-cactus graph*, that is, a graph whose maximal 2-connected components are cliques or cycles.

Our last result states that if P is a parity game and v a vertex such that $P \setminus \{v\}$ can be solved in polynomial time, then we can solve P in polynomial

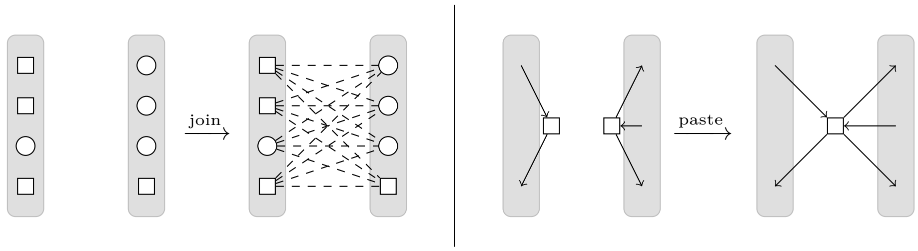

Fig. 1. The join and paste operations. The dashed lines represent necessary edges.

time. More formally, if $\mathcal{C}$ is a class of parity games, then $\mathcal{C}^+$ is the class obtained by adding a single vertex to every graph in $\mathcal{C}$ in any possible way.

Theorem 4. *If $\mathcal{C}$ is a hereditary class of games such that the decision problem (i.e., $P \in \mathcal{C}$?) is in polynomial time and games in $\mathcal{C}$ are solvable in polynomial time, then games in $\mathcal{C}^+$ are solvable in polynomial time.*

This theorem implies, for example, that if parity games can be solved in polynomial time on planar graphs, then they can also be solved in polynomial time on apex graphs, which are planar graphs with one additional vertex.

4 Conclusions

We have presented some graph operations that preserve the solvability of parity games in polynomial time. Generalizing this approach to more graph operations that generate larger classes of graphs is a possible line of future research.

References

1. Berwanger, D., Dawar, A., Hunter, P., Kreutzer, S.: DAG-Width and Parity Games. In: STACS. LNCS, vol. 3884, pp. 524–536. Springer (2006)
2. Dittmann, C., Kreutzer, S., Tomescu, A.I.: Graph operations on parity games and polynomial-time algorithms. pre-print arXiv:1208.1640 (2012)
3. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games: A Guide to Current Research, LNCS, vol. 2500. Springer (2002)
4. Jurdziński, M.: Deciding the winner in parity games is in UP ∩ co-UP. Inf. Process. Lett. 68(3), 119–124 (1998)
5. Jurdziński, M., Paterson, M., Zwick, U.: A deterministic subexponential algorithm for solving parity games. In: SODA '06. pp. 117–123 (2006)
6. McNaughton, R.: Infinite games played on finite graphs. Ann. Pure Appl. Logic 65(2), 149–184 (1993)
7. Obdržálek, J.: Fast Mu-Calculus Model Checking when Tree-Width Is Bounded. In: CAV '03. LNCS, vol. 2725, pp. 80–92. Springer (2003)
8. Obdržálek, J.: Clique-width and parity games. In: Duparc, J., Henzinger, T.A. (eds.) CSL '07. LNCS, vol. 4646, pp. 54–68. Springer (2007)
9. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theor. Comput. Sci. 200(1-2), 135–183 (1998)