



The Theory of Universal Graphs for Games: Past and Future

Nathanaël Fijalkow

► To cite this version:

Nathanaël Fijalkow. The Theory of Universal Graphs for Games: Past and Future. 15th International Workshop on Coalgebraic Methods in Computer Science (CMCS), Apr 2020, Dublin, Ireland. pp.1-10, 10.1007/978-3-030-57201-3_1 . hal-03232352

HAL Id: hal-03232352

<https://inria.hal.science/hal-03232352>

Submitted on 21 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

The Theory of Universal Graphs for Games: Past and Future

Nathanaël Fijalkow^{1,2}[0000–0002–6576–4680]

¹ CNRS, LaBRI, Université de Bordeaux, France
nathanael.fijalkow@labri.fr

² The Alan Turing Institute of data science, London, United Kingdom

Abstract. This paper surveys recent works about the notion of universal graphs. They were introduced in the context of parity games for understanding the recent quasipolynomial time algorithms, but they are defined for arbitrary objectives yielding a new approach for constructing efficient algorithms for solving different classes of games.

Keywords: Games · Parity games · Universal graphs.

1 The quasipolynomial era for parity games

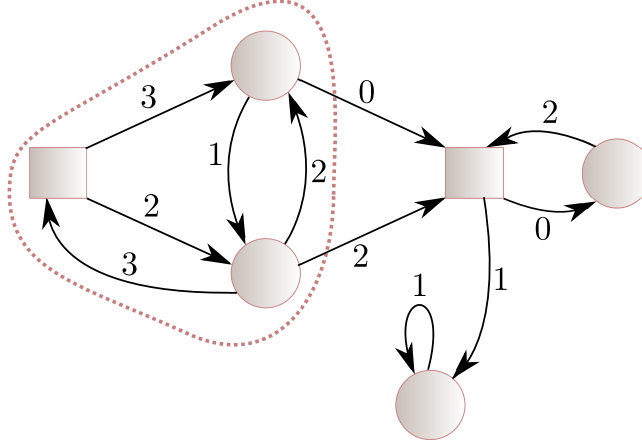


Fig. 1: A parity game. The vertices controlled by Eve are represented by circles and the ones controlled by Adam as squares. The edges are labelled by priorities inducing the parity objective. The dotted region is the winning region for Eve.

Parity games are a central model in the study of logic and automata over infinite trees and for their tight relationship with model-checking games for the modal μ -calculus. The central question left open for decades is whether there

exists a polynomial time algorithm for solving parity games. This section is a historical account on the progress made on this question since 2017.

The breakthrough result of Calude, Jain, Khoussainov, Li, and Stephan [2] was to construct a quasipolynomial time algorithm for solving parity games. Following decades of exponential and subexponential algorithms, this very surprising result triggered further research: soon after further quasipolynomial time algorithms were constructed reporting almost the same complexity, which is roughly $n^{O(\log d)}$. Let us classify them in two families.

The first family of algorithms includes the original algorithm by Calude, Jain, Khoussainov, Li, and Stephan [2] (see also [6] for a presentation of the algorithm as value iteration), then the succinct progress measure algorithm by Jurdziński and Lazić [9] (see also [7] for a presentation of the algorithm using universal trees explicitly) and the register games algorithm by Lehtinen [11] (see also [14] for a presentation of the algorithm using good-for-small-games automata explicitly). Bojańczyk and Czerwiński [1] introduced the separation question, describing a family of algorithms for solving parity games based on *separating automata*, and showed that the first quasipolynomial time algorithm yields a quasipolynomial solution to the separation question. Later Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, and Parys [5] showed that the other two algorithms also yield quasipolynomial solutions to the separation question. The main contribution of [5] is to show that any separating automaton contains a *universal tree* in its set of states; in other words, the three algorithms in this first family induce three (different) constructions of universal trees.

The second family of algorithms is so-called Zielonka-type algorithms, inspired by the exponential time algorithm by Zielonka [15]. The first quasipolynomial time algorithm is due to Parys [13], its complexity was improved by Lehtinen, Schewe, and Wojtczak [12]. Recently Jurdziński and Morvan [10] constructed a universal attractor decomposition algorithm encompassing all three algorithms: each algorithm is parametrised by the choice of two universal trees (one of each player).

All quasipolynomial time algorithms for parity games fall in one of two families, and both are based on the combinatorial notion of universal trees. This notion is by now well understood, with almost matching upper and lower bounds on the size of universal trees [7,5]. The lower bound implies a complexity barrier applying to both families of algorithms, hence to all known quasipolynomial time algorithms.

2 Universal graphs

Colcombet and Fijalkow [3,4] revisited the relationship between separating automata and universal trees by introducing the notions of *good-for-small-games automata* and of *universal graphs*.

The notion of good-for-small-games automata extends separating automata and resolves a disturbing situation that we describe now. The first three quasipolynomial time algorithms discussed above construct separating automata, but for

the third algorithm the automaton is *non-deterministic*. For that reason its correctness does not follow from the generic argument applying to the class of algorithms based on *deterministic* separating automata. (We note however that the lower bound result of [5] does apply to non-deterministic separating automata.) The notion of good-for-small-games automata resolves this conundrum, since the automaton constructed in Lehtinen's algorithm is indeed good-for-small-games and the generic argument for deterministic separating automata naturally extends to this class of automata³.

Universal trees arise in the study of the parity objective, the tree structure representing the nested behaviour of priorities. The notion of *universal graphs* extends universal trees from parity objectives to arbitrary (positionally determined) objectives. The main result of [4] is an equivalence result between *good-for-small-games automata* and *universal graphs*. More specifically, a good-for-small-games automaton induces a universal graph of the same size, and conversely. This equivalence extends the results of [5] relating separating automata and universal trees to any positionally determined objectives, so in particular for parity and mean payoff games.

The remainder of this section gives the formal definitions of universal graphs, good-for-small-games automata, and separating automata, their use for solving games, and state the equivalence result.

Universal graphs

We deliberately postpone the definitions related to games: indeed the notion of universal graphs is a purely combinatorial notion on graphs and ignores the interactive aspects of games.

Graphs We consider edge labelled directed graphs: a graph is given by a (finite) set V of vertices and a (finite) set $E \subseteq V \times C \times V$ of edges. A vertex v for which there exists no outgoing edge $(v, \ell, v') \in E$ is called a sink. We let n denote the number of vertices and m the number of edges. The size of a graph is its number of vertices.

Homomorphisms For two graphs G, G' , a homomorphism $\phi : G \rightarrow G'$ maps the vertices of G to the vertices of G' such that

$$(v, \ell, v') \in E \implies (\phi(v), \ell, \phi(v')) \in E'.$$

As a simple example that will be useful later on, note that if G' is a super graph of G , meaning they have the same set of vertices and every edge in G is also in G' , then the identity is a homomorphism $G \rightarrow G'$.

³ A closely related solution was given in [14] by giving syntactic properties implying that a non-deterministic automaton is good-for-small-games and showing that the automaton constructed in Lehtinen's algorithm has this property.

Paths A path π is a (finite or infinite) sequence of consecutive edges, where consecutive means that the third component of a triple in the sequence matches the first component of the next triple. In the case of a finite path we write $\text{last}(\pi)$ for the last vertex in π . We write $\pi = (v_0, \ell_0, v_1)(v_1, \ell_1, v_2) \cdots$ and let $\pi_{\leq i}$ denote the prefix of π of length i , meaning $\pi_{\leq i} = (v_0, \ell_0, v_1) \cdots (v_{i-1}, \ell_{i-1}, v_i)$.

Objectives An objective is a set $\Omega \subseteq C^\omega$ of infinite sequences of colours. A sequence of colours belonging to Ω is said to satisfy Ω . We say that a path in a graph satisfies Ω , or that it is winning when Ω is clear from context, if the sequence of labels it visits belongs to Ω . A path is said to be maximal if it is either infinite or ends in a sink.

We will work with a generic objective Ω . However it is convenient to assume that Ω is *prefix independent*. Formally, this means that $\Omega = C^* \cdot \Omega$, or equivalently for all $w \in C^*$ and $\rho \in C^\omega$, we have $\rho \in \Omega \iff w \cdot \rho \in \Omega$. This assumption can be lifted at the price of working with graphs with a distinguished initial vertex v_0 which must be preserved by homomorphisms.

Definition 1 (Graphs satisfying an objective). *Let Ω be a prefix independent objective. A graph satisfies Ω if all maximal paths are infinite and winning.*

Note that if a graph contains a sink, it does not satisfy Ω because it contains some finite maximal path.

Definition 2 (Universal graphs). *Let Ω be a prefix independent objective. A graph \mathcal{U} is (n, Ω) -universal if it satisfies the following two properties:*

1. *it satisfies Ω ,*
2. *for any graph G of size at most n satisfying Ω , there exists a homomorphism $\phi : G \rightarrow \mathcal{U}$.*

It is not clear that for any objective Ω and $n \in \mathbb{N}$ there exists an (n, Ω) -universal graph. Indeed, the definition creates a tension between “satisfying Ω ”, which restricts the structure, and “homomorphically mapping any graph of size at most n satisfying Ω ”, implying that the graph witnesses many different behaviours. A simple construction of an (n, Ω) -universal graph is to take the disjoint union of all (n, Ω) -graphs satisfying Ω . Since up to renaming of vertices there are finitely many such graphs this yields a very large but finite (n, Ω) -universal graph.

When considering a universal graph \mathcal{U} we typically let $V_{\mathcal{U}}$ and $E_{\mathcal{U}}$ denote the sets of vertices and edges of \mathcal{U} , respectively.

Solving games using universal graphs by reduction to safety games

We now introduce infinite duration games on graphs and construct a family of algorithms for solving games using universal graphs by reduction to safety games [1,4].

Arenas An arena is given by a graph *containing no sink* together with a partition $V_{\text{Eve}} \uplus V_{\text{Adam}}$ of its set V of vertices describing which player controls each vertex. We represent vertices controlled by Eve with circles and those controlled by Adam with squares.

Games A game is given by an arena and an objective. We often let \mathcal{G} denote a game, its size is the size of the underlying graph. It is played as follows. A token is initially placed on some initial vertex v_0 , and the player who controls this vertex pushes the token along an edge, reaching a new vertex; the player who controls this new vertex takes over, and this interaction goes on forever describing an infinite path.

Strategies We write Path_{v_0} for the set of finite paths of \mathcal{G} starting from v_0 and ending in V_{Eve} . A strategy for Eve is a map $\sigma : \text{Path}_{v_0} \rightarrow E$ such that for all $\pi \in \text{Path}_{v_0}$, $\sigma(\pi)$ is an edge in E from $\text{last}(\pi)$. Note that we always take the point of view of Eve, so a strategy implicitly means a strategy of Eve, and winning means winning for Eve. We say that an infinite path $\pi = (v_0, \ell_0, v_1)(v_1, \ell_1, v_2) \dots$ is consistent with the strategy σ if for all i , if $v_i \in V_{\text{Eve}}$, then $\sigma(\pi_{\leq i}) = (v_i, \ell_i, v_{i+1})$.

A strategy σ is winning from v_0 if all infinite paths starting from v_0 and consistent with σ are winning. Solving a game is the following decision problem:

INPUT: a game \mathcal{G} and an initial vertex v_0

OUTPUT: “yes” if Eve has a winning strategy from v_0 , “no” otherwise.

We say that v_0 is a winning vertex of \mathcal{G} when the answer to the above problem is positive. The winning region of Eve of \mathcal{G} is the set of winning vertices.

Positional strategies Positional strategies make decisions only considering the current vertex: $\sigma : V_{\text{Eve}} \rightarrow E$. A positional strategy induces a strategy $\hat{\sigma} : \text{Path}_{v_0} \rightarrow E$ by $\hat{\sigma}(\pi) = \sigma(\text{last}(\pi))$, where by convention the last vertex of the empty path is the initial vertex v_0 .

Definition 3 (Positionally determined objectives). *We say that an objective Ω is positionally determined if for every game with objective Ω and initial vertex v_0 , whenever there exists a winning strategy from v_0 then there exists a positional winning strategy from v_0 .*

Given a game \mathcal{G} , an initial vertex v_0 , and a positional strategy σ we let $\mathcal{G}[\sigma, v_0]$ denote the graph obtained by restricting \mathcal{G} to vertices reached by σ from v_0 and to the moves prescribed by σ . Formally, the set of vertices and edges is

$$\begin{aligned} V[\sigma, v_0] &= \{v \in V : \text{there exists a path from } v_0 \text{ to } v \text{ consistent with } \sigma\}, \\ E[\sigma, v_0] &= \{(v, \ell, v') \in E : v \in V_{\text{Adam}} \text{ or } (v \in V_{\text{Eve}} \text{ and } \sigma(v) = (v, \ell, v'))\} \\ &\quad \cap V[\sigma, v_0] \times C \times V[\sigma, v_0]. \end{aligned}$$

Fact 1 *Let Ω be a prefix independent objective, \mathcal{G} be a game, v_0 be an initial vertex, and σ a positional strategy. Then the strategy σ is winning from v_0 if and only if the graph $\mathcal{G}[\sigma, v_0]$ satisfies Ω .*

Safety games The safety objective **Safe** on two colours $C = \{\varepsilon, \text{Lose}\}$ is given by $\text{Safe} = \{\varepsilon^\omega\}$. In words, an infinite path is winning if it avoids the letter Lose. The following lemma is folklore.

Lemma 1. *Given a safety game with n vertices and m edges, there exists an algorithm running in time and space $O(n + m)$ which computes the winning region of Eve.*

Consider a game \mathcal{G} with objective Ω of size n and an (n, Ω) -universal graph \mathcal{U} , we construct a safety game $\mathcal{G} \triangleright \mathcal{U}$ as follows. The arena for the game $\mathcal{G} \triangleright \mathcal{U}$ is given by the following set of vertices and edges

$$\begin{aligned} V'_{\text{Eve}} &= V_{\text{Eve}} \times V_{\mathcal{U}} \uplus V \times V_{\mathcal{U}} \times C, \\ V'_{\text{Adam}} &= V_{\text{Adam}} \times V_{\mathcal{U}}, \\ E' &= \{((v, s), \varepsilon, (v', s, \ell)) : (v, \ell, v') \in E\} \\ &\cup \{((v', s, \ell), \varepsilon, (v', s')) : (s, \ell, s') \in E_{\mathcal{U}}\} \\ &\cup \{((v, s, \ell), \text{Lose}, (v, s, \ell)) : v \in V_{\mathcal{G}}, s \in V_{\mathcal{U}}, \ell \in C\}. \end{aligned}$$

In words: the game $\mathcal{G} \triangleright \mathcal{U}$ simulates \mathcal{G} . From (v, s) , the following two steps occur. First, the player who controls v picks an edge $(v, \ell, v') \in E$ as he would in \mathcal{G} which leads to (v', s, ℓ) , and second, Eve chooses which edge (s, ℓ, s') to follow in the universal graph \mathcal{U} which leads to (v', s') . If Eve is unable to play in \mathcal{U} (because there are no outgoing edges of the form (s, ℓ, s')), she is forced to choose $((v, s, \ell), \text{Lose}, (v, s, \ell))$ and lose the safety game.

Theorem 1. *Let Ω be a prefix independent positionally determined objective. Let \mathcal{G} be a game of size n with objective Ω and \mathcal{U} an (n, Ω) -universal graph. Let v_0 be an initial vertex. Then Eve has a winning strategy in \mathcal{G} from v_0 if and only if there exists a vertex s_0 in \mathcal{U} such that she has a winning strategy in $\mathcal{G} \triangleright \mathcal{U}$ from (v_0, s_0) .*

This theorem can be used to reduce games with objective Ω to safety games, yielding an algorithm whose complexity is proportional to the number of edges of \mathcal{U} . Indeed, recall that the complexity of solving a safety game is proportional to the number of vertices plus edges. The number of vertices of $\mathcal{G} \triangleright \mathcal{U}$ is $O(n \cdot n_{\mathcal{U}} |C|)$, and the number of edges is $O(n \cdot m_{\mathcal{U}})$ so the overall complexity is $O(n \cdot (n_{\mathcal{U}} |C| + m_{\mathcal{U}}))$.

Proof. Let us assume that Eve has a winning strategy σ in \mathcal{G} from v_0 , which can be chosen to be positional, Ω being positionally determined. We consider the graph $\mathcal{G}[\sigma, v_0]$: it has size at most n and since σ is winning from v_0 it satisfies Ω . Because \mathcal{U} is an (n, Ω) -universal graph there exists a homomorphism ϕ from $\mathcal{G}[\sigma, v_0]$ to \mathcal{U} . We construct a winning strategy in $\mathcal{G} \triangleright \mathcal{U}$ from $(v_0, \phi(v_0))$ by playing as in σ in the first component and following the homomorphism ϕ on the second component.

Conversely, let σ be a winning strategy from (v_0, s_0) in $\mathcal{G} \triangleright \mathcal{U}$. Consider the strategy σ' in \mathcal{G} which mimics σ by keeping an additional component in \mathcal{U} which is initially set to s_0 and updated following σ . The strategy σ' is winning since all infinite paths in \mathcal{U} satisfy Ω .

Good-for-small-games automata

The automata we consider are non-deterministic safety automata over infinite words on the alphabet C , where safety means that all states are accepting: a word is rejected if there exist no run for it. Formally, an automaton is given by a set of states Q , an initial state $q_0 \in Q$, and a transition relation $\Delta \subseteq Q \times C \times Q$. A run is a finite or infinite sequence of consecutive transitions starting with the initial state q_0 , where consecutive means that the third component of a triple in the sequence matches the first component of the next triple. We say that it is a run over a word if the projection of the run on the colours matches the word. An infinite word is accepted if it admits an infinite run. We let $L(\mathcal{A})$ denote the set of infinite words accepted by an automaton \mathcal{A} .

We introduce the notion of a *guiding strategy* for resolving non-determinism. A guiding strategy is a function $\sigma : C^+ \rightarrow \Delta$: given a finite word $\rho \in C^+$ it picks a transition. More specifically for an infinite word $\rho = c_0c_1 \dots \in C^\omega$ the guiding strategy chooses the following infinite sequence of transitions:

$$\sigma(c_0)\sigma(c_0c_1)\sigma(c_0c_1c_2) \dots$$

It may fail to be a run over ρ even if there exists one: the choice of transitions may require knowing the whole word ρ in advance, and the guiding strategy has to pick the transition only based on the current prefix. We say that the automaton guided by σ accepts a word ρ if the guiding strategy indeed induces an infinite run.

In the following we say that a path in a graph is accepted or rejected by an automaton; this is an abuse of language since what the automaton reads is only the sequence of colours of the corresponding path.

Definition 4. *An automaton is (n, Ω) -good-for-small-games if the two following properties hold:*

1. *there exists a guiding strategy σ such that for all graphs G of size n satisfying Ω , the automaton guided by σ accepts all paths in the graph G ,*
2. *the automaton rejects all paths not satisfying Ω .*

In a very similar way as universal graphs, good-for-small-games automata can be used to construct reductions to safety games.

Consider a game \mathcal{G} with objective Ω of size n and an (n, Ω) -good-for-small-games \mathcal{A} , we construct a safety game $\mathcal{G} \triangleright \mathcal{A}$ as follows. We let Q denote the set of states of \mathcal{A} and Δ the transition relation. The arena for the game $\mathcal{G} \triangleright \mathcal{A}$ is given by the following set of vertices and edges

$$\begin{aligned} V'_{\text{Eve}} &= V_{\text{Eve}} \times Q \uplus V_{\text{Eve}} \times Q \times C, \\ V'_{\text{Adam}} &= V_{\text{Adam}} \times Q, \\ E' &= \{((v, q), \varepsilon, (v', q, \ell)) : (v, \ell, v') \in E\} \\ &\cup \{((v', q, \ell), \varepsilon, (v', q')) : (q, \ell, q') \in \Delta\} \\ &\cup \{((v, q, \ell), \text{Lose}, (v, q, \ell)) : v \in V, q \in Q, \ell \in C\}. \end{aligned}$$

In words: the game $\mathcal{G} \triangleright \mathcal{A}$ simulates \mathcal{G} . From (v, q) , the following two steps occur. First, the player who controls v picks an edge $(v, \ell, v') \in E$ as he would in \mathcal{G} which leads to (v', q, ℓ) , and second, Eve chooses which transition (q, ℓ, q') to follow in the automaton \mathcal{A} which leads to (v', q') . If Eve is unable to choose a transition in \mathcal{A} (because there are no transitions of the form (q, ℓ, q')), she is forced to choose $((v, q, \ell), \text{Lose}, (v, q, \ell))$ and lose the safety game.

Theorem 2. *Let Ω be a prefix independent positionally determined objective. Let \mathcal{G} be a game of size n with objective Ω and \mathcal{A} an (n, Ω) -good-for-small-games automaton. Let v_0 be an initial vertex. Then Eve has a winning strategy in \mathcal{G} from v_0 if and only if she has a winning strategy in $\mathcal{G} \triangleright \mathcal{A}$ from (v_0, q_0) .*

As for universal graphs, this theorem can be used to reduce games with objective Ω to safety games, and the complexity is dominated by the number of transitions of \mathcal{A} .

Separating automata

Definition 5. *An (n, Ω) -separating automaton is a deterministic (n, Ω) -good-for-small-games automaton.*

In the deterministic case the first property reads: for all graphs G of size n satisfying Ω , the automaton accepts all paths in the graph G . The advantage of using deterministic automata is that the resulting safety game is smaller, and indeed the complexity of the algorithm above is proportional to the number of states of \mathcal{A} , more specifically it is $O(n \cdot |Q|)$ where Q is the set of states of \mathcal{A} .

The equivalence result

Theorem 3 ([4]). *Let Ω be a prefix independent positionally determined objective and n a natural number.*

- *For any (n, Ω) -separating automaton one can construct an (n, Ω) -good-for-small-games automaton of the same size.*
- *For any (n, Ω) -good-for-small-games automaton one can construct an (n, Ω) -universal graph of the same size.*
- *For any (n, Ω) -universal graph one can construct an (n, Ω) -separating automaton of the same size.*

The first item is trivial because by definition a separating automaton is a good-for-small-games automaton. The other two constructions are non-trivial, and in particular it is not true that a good-for-small-games automaton directly yields a universal graph as is, and the same holds conversely. Both constructions are based on a saturation method which constructs a linear order either on the set of states or on the set of vertices.

As a corollary of this equivalence result we can refine the reduction to safety games and present it in a value iteration style which has the same time complexity but also achieves a very small space complexity. We refer to [8] for more details.

3 Perspectives

The notion of universal graphs is a new tool for constructing algorithms for several classes of games by reducing them to safety games. For each objective, the question becomes whether there exist small universal graphs. A benefit of this approach is that this becomes a purely combinatorial question about graphs which does not involve reasoning about games.

Let us briefly discuss some of the results obtained so far, and give some perspectives for future research. The first example is parity objectives. In that case universal graphs have been combinatorially well understood: they are equivalently described as universal trees and their size is quasipolynomial with almost matching upper and lower bounds [7]. A more systematic study of ω -regular objectives reveals that similar ideas lead to small universal graphs for objectives such as Rabin or disjunctions of parity objectives. The second example is mean payoff objectives, which extend parity objectives. Upper and lower bounds have been obtained recently for this class of objectives, yielding new algorithms with (slightly) improved complexity [8].

In all examples mentioned above, the algorithms obtained using the universal graph technology match the best known complexity for solving these games.

Universal graphs for families of graphs

One can easily extend the theory of universal graphs to consider games over a given family of graphs. This motivates the question whether there exist small universal graphs for different families of graphs such as graphs of bounded tree or clique width.

Combining objectives: compositional constructions of universal graphs

An appealing question is whether universal graphs can be constructed compositionally. Indeed, it is often useful to consider objectives which are described as boolean combinations of objectives. Can we give generic constructions using universal graphs for basic objectives to build universal graphs for more complicated objectives? As examples, let us mention unions of mean payoff objectives (with both infimum limit or supremum limit semantics) and disjunctions of parity and mean payoff objectives.

References

1. Bojańczyk, M., Czerwiński, W.: An automata toolbox (February 2018), <https://www.mimuw.edu.pl/~bojan/papers/toolbox-reduced-feb6.pdf>
2. Calude, C.S., Jain, S., Khousainov, B., Li, W., Stephan, F.: Deciding parity games in quasipolynomial time. In: STOC. pp. 252–263 (2017). <https://doi.org/10.1145/3055399.3055409>

3. Colcombet, T., Fijalkow, N.: Parity games and universal graphs. CoRR **abs/1810.05106** (2018)
4. Colcombet, T., Fijalkow, N.: Universal graphs and good for games automata: New tools for infinite duration games. In: FoSSaCS. pp. 1–26 (2019). https://doi.org/10.1007/978-3-030-17127-8_1
5. Czerwiński, W., Daviaud, L., Fijalkow, N., Jurdziński, M., Lazić, R., Parys, P.: Universal trees grow inside separating automata: Quasi-polynomial lower bounds for parity games. CoRR **abs/1807.10546** (2018)
6. Fearnley, J., Jain, S., Schewe, S., Stephan, F., Wojtczak, D.: An ordered approach to solving parity games in quasi polynomial time and quasi linear space. In: SPIN. pp. 112–121 (2017)
7. Fijalkow, N.: An optimal value iteration algorithm for parity games. CoRR **abs/1801.09618** (2018)
8. Fijalkow, N., Gawrychowski, P., Ohlmann, P.: The complexity of mean payoff games using universal graphs. CoRR **abs/1812.07072** (2018)
9. Jurdziński, M., Lazić, R.: Succinct progress measures for solving parity games. In: LICS. pp. 1–9 (2017)
10. Jurdziński, M., Morvan, R.: A universal attractor decomposition algorithm for parity games. CoRR **abs/2001.04333** (2020)
11. Lehtinen, K.: A modal- μ perspective on solving parity games in quasi-polynomial time. In: LICS. pp. 639–648 (2018)
12. Lehtinen, K., Schewe, S., Wojtczak, D.: Improving the complexity of Parys’ recursive algorithm. CoRR **abs/1904.11810** (2019)
13. Parys, P.: Parity games: Zielonka’s algorithm in quasi-polynomial time. In: MFCS. pp. 10:1–10:13 (2019). <https://doi.org/10.4230/LIPIcs.MFCS.2019.10>
14. Parys, P.: Parity games: Another view on lehtinen’s algorithm. In: CSL. pp. 32:1–32:15 (2020). <https://doi.org/10.4230/LIPIcs.CSL.2020.32>
15. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theoretical Computer Science **200**(1-2), 135–183 (1998). [https://doi.org/10.1016/S0304-3975\(98\)00009-7](https://doi.org/10.1016/S0304-3975(98)00009-7)