The Strahler Number of a Parity Game

Laure Daviaud
Department of Computer Science, City, University of London, UK
Laure.Daviaud@city.ac.uk

Marcin Jurdziński
Department of Computer Science, University of Warwick, UK
Marcin.Jurdzinski@warwick.ac.uk

K. S. Thejaswini
Department of Computer Science, University of Warwick, UK
Thejaswini.Raghavan.1@warwick.ac.uk

Abstract

The Strahler number of a rooted tree is the largest height of a perfect binary tree that is its minor. The Strahler number of a parity game is proposed to be defined as the smallest Strahler number of the tree of any of its attractor decompositions. It is proved that parity games can be solved in quasi-linear space and in time that is polynomial in the number of vertices n and linear in $(d/2k)^k$, where d is the number of priorities and k is the Strahler number. This complexity is quasi-polynomial because the Strahler number is at most logarithmic in the number of vertices. The proof is based on a new construction of small Strahler-universal trees.

It is shown that the Strahler number of a parity game is a robust, and hence arguably natural, parameter: it coincides with its alternative version based on trees of progress measures and—remarkably—with the register number defined by Lehtinen (2018). It follows that parity games can be solved in quasi-linear space and in time that is polynomial in the number of vertices and linear in $(d/2k)^k$, where k is the register number. This significantly improves the running times and space achieved for parity games of bounded register number by Lehtinen (2018) and by Parys (2020).

The running time of the algorithm based on small Strahler-universal trees yields a novel trade-off $k \cdot \lg(d/k) = O(\log n)$ between the two natural parameters that measure the structural complexity of a parity game, which allows solving parity games in polynomial time. This includes as special cases the asymptotic settings of those parameters covered by the results of Calude, Jain Khoussainov, Li, and Stephan (2017), of Jurdziński and Lazić (2017), and of Lehtinen (2018), and it significantly extends the range of such settings, for example to $d = 2^{O\left(\sqrt{\lg n}\right)}$ and $k = O\left(\sqrt{\lg n}\right)$.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Theory and algorithms for application domains; Theory of computation → Design and analysis of algorithms; Theory of computation → Logic and verification

Keywords and phrases parity game, attractor decomposition, progress measure, universal tree, Strahler number

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.123

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Acknowledgements This work has been supported by the EPSRC grant EP/P020992/1 (Solving Parity Games in Theory and Practice). It has started when the third author was supported by the Department of Computer Science at the University of Warwick as a visiting student from Chennai Mathematical Institute, India. We thank colleagues in the following human chain for contributing to our education on the Strahler number of a tree: the first and the second authors have learnt it from the third author, who has learnt it from K Narayan Kumar, who has learnt it from Dmitry Chistikov, who has learnt it from Radu Iosif and Stefan Kiefer at a workshop on Infinite-State Systems, which has been organized and hosted by Joël Ouaknine and Prakash Panangaden at Bellairs Research Institute.

1 Context

Parity Games. Parity games are a fundamental model in automata theory and logic [8, 32, 17, 2], and their applications to verification, program analysis, and synthesis. In particular, they are intimately linked to the problems of emptiness and complementation of non-deterministic automata on trees [8, 32], model checking and satisfiability of fixpoint logics [9, 2], and evaluation of nested fixpoint expressions [1, 18]. It is a long-standing open problem whether parity games can be solved in polynomial time [9].

The impact of parity games goes well beyond their home turf of automata theory, logic, and formal methods. For example, an answer [14] of a question posed originally for parity games [31] has strongly inspired major breakthroughs on the computational complexity of fundamental algorithms in stochastic planning [12] and linear optimization [15, 16].

Strahler Number. The Strahler number has been proposed by Horton (1945) and made rigorous by Strahler (1952), in their morphological study of river networks in hydrogeology. It has been also studied in other sciences, such as botany, anatomy, neurophysiology, physics, and molecular biology, where branching patterns appear. The Strahler number has been identified in computer science by Ershov [10] as the smallest number of registers needed to evaluate an arithmetic expression. It has since been rediscovered many times in various areas of computer science; see the surveys of Knuth [23], Viennot [30], and Esparza, Luttenberger, and Schlund [11].

Related Work. A major breakthrough in the quest for a polynomial-time algorithm for parity games was achieved by Calude, Jain, Khoussainov, Li, and Stephan [3], who have given the first quasi-polynomial algorithm. Other quasi-polynomial algorithm have been developed soon after by Jurdziński and Lazić [20], and Lehtinen [24]. Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, and Parys [4] have introduced the concepts of *universal trees* and *separating automata*, and argued that all the aforementioned quasi-polynomial algorithms were intimately linked to them.

By establishing a quasi-polynomial lower bound on the size of universal trees, Czerwiński et al. have highlighted the fundamental limitations of the above approaches, motivating further the study of the attractor decomposition algorithm due to McNaughton [27] and Zielonka [32]. Parys [28] has proposed an ingenious quasi-polynomial version of McNaughton-Zielonka algorithm, but Lehtinen, Schewe, and Wojtczak [26], and Jurdziński and Morvan [21] have again strongly linked all quasi-polynomial variants of the attractor decomposition algorithm to universal trees.

Among several prominent quasi-polynomial algorithms for parity games, Lehtinen's approach [24] has relatively least attractive worst-case running time bounds. Parys [29] has offered some running-time improvements to Lehtinen's algorithm, but it remains significantly worse than state-of-the-art bounds of Jurdziński and Lazić [20], and Fearnley, Jain, de Keijzer, Schewe, Stephan, and Wojtczak [13], in particular because it always requires at least quasi-polynomial working space.

Our Contributions. We propose the Strahler number as a parameter that measures the structural complexity of dominia in a parity game and that governs the computational complexity of the most efficient algorithms currently known for solving parity games. We establish that the Strahler number is a robust, and hence natural, parameter by proving that it coincides with its version based on trees of progress measures and with the register number defined by Lehtinen [24].

We give a construction of small Strahler-universal trees that, when used with the progress measure lifting algorithm [19, 20] or with the universal attractor decomposition algorithm [21], yield algorithms that work in quasi-linear space and quasi-polynomial time. Moreover, usage of our small Strahler-universal trees allows to solve parity games in polynomial time for a wider range of asymptotic settings of the two natural structural complexity parameters (number of priorities d and the Strahler/register number k) than previously known, and that covers as special cases the $k = O(1)$ criterion of Lehtinen [24] and the $d < \lg n$ and $d = O(\log n)$ criteria of of Calude et al. [3], and of Jurdziński and Lazić [20], respectively.

Proofs. Proofs of some of our technical results can be found in the long version of this extended abstract available on arXiv [7].

2 Dominions, Attractor Decompositions, and Their Trees

Strategies, Traps, and Dominions. A *parity game* [8] $\mathcal{G}$ consists of a finite directed graph (V, E), a partition $(V_{\text{Even}}, V_{\text{Odd}})$ of the set of vertices V, and a function $\pi : V \to \{0, 1, \ldots, d\}$ that labels every vertex $v \in V$ with a non-negative integer $\pi(v)$ called its *priority*. We say that a cycle is *even* if the highest vertex priority on the cycle is even; otherwise the cycle is *odd*. We say that a parity game is (n, d)-*small* if it has at most n vertices and all vertex priorities are at most d.

For a set S of vertices, we write $\mathcal{G} \cap S$ for the substructure of $\mathcal{G}$ whose graph is the subgraph of (V, E) induced by the sets of vertices S. Sometimes, we also write $\mathcal{G} \setminus S$ to denote $\mathcal{G} \cap (V \setminus S)$. We assume throughout that every vertex has at least one outgoing edge, and we reserve the term *subgame* to substructures $\mathcal{G} \cap S$, such that every vertex in the subgraph of (V, E) induced by S has at least one outgoing edge.

A (positional) *Steven strategy* is a set $\sigma \subseteq E$ of edges such that:
- for every $v \in V_{\text{Even}}$, there is an edge $(v, u) \in \sigma$,
- for every $v \in V_{\text{Odd}}$, if $(v, u) \in E$ then $(v, u) \in \sigma$.

For a non-empty set of vertices R, we say that a Steven strategy σ *traps Audrey in* R if $w \in R$ and $(w, u) \in \sigma$ imply $u \in R$. We say that a set of vertices R is a *trap for Audrey* [32] if there is a Steven strategy that traps Audrey in R. Observe that if R is a trap in a game $\mathcal{G}$ then $\mathcal{G} \cap R$ is a subgame of $\mathcal{G}$. For a set of vertices $D \subseteq V$, we say that a Steven strategy σ is a *Steven dominion strategy on* D if σ traps Audrey in D and every cycle in the subgraph (D, σ) is even. Finally, we say that a set D of vertices is a *Steven dominion* [22] if there is a Steven dominion strategy on it.

Audrey strategies, trapping Steven, and Audrey dominions are defined in an analogous way by swapping the roles of the two players. We note that the sets of Steven dominions and of Audrey dominions are each closed under union, and hence the largest Steven and Audrey dominions exist, and they are the unions of all Steven and Audrey dominions, respectively. Moreover, every Steven dominion is disjoint from every Audrey dominion.

Attractor Decompositions. In a parity game $\mathcal{G}$, for a target set of vertices B ("bullseye") and a set of vertices A such that $B \subseteq A$, we say that a Steven strategy σ is a *Steven reachability strategy to* B *from* A if every infinite path in the subgraph (V, σ) that starts from a vertex in A contains at least one vertex in B.

For every target set B, there is the largest (with respect to set inclusion) set from which there is a Steven reachability strategy to B in $\mathcal{G}$; we call this set the *Steven attractor to B in $\mathcal{G}$* [32]. *Audrey reachability strategies* and *Audrey attractors* are defined analogously. We highlight the simple fact that if A is an attractor for a player in $\mathcal{G}$ then its complement $V \setminus A$ is a trap for them.

If $\mathcal{G}$ is a parity game in which all priorities do not exceed a non-negative even number d then we say that $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ is a *Steven d-attractor decomposition* [5, 6, 21] of $\mathcal{G}$ if:

- A is the Steven attractor to the (possibly empty) set of vertices of priority d in $\mathcal{G}$;

and setting $\mathcal{G}_1 = \mathcal{G} \setminus A$, for all $i = 1, 2, \ldots, \ell$, we have:

- S_i is a non-empty trap for Audrey in $\mathcal{G}_i$ in which every vertex priority is at most $d-2$;
- $\mathcal{H}_i$ is a Steven $(d-2)$-attractor decomposition of subgame $\mathcal{G} \cap S_i$;
- A_i is the Steven attractor to S_i in $\mathcal{G}_i$;
- $\mathcal{G}_{i+1} = \mathcal{G}_i \setminus A_i$;

and the game $\mathcal{G}_{\ell+1}$ is empty. If $d = 0$ then we require that $\ell = 0$.

The following proposition states that if a subgame induced by a trap for Audrey has a Steven attractor decomposition then the trap is a Steven dominion. Indeed, a routine proof argues that the union of all the Steven reachability strategies, implicit in the attractors listed in the decomposition, is a Steven dominion strategy.

▶ **Proposition 1** ([32, 5, 21])**.** *If d is even, R is a trap for Audrey in $\mathcal{G}$, and there is a Steven d-attractor decomposition of $\mathcal{G} \cap R$, then R is a Steven dominion in $\mathcal{G}$.*

Attractor decompositions for Audrey can be defined in the analogous way by swapping the roles of players as expected, and then a dual version of the proposition holds routinely.

The following theorem implies that every vertex in a parity game is either in the largest Steven dominion or in the largest Audrey dominion—it is often referred to as the *positional determinacy theorem* for parity games.

▶ **Theorem 2** ([8, 27, 32, 21])**.** *For every parity game $\mathcal{G}$, there is a partition of the set of vertices into a trap for Audrey W_{Even} and a trap for Steven W_{Odd}, such that there is a Steven attractor decomposition of $\mathcal{G} \cap W_{\text{Even}}$ and an Audrey attractor decomposition of $\mathcal{G} \cap W_{\text{Odd}}$.*

Ordered Trees and Their Strahler Numbers. Ordered trees are defined inductively; the trivial tree $\langle\rangle$ is an ordered tree and so is a sequence $\langle T_1, T_2, \ldots, T_\ell\rangle$, where T_i is an ordered tree for every $i = 1, 2, \ldots, \ell$. The trivial tree has only one node called the root, which is a leaf; and a tree of the form $\langle T_1, T_2, \ldots, T_\ell\rangle$ has the root with k children, the root is not a leaf, and the i-th child of the root is the root of ordered tree T_i.

Because the trivial tree $\langle\rangle$ has just one node, we sometimes write $\circ$ to denote it. If T is an ordered tree and i is a positive integer, then we use the notation T^i to denote the sequence $T, T, \ldots, T$ consisting of i copies of tree T. Then the expression $\left\langle T^i \right\rangle = \langle T, \ldots, T\rangle$ denotes the tree whose root has i children, each of which is the root of a copy of T. We also use the $\cdot$ symbol to denote concatenation of sequences, which in the context of ordered trees can be interpreted as sequential composition of trees by merging their roots; for example, $\left\langle \left\langle \circ^3 \right\rangle\right\rangle \cdot \left\langle \circ^4, \langle\langle\circ\rangle\rangle^2 \right\rangle = \left\langle \left\langle \circ^3 \right\rangle, \circ^4, \langle\langle\circ\rangle\rangle^2 \right\rangle = \langle\langle\circ,\circ,\circ\rangle, \circ,\circ,\circ,\circ, \langle\langle\circ\rangle\rangle, \langle\langle\circ\rangle\rangle\rangle$.

For an ordered tree T, we write $\text{height}(T)$ for its *height* and $\text{leaves}(T)$ for its *number of leaves*, which are defined by the following routine induction: the trivial tree $\langle\rangle = \circ$ has 1 leaf and its height is 1; the number of leaves of tree $\langle T_1, T_2, \ldots, T_\ell\rangle$ is the sum of the numbers of leaves of trees $T_1, T_2, \ldots, T_\ell$; and its height is 1 plus the maximum height of trees $T_1, T_2,$

$\ldots, T_\ell$. For example, the tree $\left\langle \left\langle \circ^3 \right\rangle, \circ^4, \langle\langle\circ\rangle\rangle^2 \right\rangle$ has 9 leaves and height 4 We say that an ordered tree is (n,h)-*small* if it has at most n leaves and its height is at most h.

The *Strahler number* $\mathrm{Str}(T)$ of a tree T is defined to be the largest height of a perfect binary tree that is a minor of T. Alternatively, it can be defined by the following structural induction: the Strahler number of the trivial tree $\langle\rangle = \circ$ is 1; and if $T = \langle T_1, \ldots, T_\ell \rangle$ and m is the largest Strahler number of trees $T_1, \ldots, T_\ell$, then $\mathrm{Str}(T) = m$ if there is a unique i such that $\mathrm{Str}(T_i) = m$, and $\mathrm{Str}(T) = m+1$ otherwise. For example, we have $\mathrm{Str}\left(\left\langle \left\langle \circ^3 \right\rangle, \circ^4, \langle\langle\circ\rangle\rangle^2 \right\rangle\right) = 2$ because $\mathrm{Str}(\circ) = \mathrm{Str}(\langle\langle\circ\rangle\rangle) = 1$ and $\mathrm{Str}\left(\left\langle \circ^3 \right\rangle\right) = 2$.

▶ **Proposition 3.** *For every* (n,h)*-small tree* T*, we have* $\mathrm{Str}(T) \leq h$ *and* $\mathrm{Str}(T) \leq \lfloor \lg n \rfloor + 1$.

Trees of Attractor Decompositions. The definition of an attractor decomposition is inductive and we define an ordered tree that reflects the hierarchical structure of an attractor decomposition. If d is even and $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell) \rangle$ is a Steven d-attractor decomposition then we define the *tree of attractor decomposition* $\mathcal{H}$ [6, 21], denoted by $T_\mathcal{H}$, to be the trivial ordered tree $\langle\rangle$ if $\ell = 0$, and otherwise, to be the ordered tree $\langle T_{\mathcal{H}_1}, T_{\mathcal{H}_2}, \ldots, T_{\mathcal{H}_\ell} \rangle$, where for every $i = 1, 2, \ldots, \ell$, tree $T_{\mathcal{H}_i}$ is the tree of attractor decomposition $\mathcal{H}_i$. Trees of Audrey attractor decompositions are defined analogously.

Observe that the sets $S_1, S_2, \ldots, S_\ell$ in an attractor decomposition as above are non-empty and pairwise disjoint, which implies that trees of attractor decompositions are small relative to the number of vertices and the number of distinct priorities in a parity game. The following proposition can be proved by routine structural induction.

▶ **Proposition 4** ([6, 21])**.** *If* $\mathcal{H}$ *is an attractor decomposition of an* (n,d)*-small parity game then its tree* $T_\mathcal{H}$ *is* $(n, \lceil d/2 \rceil + 1)$*-small.*

We define the *Strahler number of an attractor decomposition* $\mathcal{H}$, denoted by $\mathrm{Str}(\mathcal{H})$, to be the Strahler number $\mathrm{Str}(T_\mathcal{H})$ of its tree $T_\mathcal{H}$. We define the *Strahler number of a parity game* to be the maximum of the smallest Strahler numbers of attractor decompositions of the largest Steven and Audrey dominions, respectively.

3 Strahler Strategies in Register Games

This section establishes a connection between the register number of a parity game defined by Lehtinen [24] and the Strahler number. More specifically, we argue that from every Steven attractor decomposition of Strahler number k, we can derive a dominion strategy for Steven in the k-register game. Once we establish the Strahler number upper bound on the register number, we are faced with the following two natural questions:

▶ Question 5. Do the Strahler and the register numbers coincide?

▶ Question 6. Can the relationship between Strahler and register numbers be exploited algorithmically, in particular, to improve the running time and space complexity of solving register games studied by Lehtinen [24] and Parys [29]?

This work has been motivated by those two questions and it answers them both positively (Lemma 7 and Theorem 8, and Theorem 26, respectively).

For every positive number k, a Steven k*-register game* on a parity game $\mathcal{G}$ is another parity game $\mathcal{R}^k(\mathcal{G})$ whose vertices, edges, and priorities will be referred to as *states*, *moves*, and *ranks*, respectively, for disambiguation. The states of the Steven k-register game on $\mathcal{G}$ are either pairs $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle)$ or triples $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle, p)$, where v is a vertex in $\mathcal{G}$, $d \geq r_k \geq r_{k-1} \geq \cdots \geq r_1 \geq 0$, and $1 \leq p \leq 2k+1$. The former states have rank 1 and the latter have rank p. Each number r_i, for $i = k, k-1, \ldots, 1$, is referred to as the value of the i-th register in the state. Steven owns all states $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle)$ and the owner of vertex v in $\mathcal{G}$ is the owner of states $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle, p)$ for every p. How the game is played by Steven and Audrey is determined by the available moves:

- at every state $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle)$, Steven picks i, such that $0 \leq i \leq k$, and *resets* registers $i, i-1, i-2, \ldots, 1$, leading to state $\left(v, \langle r'_k, \ldots, r'_{i+1}, r'_i, 0, \ldots, 0 \rangle, p\right)$ of rank p and with updated register values, where:

$$p = \begin{cases} 2i & \text{if } i \geq 1 \text{ and } \max\left(r_i, \pi(v)\right) \text{ is even,} \\ 2i+1 & \text{if } i = 0, \text{ or if } i \geq 1 \text{ and } \max\left(r_i, \pi(v)\right) \text{ is odd;} \end{cases}$$

  $r'_j = \max(r_j, \pi(v))$ for $j \geq i+1$, and $r'_i = \pi(v)$;
- at every state $(v, \langle r_k, r_{k-1}, \ldots, r_1 \rangle, p)$, the owner of vertex v in $\mathcal{G}$ picks an edge (v, u) in $\mathcal{G}$, leading to state $(u, \langle r_k, r_{k-1}, \ldots, r_1 \rangle)$ of rank 1 and with unchanged register values.

For example, at state $(v, \langle 9, 6, 4, 4, 3 \rangle)$ of rank 1, if the priority $\pi(v)$ of vertex v is 5 and Steven picks $i = 3$, this leads to state $(v, \langle 9, 6, 5, 0, 0 \rangle, 7)$ of rank $2i + 1 = 7$ because $\max(r_3, \pi(v)) = \max(4, 5) = 5$ is odd, $r'_4 = \max(r_4, \pi(v)) = \max(6, 5) = 6$, and $r'_3 = \pi(v) = 5$.

Observe that the first components of states on every cycle in game $\mathcal{R}^k(\mathcal{G})$ form a (not necessarily simple) cycle in parity game $\mathcal{G}$; we call it the cycle in $\mathcal{G}$ *induced* by the cycle in $\mathcal{R}^k(\mathcal{G})$. If a cycle in $\mathcal{R}^k(\mathcal{G})$ is even (that is, the highest state rank on it is even) then the induced cycle in $\mathcal{G}$ is also even. Lehtinen [24, Lemmas 3.3 and 3.4] has shown that a vertex v is in the largest Steven dominion in $\mathcal{G}$ if and only if there is a positive integer k such that a state $(v, \overline{r})$, for some register values $\overline{r}$ is in the largest Steven dominion in $\mathcal{R}^k(\mathcal{G})$. Lehtinen and Boker [25, a comment after Definition 3.1] have further clarified that for every k, if a player has a dominion strategy in $\mathcal{R}^k(\mathcal{G})$ from a state whose first component is a vertex v in $\mathcal{G}$, then they also have a dominion strategy in $\mathcal{R}^k(\mathcal{G})$ from every state whose first component is v. This allows us to say without loss of rigour that a vertex v in $\mathcal{G}$ is in a dominion in $\mathcal{R}^k(\mathcal{G})$.

By defining the *(Steven) register number* [24, Definition 3.5] of a parity game $\mathcal{G}$ to be the smallest number k such that all vertices v in the largest Steven dominion in $\mathcal{G}$ are in a Steven dominion in $\mathcal{R}^k(\mathcal{G})$, and by proving the $1 + \lg n$ upper bound on the register number of every (n, d)-small parity game [24, Theorem 4.7], Lehtinen has contributed a novel quasi-polynomial algorithm for solving parity games, adding to those by Calude et al. [3] and Jurdziński and Lazić [20].

Lehtinen [24, Definition 4.8] has also considered the concept of a Steven *defensive dominion strategy* in a k-register game (for brevity, we call it a k-defensive strategy): it is a Steven dominion strategy on a set of states in $\mathcal{R}^k(\mathcal{G})$ in which there is no state of rank $2k+1$. Alternatively, the same concept can be formalized by defining the *defensive k-register game* $\mathcal{D}^k(\mathcal{G})$, which is played exactly like the k-register game $\mathcal{R}^k(\mathcal{G})$, but in which Audrey can also win just by reaching a state of rank $2k+1$. Note that the game $\mathcal{D}^k(\mathcal{G})$ can be thought of as having the winning criterion for Steven as being a conjunction of a parity and a safety criteria, and the winning criterion for Audrey as a disjunction of a parity and a reachability criteria. Routine arguements allow to extend positional determinacy from parity games to such games with combinations of parity, and safety or reachability winning criteria.

We follow Lehtinen [24, Definition 4.9] by defining the *(Steven) defensive register number* of a Steven dominion D in $\mathcal{G}$ as the smallest number k such that Steven has a defensive dominion strategy in $\mathcal{R}^k(\mathcal{G})$ on a set of states that includes all $(v, \langle r_k, \ldots, r_1 \rangle)$ for $v \in D$, and such that r_k is an even number at least as large as every vertex priority in D. We propose to call it the *Lehtinen number* of a Steven dominion in $\mathcal{G}$ to honour Lehtinen's insight that led to this—as we argue in this work—fundamental concept. We also define the Lehtinen number of a vertex in $\mathcal{G}$ to be the smallest Lehtinen number of a Steven dominion in $\mathcal{G}$ that includes the vertex, and the Lehtinen number of a parity game to be the Lehtinen number of its largest Steven dominion. We also note that the register and the Lehtinen numbers of a parity game nearly coincide (they differ by at most one), and hence the conclusions of our analysis of the latter also apply to the former.

▶ **Lemma 7.** *The Lehtinen number of a parity game is no larger than its Strahler number.*

The arguments used in our proof of this lemma are similar to those used in the proof of the main result of Lehtinen [24, Theorem 4.7]. Our contribution here is to pinpoint the Strahler number of an attractor decomposition as the structural parameter of a dominion that naturally bounds the number of registers used in Lehtinen's construction of a defensive dominion strategy.

Proof of Lemma 7. Consider a parity game $\mathcal{G}$ and let d be the least even integer no smaller than any of the priority in $\mathcal{G}$. Consider a Steven d-attractor decomposition $\mathcal{H}$ of $\mathcal{G}$ of Strahler number k. We construct a defensive k-register strategy for Steven on $\mathcal{R}^k(\mathcal{G})$. The strategy is defined inductively on the height of $\mathcal{T}_{\mathcal{H}}$, and has the additional property of being $\mathcal{G}$*-positional* in the following sense: if $((v, \langle r_k, \ldots, r_1\rangle), (v, \langle r'_k, \ldots, r'_1\rangle, p))$ is a move then the register reset by Steven only depends on v, not on the values in the registers. Similarly, if $((v, \langle r_k, \ldots, r_1\rangle, p), (u, \langle r_k, \ldots, r_1\rangle))$ is a move and v is owned by Steven, u only depends on v and not on the values of the registers or p.

Strategy for Steven. If $\mathcal{H} = \langle A, \emptyset\rangle$, then $\mathcal{G}$ consists of the set of vertices of priority d and of its Steven attractor. In this case, Steven follows the strategy induced by the reachability strategy in A to the set of vertices of priority d, only resetting register r_1 immediately after visiting a state with first component a vertex of priority d in $\mathcal{G}$. More precisely, the Steven defensive strategy is defined with the following moves:

- $((v, \langle r_1\rangle), (v, \langle r_1\rangle, 1))$ if v is not a vertex of priority d in $\mathcal{G}$;
- $((v, \langle r_1\rangle), (v, \langle r'_1\rangle, 2))$ if v is a vertex of priority d in $\mathcal{G}$ and $r'_1 = \max(r_1, d)$ is even;
- $((v, \langle r_1\rangle), (v, \langle r'_1\rangle, 3))$ if v is a vertex of priority d in $\mathcal{G}$ and $r'_1 = \max(r_1, d)$ is odd (we state this case for completeness but this will never occur);
- $((v, \langle r_1\rangle, p), (u, \langle r_1\rangle))$ where (v, u) belongs to the Steven reachability strategy from A to the set of vertices of priority d in $\mathcal{G}$.

Note that this strategy is $\mathcal{G}$-positional.

Suppose now that $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ and that it has Strahler number k. For all $i = 1, 2, \ldots, \ell$, let k_i be the Strahler number of $\mathcal{H}_i$. By induction, for all i, we have a Steven defensive k_i-register strategy σ_i, which is $(\mathcal{G} \cap S_i)$-positional, on a set of states Ω_i in $\mathcal{R}^{k_i}(\mathcal{G} \cap S_i)$ including all the states $(v, \langle r_{k_i}, \ldots, r_1\rangle)$ for $v \in S_i$ and r_{k_i} an even number at least as large as every vertex priority in S_i. Let Γ_i be the set of states in $\mathcal{R}^k(\mathcal{G} \cap S_i)$ defined as all the states $(v, \langle d, r_{k-1}, \ldots, r_1\rangle)$ for $v \in S_i$ if $k_i \neq k$ and as the union of the states $(v, \langle d, r_{k-1}, \ldots, r_1\rangle)$ for $v \in S_i$ and Ω_i, otherwise.

The strategy σ_i induces a strategy on Γ_i in $\mathcal{R}^k(\mathcal{G} \cap S_i)$ by simply ignoring registers $r_{k_i+1}, \ldots, r_k$, and using $(\mathcal{G} \cap S_i)$-positionality to define moves from the states not in Ω_i. More precisely, in a state $(v, \langle r_k, \ldots, r_1\rangle)$, Steven resets register j if and only if register j is reset in a state $\left(v, \left\langle r'_{k_i}, \ldots, r'_1\right\rangle\right)$ of Ω_i according to σ_i. This is well defined by $(\mathcal{G} \cap S_i)$-positionality. Similarly, we add moves $((v, \langle r_k, \ldots, r_1\rangle, p), (u, \langle r_k, \ldots, r_1\rangle))$ to the strategy if and only if there is a move $\left(\left(v, \left\langle r'_{k_i}, \ldots, r'_1\right\rangle, p'\right), \left(u, \left\langle r'_{k_i}, \ldots, r'_1\right\rangle\right)\right)$ in σ_i. This is again well-defined by $(\mathcal{G} \cap S_i)$-positionality.

This strategy is denoted by τ_i. Note that τ_i is a defensive k-register strategy on Γ_i, which is $\mathcal{G}$-positional.

The Steven defensive strategy in $\mathcal{R}^k(\mathcal{G})$ is defined by the following moves, where S denotes the set of vertices of priority d in $\mathcal{G}$:

- On the set of states with first component a vertex of $A_i \setminus S_i$, the moves are given by τ_i.
- On the set of states with first component a vertex of $A \setminus S$, Steven uses the strategy induced by the reachability strategy from A_i to S_i, without resetting any registers.
- On $\mathcal{R}^k(\mathcal{G} \cap (A \setminus S))$, Steven uses the strategy induced by the reachability strategy from A to S, without resetting any registers.
- On the set of states with first component a vertex of S,
  - $((v, \langle r_k, \ldots, r_1\rangle), (v, \langle d, 0, \ldots, 0\rangle, p))$ where v is a vertex in S and $p = 2k$ if $\max(r_k, d)$ is even and $p = 2k+1$ otherwise.
  - $((v, \langle r_k, \ldots, r_1\rangle, p), (u, \langle r_k, \ldots, r_1\rangle))$ for some uniquely chosen u such that (v, u) in E if v is owned by Steven and for all u such that (v, u) in E if v is owned by Audrey.

Observe that this strategy is $\mathcal{G}$-positional.

Correctness of the Strategy. We prove now that the strategy defined above is indeed a defensive k-register strategy. We proceed by induction on the height of $\mathcal{T}_{\mathcal{H}}$ and define a set of states Γ, including all the states $(v, \langle d, r_{k-1}, \ldots, r_1\rangle)$ such that v is a vertex of $\mathcal{G}$.

Base Case: If the height of $\mathcal{T}_{\mathcal{H}}$ is 0 and $\mathcal{H} = \langle A, \emptyset\rangle$, let Γ be the set of states $(v, \langle r_1\rangle)$ and $(v, \langle r_1\rangle, p)$ with v a vertex of $\mathcal{G}$, $1 \leq r_1 \leq d$ and p being either 1 or 2. It is easy to see that the strategy defined above is a defensive dominion strategy on this set.

Inductive step: If $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ with Strahler number k and k_i being the Strahler number of $\mathcal{H}_i$ for all i (note that $k_i \leq k$ for all i, and by definition of Strahler number, there is at most one m such that $k_m = k$), we define Γ to be the set comprising the union of the Γ_i and all the states of the form $(v, \langle r_k, \ldots, r_1\rangle)$ and $(v, \langle r_k, \ldots, r_1\rangle, p)$ with v a vertex of $(A_i \setminus S_i) \cup A$ and $1 \leq p \leq 2k$.

Case 1: For each i, $k_i < k$.

We first show that Γ is a trap for Audrey for the strategy defined above, showing that rank $2k+1$ can never be reached (implying that the strategy is defensive). This comes from the fact that the register of rank k is only reset in a state $(v, \langle r_k, \ldots, r_1\rangle)$ with v in S. Since $\max(r_k, d) = d$ is even then this leads to a state $(v, \langle d, 0, \ldots, 0\rangle, 2k)$. Otherwise, register k is never reset, so a state with rank $2k+1$ cannot be reached.

Consider now any cycle in $\mathcal{R}^k(\mathcal{G})$ with moves restricted to the strategy constructed above. If this cycle contains a state whose first component is a vertex of S, then as explained above, the highest rank in the cycle is $2k$. Otherwise, the cycle is necessarily in $\mathcal{R}^k(\mathcal{G} \cap S_i)$ for some i. By induction, τ_i is winning and so the cycle is even.

Case 2: There is a unique m such that $k_m = k$.

We first show that a state of rank $2k+1$ is never reached. Observe that register k is reset in two places: (1) immediately after a state with first component a vertex of S is visited, (2) if register k is reset by τ_m. In the first case, similarly as shown above, a state of rank $2k$ is reached. In the second case, register k is either reset in a state $(v, \langle d, r_{k-1}, \ldots, r_1\rangle)$, and similarly as above, a state of rank $2k$ is reached, or in a state of Ω_i. In this case, as τ_i is defensive on Ω_i by induction, a state of rank $2k+1$ cannot be reached, and the highest rank that can be reached is $2k$.

Proving that every cycle is even is similar to the previous case. ◀

4 Strahler-Optimal Attractor Decompositions

In this section we prove that every parity game whose Lehtinen number is k has an attractor decomposition of Strahler number at most k. In other words, we establish the Lehtinen number upper bound on the Strahler number, which together with Lemma 7 provides a positive answer to Question 5.

▶ **Theorem 8.** *The Strahler number of a parity game is no larger than its Lehtinen number.*

When talking about strategies in parity games in Section 2, we only considered positional strategies, for which it was sufficient to verify the parity criterion on (simple) cycles. Instead, we explicitly consider the parity criterion on infinite paths here, which we find more convenient to establish properties of Audrey strategies in the proof of Theorem 8.

First, we introduce the concepts of *tight* and *offensively optimal* attractor decompositions.

▶ **Definition 9.** *A Steven d-attractor decomposition $\mathcal{H}$ of $\mathcal{G}$ is* tight *if Audrey has a winning strategy from at least one state in $\mathcal{D}^{\mathrm{Str}(\mathcal{H})-1}(\mathcal{G})$ in which the value of register* $\mathrm{Str}(\mathcal{H})-1$ *is d.*

By definition, the existence of a tight Steven d-attractor decomposition on a parity game implies that the Lehtinen number of the game is at least its Strahler number, from which Theorem 8 follows. Offensive optimality of an attractor decomposition, the concept we define next, may seem less natural and more technical than tightness, but it facilitates our proof that every game has a tight attractor decomposition.

▶ **Definition 10.** *Let $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ be a Steven d-attractor decomposition, let games $\mathcal{G}_i$ for $i = 1, 2, \ldots, \ell$ be as in the definition of an attractor decomposition, let A'_i be the Audrey attractor of the set of vertices of priority $d-1$ in $\mathcal{G}_i$, and let $\mathcal{G}'_i = \mathcal{G}_i \setminus A'_i$. We say that $\mathcal{H}$ is* offensively optimal *if for every $i = 1, 2, \ldots, \ell$, we have:*

- *Audrey has a dominion strategy on $\mathcal{R}^{\mathrm{Str}(\mathcal{H}_i)-1}(\mathcal{G}'_i)$;*
- *Audrey has a dominion strategy on $\mathcal{D}^{\mathrm{Str}(\mathcal{H}_i)}(\mathcal{G}'_i \setminus S_i)$.*

Proving that every offensively optimal Steven attractor decomposition is tight (Lemma 11), and that every Steven dominion in a parity game has an offensively optimal Steven attractor decomposition (Lemma 12), will complete the proof of Theorem 8.

▶ **Lemma 11.** *Every offensively optimal Steven attractor decomposition is tight.*

Proof. Let $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ be an offensively optimal d-attractor decomposition of a parity game and let $k = \mathrm{Str}(\mathcal{H})$. We construct a strategy for Audrey in $\mathcal{D}^{k-1}(\mathcal{G})$ that is winning for her from at least one state in which the value of register $k-1$ is d. We define $\mathcal{G}'_i$ and A'_i as in Definition 10.

Case 1: $\mathrm{Str}(\mathcal{H}_i) = k$ for some unique i in $\{1, \ldots, \ell\}$. In this case, we show that Audrey has a dominion strategy on $\mathcal{D}^{k-1}(\mathcal{G}_i)$. Since $\mathcal{G}_i$ is a trap for Steven in $\mathcal{G}$, this gives the desired result. Consider the following strategy in $\mathcal{D}^{k-1}(\mathcal{G}_i)$:

- On the set of states whose vertex components are in A'_i, Audrey follows a strategy induced by the reachability strategy in A'_i to a vertex of priority $d-1$ (picking any move if v is of priority $d-1$);
- In states whose vertex component is in $\mathcal{G}'_i$, Audrey plays a $(k-1)$-register dominion strategy on $\mathcal{R}^{k-1}(\mathcal{G}'_i)$. Such a strategy exists by the definition of offensive optimality and by the assumption that $\mathrm{Str}(\mathcal{H}_i) = k$.

This strategy is indeed an Audrey dominion strategy on $\mathcal{D}^{k-1}(\mathcal{G}_i)$, because any play either visits a state whose first component is a vertex in A'_i infinitely often, or it eventually remains in $\mathcal{R}^{k-1}(\mathcal{G}'_i)$. In the former case, the play visits a state whose first component is a vertex of priority $d-1$ infinitely often. In the latter case, the state parity criterion holds. Note that this even defines an Audrey dominion strategy on $\mathcal{R}^{k-1}(\mathcal{G}_i)$.

Case 2: There are $1 \leq i < j \leq \ell$ such that $\mathrm{Str}(\mathcal{H}_i) = \mathrm{Str}(\mathcal{H}_j) = k-1$. We construct a strategy for Audrey in $\mathcal{D}^{k-1}(\mathcal{G})$ that is winning for her from all states in $\mathcal{G}_j$ whose register $k-1$ has value d. Firstly, since $\mathcal{H}$ is offensively optimal, Audrey has a dominion strategy on $\mathcal{D}^{k-1}(\mathcal{G}'_i \setminus S_i)$, denoted by τ_i, and a dominion strategy on $\mathcal{R}^{k-2}(\mathcal{G}'_i)$, denoted by τ'_i. Moreover, since $\langle \emptyset, (S_j, \mathcal{H}_j, A_j), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell)\rangle$ is an offensively optimal attractor decomposition of $\mathcal{G}_j$, an argument similar to the one in Case 1. yields that Audrey has a dominion strategy, denoted by τ_j, on $\mathcal{R}^{k-2}(\mathcal{G}_j)$ (note that $\mathcal{G}_j$ is a trap for Steven in $\mathcal{G}$). Consider the following strategy for Audrey in $\mathcal{D}^{k-1}(\mathcal{G})$, starting from a state whose vertex component is in $\mathcal{G}_j$ and register $k-1$ has value d:

- As long as the value of register $k-1$ is larger than $d-1$, Audrey follows the strategy induced by τ_j, while ignoring the value of register $k-1$, as long as this value is larger than $d-1$.
- If the value in register $k-1$ is at most $d-1$:
  - In states whose vertex component is in A'_i, Audrey follows a strategy induced by the reachability strategy from A'_i to a vertex of priority $d-1$ (picking any move if the vertex has priority $d-1$);
  - In states whose vertex component is in $\mathcal{G}'_i \setminus S_i$ and whose register $k-2$ has value at most $d-2$, Audrey follows τ_i;
  - In states whose vertex component is in $\mathcal{G}'_i$ and whose register $k-1$ has value $d-1$, Audrey follows the strategy induced by τ'_i, while ignoring the value of regiser $k-1$.

Audrey plays any move if none of the above applies.

We argue that this strategy is winning for Audrey in $\mathcal{D}^{k-1}(\mathcal{G})$ from states whose vertex component is in $\mathcal{G}_j$ and register $k-1$ has value d. Consider an infinite path that starts in such a state. As long as register $k-1$ has value d, Audrey follows τ_j. If Steven never resets register $k-1$ then Audrey wins. Otherwise, once register $k-1$ has been reset, its value is at most $d-1$. Note that $\mathcal{G}_j$ is included in $A'_i \cup (\mathcal{G}'_i \setminus S_i)$. If register $k-1$ has a value smaller than $d-1$, and the play never visits a state whose vertex component is in A'_i, then Audrey has followed τ_i along the play (she has never left $\mathcal{G}'_i \setminus S_i$ as the only way for Steven to go out $\mathcal{G}'_i \setminus S_i$ is to go to A'_i) and wins. Otherwise, the play visits a state whose vertex component is in A'_i, and so it visits a state whose vertex component has priority $d-1$, leading to a state in which register $k-1$ has value $d-1$. Finally, if a state whose vertex component is in A'_i is visited infinitely many times then Audrey wins. Otherwise, Audrey eventually plays according to τ'_i. If Steven never resets register $k-1$ then Audrey wins. Otherwise, if Steven resets register $k-1$, which at this point has value $d-1$, a state of rank $2k-1$ is visited and Audrey wins. ◀

▶ **Lemma 12.** *Every Steven dominion in a parity game has an offensively optimal Steven attractor decomposition.*

Proof. Consider a parity game $\mathcal{G}$ which is a Steven dominion. Let k be the Lehtinen number of $\mathcal{G}$ and let d be the largest even value such that $\pi^{-1}(\{d, d-1\}) \neq \emptyset$. We construct an offensively optimal Steven attractor decomposition by induction.

If $d = 0$, it is enough to consider $\langle A, \emptyset\rangle$, where A is the set of all vertices in $\mathcal{G}$.

If $d > 1$, let A be the Steven attractor of the set of vertices of priority d in $\mathcal{G}$. Let $\mathcal{G}_0 = \mathcal{G} \setminus A$. If $\mathcal{G}_0 = \emptyset$ then $\langle A, \emptyset \rangle$ is an offensively optimal Steven attractor decomposition for $\mathcal{G}$. Otherwise, $\mathcal{G}_0$ is a non-empty trap for Steven in $\mathcal{G}$ and therefore $\mathcal{G}_0$ has a Lehtinen number at most k. Let A' be the Audrey attractor of all the vertices of priority $d-1$ in the sub-game $\mathcal{G}_0$ and let $\mathcal{G}'_0 = \mathcal{G}_0 \setminus A'$.

Given a positive integer b, let L^b be the largest dominion in $\mathcal{G}'_0$ such that Steven has a dominion strategy on $\mathcal{D}^b(\mathcal{G}'_0)$. We define m to be the smallest number such that $L^m \neq \emptyset$ and let $S_0 = L^m$. We show that $m \leq k$. To prove this, we construct an Audrey dominion strategy on $\mathcal{D}^b(\mathcal{G}_0)$ for all b such that $L^b = \emptyset$. Since the Lehtinen number of $\mathcal{G}_0$ is at most k, this implies that $m \leq k$. The Audrey dominion strategy on $\mathcal{D}^b(\mathcal{G}_0)$, assuming $L^b = \emptyset$, is as follows:

- If the vertex component of a state is in A' then Audrey uses the strategy in A' induced by the reachability strategy to vertices of priority $d-1$;
- If the vertex component of a state is in $\mathcal{G}'_0$ then Audrey uses her dominion strategy on $\mathcal{D}^b(\mathcal{G}'_0)$, which exists because the Steven dominion L^b in $\mathcal{D}^b(\mathcal{G}'_0)$ is empty.

Any play following the above strategy and visiting infinitely often a state of $\mathcal{D}^b(\mathcal{G}_0 \cap A')$ is winning for Audrey. A play following the above strategy and remaining eventually in $\mathcal{D}^b(\mathcal{G}'_0)$ is also winning for Audrey.

Let $\mathcal{H}_0$ be the $(d-2)$-attractor decomposition of S_0 obtained by induction. In particular, $\mathcal{H}_0$ is offensively optimal.

Let A_0 be the Steven attractor to S_0 in $\mathcal{G}_0$ and let $\mathcal{G}_1 = \mathcal{G}_0 \setminus A_0$. Subgame $\mathcal{G}_1$ is a trap for Steven and therefore it is a Steven dominion. Let $\mathcal{H}' = \langle \emptyset, (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell) \rangle$ be an offensively optimal Steven d-attractor decomposition of $\mathcal{G}_1$ obtained by induction.

We claim that $\mathcal{H} = \langle A, (S_0, \mathcal{H}_0, A_0), (S_1, \mathcal{H}_1, A_1), \ldots, (S_\ell, \mathcal{H}_\ell, A_\ell) \rangle$ is an offensively optimal Steven d-attractor decomposition of $\mathcal{G}$. Since $\mathcal{H}'$ is offensively optimal, it is enough to show that:

- Audrey has a dominion strategy on $\mathcal{R}^{\mathrm{Str}(\mathcal{H}_0)-1}(\mathcal{G}'_0)$,
- Audrey has a dominion strategy on $\mathcal{D}^{\mathrm{Str}(\mathcal{H}_0)}(\mathcal{G}'_0 \setminus S_0)$.

Since $\mathcal{H}_0$ is offensively optimal, Audrey has a dominion strategy in $\mathcal{R}^{\mathrm{Str}(\mathcal{H}_0)-1}(S_0)$, by Lemma 11, and hence $m \geq \mathrm{Str}(\mathcal{H}_0)$. Moreover, by construction of S_0, Audrey has a dominion strategy on $\mathcal{D}^m(\mathcal{G}'_0 \setminus S_0)$. This implies that Audrey has a dominion strategy on $\mathcal{D}^{\mathrm{Str}(\mathcal{H}_0)}(\mathcal{G}'_0 \setminus S_0)$.

By choice of m, Steven does not have a defensive dominion strategy on $\mathcal{D}^{\mathrm{Str}(\mathcal{H}_0)-1}(\mathcal{G}'_0)$ from any state. This means that for all states s, Audrey has a winning strategy τ_s on $\mathcal{D}^{\mathrm{Str}(\mathcal{H}_0)-1}(\mathcal{G}'_0)$ starting in s. We construct a dominion strategy for her on $\mathcal{R}^{\mathrm{Str}(\mathcal{H}_0)-1}(\mathcal{G}'_0)$: after every visit to a state of rank $2\mathrm{Str}(\mathcal{H}_0) - 1$, Audrey follows τ_s, where s is the first state that follows on the path and whose rank is smaller than $2\mathrm{Str}(\mathcal{H}_0) - 1$. This defines a dominion strategy on $\mathcal{R}^{\mathrm{Str}(\mathcal{H}_0)-1}(\mathcal{G}'_0)$. ◀

5 Strahler-Universal Trees

Our attention now shifts to tackling Question 6. The approach is to develop constructions of small ordered trees into which trees of attractor decompositions or of progress measures can be embedded. Such trees can be seen as natural search spaces for dominion strategies, and existing meta-algorithms such as the universal attractor decomposition algorithm [21] and progress measure lifting algorithm [19, 20] can use them to guide their search, performed in time proportional to the size of the trees in the worst case.

An ordered tree is *universal* for a class of trees if all trees from the class can be embedded into it. The innovation offered in this work is to develop optimized constructions of trees that are universal for classes of trees whose complex structural parameter, such as the Strahler number, is bounded. This is in contrast to less restrictive universal trees introduced by Czerwiński et al. [4] and implicitly constructed by Jurdziński and Lazić [20], whose sizes therefore grow faster with size parameters, leading to slower algorithms.

Firstly, we give an inductive construction of Strahler-universal trees and an upper bound on their numbers of leaves. Then we introduce labelled ordered trees, provide a succinct bit-string labelling of the Strahler-universal trees, and give an alternative and more explicit characterization of the succinctly-labelled Strahler-universal trees. Finally, we argue how the succinct bit-string labelling of Strahler-universal trees facilitates efficient computation of the so-called "level-p successors" in them, which is the key computational primitive that allows using ordered trees to solve parity games. The constructions and techniques we develop here are inspired by and significantly refine those introduced by Jurdziński and Lazić [20].

Strahler-Universal Trees and Their Sizes Intuitively, an ordered tree *can be embedded in* another if the former can be obtained from the latter by pruning some subtrees. More formally, the trivial tree $\langle\rangle$ can be embedded in every ordered tree, and $\langle T_1, T_2, \ldots, T_k\rangle$ can be embedded in $\langle T'_1, T'_2, \ldots, T'_\ell\rangle$ if there are indices $i_1, i_2, \ldots, i_k$ such that $1 \le i_1 < i_2 < \cdots < i_k \le \ell$ and for every $j = 1, 2, \ldots, k$, we have that T_j can be embedded in T'_{i_j}.

An ordered tree is (n,h)*-universal* [4] if every (n,h)-small ordered tree can be embedded in it. We define an ordered tree to be k*-Strahler* (n,h)*-universal* if every (n,h)-small ordered tree whose Strahler number is at most k can be embedded in it, and we give a construction of small Strahler-universal trees.

▶ **Definition 13** (Trees $U^k_{t,h}$ and $V^k_{t,h}$)**.** *For all $t \ge 0$, we define trees $U^k_{t,h}$ (for all h and k such that $h \ge k \ge 1$) and $V^k_{t,h}$ (for all h and k such that $h \ge k \ge 2$) by mutual induction:*

1. *if $h = k = 1$ then $U^k_{t,h} = \langle\rangle$;*
2. *if $h > 1$ and $k = 1$ then $U^k_{t,h} = \left\langle U^k_{t,h-1}\right\rangle$;*
3. *if $h \ge k \ge 2$ and $t = 0$ then $U^k_{t,h} = V^k_{t,h} = \left\langle U^{k-1}_{t,h-1}\right\rangle$;*
4. *if $h \ge k \ge 2$ and $t \ge 1$ then $V^k_{t,h} = V^k_{t-1,h} \cdot \left\langle U^{k-1}_{t,h-1}\right\rangle \cdot V^k_{t-1,h}$;*
5. *if $h = k \ge 2$ and $n \ge 2$ then $U^k_{t,h} = V^k_{t,h}$;*
6. *if $h > k \ge 2$ and $n \ge 2$ then $U^k_{t,h} = V^k_{t,h} \cdot \left\langle U^k_{t,h-1}\right\rangle \cdot V^k_{t,h}$.*

For $g \ge 0$, let I_g be the trivial tree, that is the tree with exactly one leaf, of height g. For example, $I_1 = \langle\rangle$ and $I_3 = \langle\langle\langle\rangle\rangle\rangle = \langle\langle\circ\rangle\rangle$. It is routine to verify that if $h \ge k = 1$ or $t = 0$ then $U^k_{t,h} = I_h$, and if $h \ge k \ge 2$ and $t = 0$ then $V^k_{t,h} = I_h$.

▶ **Lemma 14.** *For all $n \ge 1$ and $h \ge k \ge 1$, the ordered tree $U^k_{\lfloor \lg n\rfloor,h}$ is k-Strahler (n,h)-universal.*

Proof. We say that a tree has *weak Strahler number* at most k if every subtree rooted in a child of the root has Strahler number at most $k-1$. A tree is then *weakly k-Strahler (n,h)-universal* if every (n,h)-small ordered tree whose weak Strahler number is at most k can be embedded in it. We proceed by induction on the number of leaves in an ordered tree and its height, using the following strengthened inductive hypothesis:

- for all $n \ge 1$ and $h \ge k \ge 1$, ordered tree $U^k_{\lfloor \lg n\rfloor,h}$ is k-Strahler (n,h)-universal;
- for all $n \ge 1$ and $h \ge k \ge 2$, ordered tree $V^k_{\lfloor \lg n\rfloor,h}$ is weakly k-Strahler (n,h)-universal.

Let T be an (n, h)-small ordered tree of Strahler number at most k. If $n = 1$, $h = 1$, or $k = 1$, then T is the trivial tree (with just one leaf) of height at most h, and hence it can be embedded in $U^k_{\lfloor \lg n \rfloor, h} = I_h$, the trivial tree of height h. Likewise, if $h \geq k \geq 2$ and $n = 1$, then T is the trivial tree of height at most h, and hence it can be embedded in $V^k_{\lfloor \lg n \rfloor, h} = I_h$, the trivial tree of height h.

Otherwise, we have that $T = \langle T_1, \ldots, T_j \rangle$ for some $j \geq 1$. We consider two cases: either $\mathrm{Str}(T_i) \leq k - 1$ for all $i = 1, \ldots, j$, or there is q such that $\mathrm{Str}(T_q) = k$. Note that by Proposition 3, the latter case can only occur if $h > k$.

If $\mathrm{Str}(T_i) \leq k - 1$ for all $i = 1, \ldots, j$, then we argue that T can be embedded in $V^k_{\lfloor \lg n \rfloor, h}$, and hence also in $U^k_{\lfloor \lg n \rfloor, h}$, because $V^k_{\lfloor \lg n \rfloor, h}$ can be embedded in $U^k_{\lfloor \lg n \rfloor, h}$ by definition (see items 3., 5., and 6. of Definition 13). Let p (a pivot) be an integer such that both trees $T' = \langle T_1, \ldots, T_{p-1} \rangle$ and $T'' = \langle T_{p+1}, \ldots, T_j \rangle$ are $(\lfloor n/2 \rfloor, h)$-small. Then by the strengthened inductive hypothesis, each of the two trees T' and T'' can be embedded in tree $V^k_{\lfloor \lg \lfloor n/2 \rfloor \rfloor, h} = V^k_{\lfloor \lg n \rfloor - 1, h}$ and tree T_p can be embedded in $U^{k-1}_{\lfloor \lg n \rfloor, h-1}$. It then follows that tree $T = T' \cdot \langle T_p \rangle \cdot T''$ can be embedded in $V^k_{\lfloor \lg n \rfloor, h} = V^k_{\lfloor \lg n \rfloor - 1, h} \cdot \left\langle U^{k-1}_{\lfloor \lg n \rfloor, h-1} \right\rangle \cdot V^k_{\lfloor \lg n \rfloor - 1, h}$.

If $\mathrm{Str}(T_q) = k$ for some q (the pivot), then we argue that T can be embedded in $U^k_{\lfloor \lg n \rfloor, h}$. Note that each of the two trees $T' = \langle T_1, \ldots, T_{q-1} \rangle$ and $T'' = \langle T_{q+1}, \ldots, T_j \rangle$ is (n, h)-small and all trees $T_1, \ldots, T_{q-1}$ and $T_{q+1}, \ldots, T_j$ have Strahler numbers at most $k - 1$. By the previous paragraph, it follows that each of the two trees T' and T'' can be embedded in $V^k_{\lfloor \lg n \rfloor, h}$. Moreover, tree T_q is $(n, h - 1)$-small and hence, by the inductive hypothesis, it can be embedded in $U^k_{\lfloor \lg n \rfloor, h-1}$. It follows that tree $T = T' \cdot \langle T_q \rangle \cdot T''$ can be embedded in $U^k_{\lfloor \lg n \rfloor, h} = V^k_{\lfloor \lg n \rfloor, h} \cdot \left\langle U^k_{\lfloor \lg n \rfloor, h-1} \right\rangle \cdot V^k_{\lfloor \lg n \rfloor, h}$. ◀

▶ **Lemma 15.** *For all $t \geq 0$, we have:*

- *if $h \geq k = 1$ then* $\mathrm{leaves}\left(U^k_{t,h}\right) = 1$;
- *if $h \geq k \geq 2$ then* $\mathrm{leaves}\left(U^k_{t,h}\right) \leq 2^{t+k} \binom{t+k-2}{k-2} \binom{h-1}{k-1}$.

▶ **Theorem 16.** *For $k \leq \lg n$, the number of leaves of the k-Strahler (n, h)-universal ordered trees $U^k_{\lfloor \lg n \rfloor, h}$ is $n^{O(1)} \cdot (h/k)^k = n^{k \lg(h/k)/\lg n + O(1)}$, which is polynomial in n if $k \cdot \lg(h/k) = O(\log n)$. In more detail, the number is at most $n^{c(n)} \cdot (h/k)^k$, where $c(n) = 5.45$ if $k \leq \lg n$, $c(n) = 3 + o(1)$ if $k = o(\log n)$, and $c(n) = 1 + o(1)$ if $k = O(1)$.*

▶ Remark 17. By Proposition 3 and Lemma 14, for all positive integers n and h, the tree $U^{\lfloor \lg n \rfloor + 1}_{\lfloor \lg n \rfloor, h}$ is (n, h)-universal. Theorem 16 implies that the number of leaves of $U^{\lfloor \lg n \rfloor + 1}_{\lfloor \lg n \rfloor, h}$ is $n^{\lg(h/\lg n) + O(1)}$, which matches the asymptotic number of leaves of (n, h)-universal trees of Jurdziński and Lazić [20, Lemma 6]. In particular, if $h = O(\log n)$ then $\lg(h/\lg n) = O(1)$, and hence the number of leaves of $U^{\lfloor \lg n \rfloor + 1}_{\lfloor \lg n \rfloor, h}$ is polynomial in n.

Labelled Strahler-Universal Trees *Labelled ordered tree* are similar to ordered trees: the trivial tree $\langle \rangle$ is an *A-labelled ordered tree* and so is a sequence $\langle (a_1, \mathcal{L}_1), (a_2, \mathcal{L}_2), \ldots, (a_k, \mathcal{L}_k) \rangle$, where $\mathcal{L}_1$, $\mathcal{L}_2$, …, $\mathcal{L}_k$ are A-labelled ordered trees, and a_1, a_2, …, a_k are distinct elements of a linearly ordered set $(A, \leq)$ and $a_1 < a_2 < \cdots < a_k$ in that linear order. We define the *unlabelling* of a labelled ordered tree $\langle (a_1, \mathcal{L}_1), (a_2, \mathcal{L}_2), \ldots, (a_k, \mathcal{L}_k) \rangle$, by straightforward induction, to be the ordered tree $\langle T_1, T_2, \ldots, T_k \rangle$, where T_i is the unlabelling of $\mathcal{L}_i$ for every $i = 1, 2, \ldots, k$. An *A-labelling* of an ordered tree T is an A-labelled tree $\mathcal{L}$ whose unlabelling is T. We define the *natural labelling* of an ordered tree $T = \langle T_1, \ldots, T_k \rangle$, again by a straightfoward induction, to be the $\mathbb{N}$-labelled tree $\langle (1, \mathcal{L}_1), \ldots, (k, \mathcal{L}_k) \rangle$, where $\mathcal{L}_1$, …, $\mathcal{L}_k$ are the natural labellings of trees T_1, …, T_k.

For an A-labelled tree $\langle (a_1, \mathcal{L}_1), \ldots, (a_k, \mathcal{L}_k) \rangle$, its set of *nodes* is defined inductively to consist of the root $\langle\rangle$ and all the sequences in A^* of the form $\langle a_i \rangle \cdot v$, where $v \in A^*$ is a node in $\mathcal{L}_i$ for some $i = 1, \ldots, k$, and where the symbol $\cdot$ denotes concatenation of sequences. For example, the natural labelling of tree $\left\langle \langle \circ^3 \rangle, \circ^4, \langle \langle \circ \rangle \rangle^2 \right\rangle$ has the set of nodes that consists of the following set of leaves $\langle 1, 1 \rangle$, $\langle 1, 2 \rangle$, $\langle 1, 3 \rangle$, $\langle 2 \rangle$, $\langle 3 \rangle$, $\langle 4 \rangle$, $\langle 5 \rangle$, $\langle 6, 1, 1 \rangle$, $\langle 7, 1, 1 \rangle$, and all of their prefixes. Indeed, the set of nodes of a labelled ordered tree is always prefix-closed. Moreover, if $L \subseteq A^*$ then its closure under prefixes uniquely identifies a labelled ordered tree that we call the labelled ordered tree *generated* by L, and its unlabelling is the ordered tree generated by L. For example, the set $\{ \langle 1 \rangle, \langle 3, 1 \rangle, \langle 3, 4, 1 \rangle, \langle 6, 1 \rangle \}$ generates ordered tree $\langle \circ, \langle \circ, \langle \circ \rangle \rangle, \langle \circ \rangle \rangle$.

Consider the following linear order on the set $\{ 0, 1 \}^*$ of bit strings: for each bit $b \in \{ 0, 1 \}$, and for all bit strings $\beta, \beta' \in \{ 0, 1 \}^*$, if ε is the empty string, then we have $0\beta < \varepsilon$, $\varepsilon < 1\beta$, and $b\beta < b\beta'$ iff $\beta < \beta'$.

For a bit string $\beta \in \{ 0, 1 \}^*$, we write $|\beta|$ for the number of bits used in the string. For example, we have $|\varepsilon| = 0$ and $|010| = 3$, and $|11| = 2$. Suppose that $\langle \beta_i, \beta_{i-1}, \ldots, \beta_1 \rangle$ is a node in a $\{ 0, 1 \}^*$-labelled ordered tree. Then if $\beta_j = b\beta$ for some $j = 1, 2, \ldots, i$, $b \in \{ 0, 1 \}$, and $\beta \in \{ 0, 1 \}^*$, then we refer to the first bit b as the *leading bit* in β_j, and we refer to all the following bits in β as *non-leading bits* in β_j. For example, node $\langle \varepsilon, 010, \varepsilon, \varepsilon, 11 \rangle$ has two non-empty strings and hence two leading bits, and it uses three non-leading bits overall, because $|010| + |11| - 2 = 3$.

For a bit $b \in \{ 0, 1 \}$ and a $\{ 0, 1 \}^*$-labelled ordered tree $\mathcal{L} = \langle (\beta_1, \mathcal{L}_1), \ldots, (\beta_\ell, \mathcal{L}_\ell) \rangle$, we define the $\{ 0, 1 \}^*$-labelled ordered tree $[\mathcal{L}]^b$ to be equal to $\mathcal{L} = \langle (b\beta_1, \mathcal{L}_1), \ldots, (b\beta_\ell, \mathcal{L}_\ell) \rangle$. In other words, $[\mathcal{L}]^b$ is the labelled ordered tree that is obtained from $\mathcal{L}$ by adding an extra copy of bit b as the leading bit in the labels of all children of the root of $\mathcal{L}$.

The inductive structure of the next definition is identical to that of Definition 13, and hence labelled ordered trees $\mathcal{U}^k_{t,h}$ and $\mathcal{V}^k_{t,h}$ defined here are labellings of the ordered trees $U^k_{t,h}$ and $V^k_{t,h}$, respectively.

▶ **Definition 18** (Trees $\mathcal{U}^k_{t,h}$ and $\mathcal{V}^k_{t,h}$)**.** *For all $t \geq 0$, we define $\{ 0, 1 \}^*$-labelled ordered trees $\mathcal{U}^k_{t,h}$ (for all h and k such that $h \geq k \geq 1$) and $\mathcal{V}^k_{t,h}$ (for all h and k such that $h \geq k \geq 2$) by mutual induction:*

1. *if $h = k = 1$ then $\mathcal{U}^k_{t,h} = \langle\rangle$;*
2. *if $h > 1$ and $k = 1$ then $\mathcal{U}^k_{t,h} = \left\langle \left(\varepsilon, \mathcal{U}^k_{t,h-1} \right) \right\rangle$;*
3. *if $h \geq k \geq 2$ and $t = 0$ then $\mathcal{V}^k_{t,h} = \left\langle \left(\varepsilon, \mathcal{U}^{k-1}_{t,h-1} \right) \right\rangle$ and $\mathcal{U}^k_{t,h} = \left[\mathcal{V}^k_{t,h} \right]^0 = \left\langle \left(0, \mathcal{U}^{k-1}_{t,h-1} \right) \right\rangle$;*
4. *if $h \geq k \geq 2$ and $t \geq 1$ then $\mathcal{V}^k_{t,h} = \left[\mathcal{V}^k_{t-1,h} \right]^0 \cdot \left\langle \left(\varepsilon, \mathcal{U}^{k-1}_{t,h-1} \right) \right\rangle \cdot \left[\mathcal{V}^k_{t-1,h} \right]^1$;*
5. *if $h = k \geq 2$ and $t \geq 1$ then $\mathcal{U}^k_{t,h} = \left[\mathcal{V}^k_{t,h} \right]^0$;*
6. *if $h > k \geq 2$ and $t \geq 1$ then $\mathcal{U}^k_{t,h} = \left[\mathcal{V}^k_{t,h} \right]^0 \cdot \left\langle \left(\varepsilon, \mathcal{U}^k_{t,h-1} \right) \right\rangle \cdot \left[\mathcal{V}^k_{t,h} \right]^1$.*

The inductive definition of labelled ordered trees $\mathcal{U}^k_{t,h}$ and $\mathcal{V}^k_{t,h}$ makes it straightforward to argue that their unlabellings are equal to trees $U^k_{t,h}$ and $V^k_{t,h}$, respectively, and hence to transfer to them Strahler-universality established in Lemma 14 and upper bounds on the numbers of leaves established in Lemma 15 and Theorem 16. We now give an alternative and more explicit characterization of those trees, which will be more suitable for algorithmic purposes. To that end, we define $\{ 0, 1 \}^*$-labelled trees $\mathcal{B}^k_{t,h}$ and $\mathcal{C}^k_{t,h}$ and then we argue that they are equal to trees $\mathcal{U}^k_{t,h}$ and $\mathcal{V}^k_{t,h}$, respectively, by showing that they satisfy all the recurrences in Definition 18.

▶ **Definition 19** (Trees $\mathcal{B}^k_{t,h}$ and $\mathcal{C}^k_{t,h}$). *For all $t \geq 0$ and $h \geq k \geq 1$, we define $\{0,1\}^*$-labelled ordered trees $\mathcal{B}^k_{t,h}$ as the tree generated by sequences $\langle \beta_{h-1}, \ldots, \beta_1 \rangle$ such that:*

1. *the number of non-empty bit strings among β_{h-1}, ..., β_1 is $k-1$;*
2. *the number of bits used in bit strings β_{h-1}, ..., β_1 overall is at most $(k-1)+t$;*

and for every $i = 1, \ldots, h-1$, we have the following:

3. *if there are less than $k-1$ non-empty bit strings among β_{h-1}, ..., β_{i+1}, but there are t non-leading bits used in them, then $\beta_i = 0$;*
4. *if all bit strings β_i, ..., β_1 are non-empty, then each of them has 0 as its leading bit.*

For all $t \geq 0$ and $h \geq k \geq 2$, we define $\{0,1\}^*$-labelled ordered trees $\mathcal{C}^k_{t,h}$ as the tree generated by sequences $\langle \beta_{h-1}, \ldots, \beta_1 \rangle$ such that:

1. *the number of non-empty bit strings among β_{h-2}, ..., β_1 is $k-2$;*
2. *the number of bits used in bit strings β_{h-1}, ..., β_1 overall is at most $(k-2)+t$;*

and for every $i = 1, \ldots, h-1$, we have the following:

3. *if there are less than $k-2$ non-empty bit strings among β_{h-2}, ..., β_{i+1}, but there are $t - |\beta_{h-1}|$ non-leading bits used in them, then $\beta_i = 0$;*
4. *if all bit strings β_i, ..., β_1 are non-empty, then each of them has 0 as its leading bit.*

▶ **Lemma 20.** *For all $t \geq 0$ and $h \geq k \geq 1$, we have $\mathcal{U}^k_{t,h} = \mathcal{B}^k_{t,h}$.*

The following corollary follows from Lemma 20, and from the identical inductive structures of Definitions 13 and 18.

▶ **Corollary 21.** *For all $t \geq 0$ and $h \geq k \geq 1$, the unlabelling of $\mathcal{B}^k_{t,h}$ is equal to $U^k_{t,h}$.*

Efficiently Navigating Labelled Strahler-Universal Trees. The computation of the *level-p successor* of a leaf in a labelled ordered tree of height h is the following problem: given a leaf $\langle \beta_h, \beta_{h-1}, \ldots, \beta_1 \rangle$ in the tree and given a number p, such that $1 \leq p \leq h$, compute the $<_{\text{lex}}$-smallest leaf $\langle \beta'_h, \beta'_{h-1}, \ldots, \beta'_1 \rangle$ in the tree, such that $\langle \beta_h, \ldots, \beta_p \rangle <_{\text{lex}} \langle \beta'_h, \ldots, \beta'_p \rangle$. As (implicitly) explained by Jurdziński and Lazić [20, Proof of Theorem 7], the level-p successor computation is the key primitive used extensively in an implementation of a progress measure lifting algorithm.

▶ **Lemma 22.** *Every leaf in tree $\mathcal{B}^k_{t,h}$ can be represented using $O((k+t)\log h)$ bits and for every $p = 1, 2, \ldots, h$, the level-p successor of a leaf in tree $\mathcal{B}^k_{t,h}$ can be computed in time $O((k+t)\log h)$.*

6 Progress-Measure Strahler Numbers

Consider a parity game $\mathcal{G}$ in which all vertex priorities are at most an even number d. If $(A, \leq)$ is a well-founded linear order then we write sequences in $A^{d/2}$ in the following form $\langle m_{d-1}, m_{d-3}, \ldots, m_1 \rangle$, and for every priority $p \in \{0, 1, \ldots, d\}$, we define the *$p$-truncation* of $\langle m_{d-1}, m_{d-3}, \ldots, m_1 \rangle$, denoted by $\langle m_{d-1}, m_{d-3}, \ldots, m_1 \rangle|_p$, to be the sequence $\langle m_{d-1}, \ldots, m_{p+2}, m_p \rangle$ if p is odd and $\langle m_{d-1}, \ldots, m_{p+3}, m_{p+1} \rangle$ if p is even. We use the lexicographic order $\leq_{\text{lex}}$ to linearly order the set $A^* = \bigcup_{i=0}^{\infty} A^i$.

A *Steven progress measure* [8, 19, 20] on a parity game $\mathcal{G}$ is a map $\mu : V \to A^{d/2}$ such that for every vertex $v \in V$:

- if $v \in V_{\text{Even}}$ then there is a μ-progressive edge $(v, u) \in E$;
- if $v \in V_{\text{Odd}}$ then every edge $(v, u) \in E$ is μ-progressive;

where we say that an edge $(v, u) \in E$ is *μ-progressive* if:

- if $\pi(v)$ is even then $\mu(v)|_{\pi(v)} \geq_{\text{lex}} \mu(u)|_{\pi(v)}$;
- if $\pi(v)$ is odd then $\mu(v)|_{\pi(v)} >_{\text{lex}} \mu(u)|_{\pi(v)}$.

We define *the tree of a progress measure* μ to be the ordered tree generated by the image of V under μ.

▶ **Theorem 23** ([8, 19, 20])**.** *There is a Steven progress measure on a parity game $\mathcal{G}$ if and only if every vertex in $\mathcal{G}$ is in its largest Steven dominion. If game $\mathcal{G}$ is (n, d)-small then the tree of a progress measure on $\mathcal{G}$ is $(n, d/2 + 1)$-small.*

We define the *Steven progress-measure Strahler number* of a parity game $\mathcal{G}$ to be the smallest Strahler number of a tree of a progress measure on $\mathcal{G}$. The following theorem refines and strengthens Theorems 2 and 23 by establishing that the Steven Strahler number and the Steven progress-measure Strahler number of a parity game nearly coincide.

▶ **Theorem 24.** *The Steven Strahler number and the Steven progress-measure Strahler number of a parity game differ by at most* 1*.*

The translations between progress measures and attractor decompositions are as given by Daviaud, Jurdziński, and Lazić [5]; here we point out that they do not increase the Strahler number of the underlying trees by more than 1. This coincidence of the two complexity measures, one based on attractor decompositions and the other based on progress measures, allows us in Section 7 to use a progress measure lifting algorithm to solve games with bounded Strahler number.

7 Strahler-Universal Progress Measure Lifting Algorithm

Jurdziński and Lazić [20, Section IV] have implicitly suggested that the progress-measure lifting algorithm [19] can be run on any ordered tree and they have established the correctness of such an algorithm if their *succinct multi-counters trees* were used. This has been further clarified by Czerwiński et al. [4, Section 2.3], who have explicitly argued that any $(n, d/2)$-universal ordered tree is sufficient to solve an (n, d)-small parity game in this way. We make explicit a more detailed observation that follows using the same standard arguments (see, for example, Jurdziński and Lazić [20, Theorem 5]).

▶ **Proposition 25.** *Suppose the progress measure-lifting algorithm is run on a parity game $\mathcal{G}$ and on an ordered tree T. Let D be the largest Steven dominion in $\mathcal{G}$ on which there is a Steven progress measure whose tree can be embedded in T. Then the algorithm returns a Steven dominion strategy on D.*

An elementary corollary of this observation is that if the progress-measure lifting algorithm is run on the tree of a progress measure on some Steven dominion in a parity game, then the algorithm produces a Steven dominion strategy on a superset of that dominion. Note that this is achieved in polynomial time because the tree of a progress measure on an (n, d)-small parity game is $(n, d/2)$-small and the running time of the algorithm is dominated by the size of the tree [20, Section IV.B].

▶ **Theorem 26.** *There is an algorithm for solving (n, d)-small parity games of Strahler number k in quasi-linear space and time $n^{O(1)} \cdot (d/2k)^k = n^{k \lg(d/k)/\lg n + O(1)}$, which is polynomial in n if $k \cdot \lg(d/k) = O(\log n)$.*

Proof. By Proposition 3, we may assume that $k \leq 1 + \lg n$. In order to solve an (n, d)-small parity game of Steven Strahler number k, run the progress-measure lifting algorithm for

Steven on tree $\mathcal{B}^{k+1}_{\lfloor \lg n \rfloor, d/2+1}$, which is $(k+1)$-Strahler $(n, d/2+1)$-universal by Lemma 14 and Corollary 21. By Theorem 24 and by Proposition 25, the algorithm will then return a Steven dominion strategy on the largest Steven dominion. The running time and space upper bounds follow from Theorem 16, by the standard analysis of progress-measure lifting as in [20, Theorem 7], and by Lemma 22. ◀

▶ Remark 27. We highlight the $k \cdot \lg(d/k) = O(\log n)$ criterion from Theorem 26 as offering a novel trade-off between two natural structural complexity parameters of parity games (number of of priorities d and the Strahler/Lehtinen number k) that enables solving them in time that is polynomial in the number of vertices n. It includes as special cases both the $d < \lg n$ criterion of Calude et al. [3, Theorem 2.8] and the $d = O(\log n)$ criterion of Jurdziński and Lazić [20, Theorem 7] (set $k = \lfloor \lg n \rfloor + 1$ and use Propositions 4 and 3 to justify it), and the $k = O(1)$ criterion of Lehtinen [24, Theorem 3.6] (by Theorem 8).

We argue that the new $k \cdot \lg(d/k) = O(\log n)$ criterion (Theorem 26) enabled by our results (coincidence of the Strahler and the Lehtinen numbers: Theorem 8) and techniques (small and efficiently navigable Strahler-universal trees: Theorem 16, Corollary 21, and Lemma 22) considerably expands the asymptotic ranges of the natural structural complexity parameters in which parity games can be solved in polynomial time. We illustrate it by considering the scenario in which the rates of growth of both k and $\lg d$ as functions of n are $O\left(\sqrt{\log n}\right)$, i.e., d is $2^{O\left(\sqrt{\log n}\right)}$. Note that the number of priorities d in this scenario is allowed to grow as fast as $2^{b \cdot \sqrt{\lg n}}$ for an arbitrary positive constant b, which is significantly larger than what is allowed by the $d = O(\log n)$ criterion of Jurdziński and Lazić [20, Theorem 7]. Indeed, its rate of growth is much larger than any poly-logarithmic function of n, because for every positive constant c, we have $(\lg n)^c = 2^{c \cdot \lg \lg n}$, and $c \cdot \lg \lg n$ is exponentially smaller than $b \cdot \sqrt{\lg n}$. At the same time, the $O\left(\sqrt{\log n}\right)$ rate of growth allowed in this scenario for the Strahler number k substantially exceeds $k = O(1)$ required by Lehtinen [24, Theorem 3.6].

References

1 P. Baldan, B. König, C. Mika-Michalski, and T. Padoan. Fixpoint games on continuous lattices. *Proceedings of the ACM on Programming Languages*, 3(POPL, January 2019):26:1–26:29, 2019.

2 J. C. Bradfield and I. Walukiewicz. *Handbook of Model Checking*, chapter The mu-calculus and model checking, pages 871–919. Springer, 2018.

3 C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. Deciding parity games in quasipolynomial time. In *STOC 2017*, pages 252–263, Montreal, QC, Canada, 2017. ACM.

4 W. Czerwiński, L. Daviaud, N. Fijalkow, M. Jurdziński, R. Lazić, and P. Parys. Universal trees grow inside separating automata: Quasi-polynomial lower bounds for parity games. In *Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 2333–2349, San Diego, CA, 2019. SIAM.

5 L. Daviaud, M. Jurdziński, and R. Lazić. A pseudo-quasi-polynomial algorithm for mean-payoff parity games. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, pages 325–334, Oxford, UK, 2018. ACM.

6 L. Daviaud, M. Jurdziński, and K. Lehtinen. Alternating weak automata from universal trees. In *30th International Conference on Concurrency Theory, CONCUR 2019*, volume 140 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:14, Amsterdam, the Netherlands, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

7 L. Daviaud, M. Jurdziński, and K. S. Thejaswini. The Strahler number of a parity game. arXiv:2003.08627, 2020.

8 E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *32nd Annual Symposium on Foundations of Computer Science*, pages 368–377, San Juan, Puerto Rico, 1991. IEEE Computer Society.

9 E. A. Emerson, C. S. Jutla, and P. Sistla. On model-checking for fragments of μ-calculus. In *CAV 1993*, volume 697 of *LNCS*, pages 385–396, Elounda, Greece, 1993. Springer.

10 A. P. Ershov. On programming of arithmetic operations. *Communications of the ACM*, 1(8):3–6, 1958.

11 J. Esparza, M. Luttenberger, and M. Schlund. A brief history of Strahler numbers—with a preface. Technical report, Technical University of Munich, 2016.

12 J. Fearnley. Exponential lower bounds for policy iteration. In *ICALP 2010*, volume 6199 of *LNCS*, pages 551–562, Bordeaux, France, 2010. Springer.

13 J. Fearnley, S. Jain, B. de Keijzer, S. Schewe, F. Stephan, and D. Wojtczak. An ordered approach to solving parity games in quasi-polynomial time and quasi-linear space. *International Journal on Software Tools for Technology Transfer*, 21(3):325–349, 2019.

14 O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *LICS 2009*, pages 145–156, Los Angeles, CA, USA, 2009. IEEE Computer Society.

15 O. Friedmann. A subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games. In *IPCO 2011*, volume 6655 of *LNCS*, pages 192–206, New York, NY, USA, 2011. Springer.

16 O. Friedmann, T. D. Hansen, and U. Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In *STOC 2011*, pages 283–292, San Jose, CA, USA, 2011. ACM.

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

18 D. Hausmann and L. Schröder. Computing nested fixpoints in quasipolynomial time. arXiv:1907.07020, 2019.

19 M. Jurdziński. Small progress measures for solving parity games. In *17th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *LNCS*, pages 290–301, Lille, France, 2000. Springer.

20 M. Jurdziński and R. Lazić. Succinct progress measures for solving parity games. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017*, pages 1–9, Reykjavik, Iceland, 2017. IEEE Computer Society.

21 M. Jurdziński and R. Morvan. A universal attractor decomposition algorithm for parity games. arXiv:2001.04333, 2020.

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

23 D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1973.

24 K. Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, pages 639–648, Oxford, UK, 2018. IEEE.

25 K. Lehtinen and U. Boker. Register games. arXiv:1902.10654, April 2020.

26 K. Lehtinen, S. Schewe, and D. Wojtczak. Improving the complexity of Parys' recursive algorithm, 2019. `arXiv:1904.11810`.

27 R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65(2):149–184, 1993.

28 P. Parys. Parity games: Zielonka's algorithm in quasi-polynomial time. In *MFCS 2019*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:13, Aachen, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

29 P. Parys. Parity games: Another view on Lehtinen's algorithm. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, volume 152 of *LIPIcs*, pages 32:1–32:15, Barcelona, Spain, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

30 X. G. Viennot. Trees everywhere. In *15th Colloquium on Trees in Algebra and Programming*, volume 431 of *LNCS*, pages 18–41, Copenhagen, Denmark, 1990. Springer.

31 J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *CAV 2000*, volume 1855 of *LNCS*, pages 202–215, Chicago, IL, USA, 2000. Springer.

32 W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.