

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 $k \cdot \lg(d/k) = O(\lg n)$ 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 Khousseinov, 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 $d = 2^{O(\sqrt{\lg n})}$ and $k = O(\sqrt{\lg n})$.

2012 ACM Subject Classification

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

Digital Object Identifier 10.4230/LIPIcs.CVIT.2020.7

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 Alexander Kozachinskiy and Matthew Richmond, the latter an undergraduate student at Warwick, for spotting gaps in some definitions in earlier drafts of this paper. We also 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 in March 2015.



© L. Daviaud, M. Jurdziński, and K. S. Thejaswini;
licensed under Creative Commons License CC-BY

42nd Conference on Very Important Topics (CVIT 2020).

Editors: John Q. Open and Joan R. Access; Article No. 7; pp. 7:1–7:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Context

Parity games. Parity games are a fundamental model in automata theory and logic [7, 30, 16, 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 [7, 30], model checking and satisfiability of fixpoint logics [8, 2], and evaluation of nexted fixpoint expressions [1, 17]. It is a long-standing open problem whether parity games can be solved in polynomial time [8].

The impact of parity games goes well beyond their home turf of automata theory, logic, and formal methods. For example, an answer [13] of a question posed originally for parity games [29] has strongly inspired major breakthroughs on the computational complexity of fundamental algorithms in stochastic planning [11] and linear optimization [14, 15].

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 [9] 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 [22], Viennot [28], and Esparza, Luttenberger, and Schlund [10].

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 [19], and Lehtinen [23]. 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 [25] and Zielonka [30]. Parys [26] has proposed an ingenious quasi-polynomial version of McNaughton-Zielonka algorithm, but Lehtinen, Schewe, and Wojtczak [24], and Jurdziński and Morvan [20] 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 [23] has relatively least attractive worst-case running time bounds. Parys [27] 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 [19], and Fearnley, Jain, de Keijzer, Schewe, Stephan, and Wojtczak [12], 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 [23].

We give a construction of small Strahler-universal trees that, when used with the progress measure lifting algorithm [18, 19] or with the universal attractor decomposition algorithm [20], 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 [23] and the $d < \lg n$ and $d = O(\log n)$ criteria of Calude et al. [3], and of Jurdziński and Lazić [19], respectively.

Proofs. Proofs of some of our technical results can be found in the Appendix. This extended abstract focuses on motivating and developing the key concepts and constructions, succinctly stating the main results that link them together, and putting our main conceptual and technical contributions in the context of the relevant literature.

2 Dominia, attractor decompositions, and their trees

Strategies, traps, and dominia. A *parity game* [7] \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 $\pi : V \rightarrow \{0, 1, \dots, d\}$ 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* [30] 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* [21] if there is a Steven dominion strategy on it.

Audrey strategies, trapping Steven, and Audrey dominia are defined in an analogous way by swapping the roles of the two players. We note that the sets of Steven dominia and of Audrey dominia are each closed under union, and hence the largest Steven and Audrey dominia exist, and they are the unions of all Steven and Audrey dominia, 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*

7:4 The Strahler number of a parity game

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}* [30]. *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 $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \dots, (S_k, \mathcal{H}_k, A_k) \rangle$ is a *Steven d -attractor decomposition* [5, 6, 20] 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 $i = 1, 2, \dots, k$, 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}_{k+1} is empty. If $d = 0$ then we require that $k = 0$.

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** ([30, 5, 20]). *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** ([7, 25, 30, 20]). *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 $\langle T_1, T_2, \dots, T_k \rangle$, where T_i is an ordered tree for every $i = 1, 2, \dots, k$. The trivial tree has only one node called the root, which is a leaf; and a tree of the form $\langle T_1, T_2, \dots, T_k \rangle$ 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, \dots, T consisting of i copies of tree T . Then the expression $\langle T^i \rangle = \langle T, \dots, T \rangle$ 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, $\langle \langle \circ^3 \rangle \rangle \cdot \langle \circ^4, \langle \langle \circ \rangle \rangle^2 \rangle = \langle \langle \circ^3 \rangle, \circ^4, \langle \langle \circ \rangle \rangle^2 \rangle = \langle \langle \circ, \circ, \circ \rangle, \circ, \circ, \circ, \circ, \langle \langle \circ \rangle \rangle, \langle \langle \circ \rangle \rangle \rangle$.

For an ordered tree T , we write $\text{height}(T)$ for its *height* and $\text{leaves}(T)$ for its *number of leaves*, and we define them by the following routine induction: the trivial tree $\langle \rangle = \circ$ has 1

leaf and its height is 0; the number of leaves of tree $\langle T_1, T_2, \dots, T_k \rangle$ is the sum of the numbers of leaves of trees T_1, T_2, \dots, T_k ; and its height is 1 plus the maximum height of trees T_1, T_2, \dots, T_k . For example, the tree $\langle \langle \circ^3 \rangle, \circ^4, \langle \langle \circ \rangle \rangle^2 \rangle$ has 9 leaves and height 3. 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* $\text{Str}(T)$ 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 0; and if $T = \langle T_1, \dots, T_k \rangle$ and m is the largest Strahler number of trees T_1, \dots, T_k , then $\text{Str}(T) = m$ if there is a unique i such that $\text{Str}(T_i) = m$, and $\text{Str}(T) = m + 1$ otherwise. For example, we have $\text{Str}(\langle \langle \langle \circ^3 \rangle, \circ^4, \langle \langle \circ \rangle \rangle^2 \rangle \rangle) = 1$ because $\text{Str}(\circ) = \text{Str}(\langle \langle \circ \rangle \rangle) = 0$ and $\text{Str}(\langle \circ^3 \rangle) = 1$.

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

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 $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \dots, (S_k, \mathcal{H}_k, A_k) \rangle$ is a Steven d -attractor decomposition then we define the *tree of attractor decomposition* \mathcal{H} [6, 20], denoted by $T_{\mathcal{H}}$, to be the trivial ordered tree $\langle \rangle$ if $k = 0$, and otherwise, to be the ordered tree $\langle T_{\mathcal{H}_1}, T_{\mathcal{H}_2}, \dots, T_{\mathcal{H}_k} \rangle$, where for every $i = 1, 2, \dots, k$, 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, \dots, S_k 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, 20]). *If \mathcal{H} is an attractor decomposition of an (n, d) -small parity game then its tree $T_{\mathcal{H}}$ is $(n, \lceil d/2 \rceil)$ -small.*

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 dominia, respectively.

3 Strahler strategies in register games

This section establishes a connection between the register number of a parity game defined by Lehtinen [23] 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 [23] and Parys [27]?

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 $(v, \langle r_k, r_{k-1}, \dots, r_1 \rangle)$ or triples $(v, \langle r_k, r_{k-1}, \dots, r_1 \rangle, p)$, where v is a vertex

7:6 The Strahler number of a parity game

in \mathcal{G} , $d \geq r_k \geq r_{k-1} \geq \dots \geq r_1 \geq 0$, and $1 \leq p \leq 2k + 1$. The former states have rank 1 and the latter have rank p . Each number r_i , for $i = k, k-1, \dots, 1$, is referred to as the value of the i -th register in the state. Steven owns all states $(v, \langle r_k, r_{k-1}, \dots, r_1 \rangle)$ and the owner of vertex v in \mathcal{G} is the owner of states $(v, \langle r_k, r_{k-1}, \dots, r_1 \rangle, p)$ for every p . How the game is played by Steven and Audrey is determined by the available moves:

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

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

$$r'_j = \max(r_j, \pi(v)) \text{ for } j \geq i + 1, \text{ and } r'_i = \pi(v);$$

- at every state $(v, \langle r_k, r_{k-1}, \dots, r_1 \rangle, p)$ the owner of vertex v in \mathcal{G} picks an edge (v, u) in \mathcal{G} , leading to state $(u, \langle r_k, r_{k-1}, \dots, r_1 \rangle)$ of rank 1 and with unchanged register values.

For example, at state $(v, \langle 9, 6, 4, 4, 3 \rangle)$ of rank 1, such that the priority $\pi(v)$ of vertex v is 5, if 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 [23, 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 for all register values \bar{r} , state (v, \bar{r}) is in the largest Steven dominion in $\mathcal{R}^k(\mathcal{G})$. By defining the *(Steven) register number* [23, Definition 3.5] of a parity game \mathcal{G} to be the smallest such number k , and by proving the $1 + \lg n$ upper bound on the register number of every (n, d) -small parity game [23, 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 [19].

Lehtinen [23, Definition 4.8] has also considered the concept of a Steven *defensive dominion strategy on a set of states* in a k -register game: it is a Steven dominion strategy on the set of states in $\mathcal{R}^k(\mathcal{G})$ in which there is no state of rank $2k + 1$. The *(Steven) defensive register number* [23, Definition 4.9] of a Steven dominion D in \mathcal{G} is then defined as the smallest number k such that Steven has a defensive dominion strategy in $\mathcal{R}^k(\mathcal{G})$, which for every $v \in D$ includes all states $(v, \langle r_k, \dots, r_1 \rangle)$, 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.

► **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 [23, 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} winning for Steven from all the vertices and a Steven attractor decomposition \mathcal{H} of \mathcal{G} of Strahler number k . Let us construct a defensive k -register strategy for Steven. The strategy is defined inductively on the height of $\mathcal{T}_{\mathcal{H}}$. We fix d to be the least even integer no smaller than any of the priority in \mathcal{G} .

Strategy for Steven. If $\mathcal{H} = \langle A, \emptyset \rangle$, then \mathcal{G} consists of vertices of priority d and of their Steven attractor. In this case, Steven plays the reachability strategy in A and resets register 1 immediately after seeing a vertex of priority d . We define a Steven strategy in \mathcal{G} with the following moves:

Suppose now that $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \dots, (S_j, \mathcal{H}_j, A_j) \rangle$ and has Strahler number k . For all i , let k_i to be the Strahler number of $\mathcal{T}_{\mathcal{H}_i}$, for some $k_i \geq 1$. By induction, for all i , we have a Steven defensive k_i -register strategy in $\mathcal{G} \cap S_i$, which induces a Steven defensive k -register strategy in $\mathcal{G} \cap S_i$ denoted by τ_i .

We define the strategy for Steven in \mathcal{G} as follows, where S denote the set of vertices of priority d in \mathcal{G} :

- On $\mathcal{G} \cap S_i$, Steven uses the defensive k -register strategy τ_i ;
- On $\mathcal{G} \cap (A_i \setminus S_i)$, Steven uses his the reachability strategy to reach a vertex in S_i without doing any resets;
- On $\mathcal{G} \cap (A \setminus S)$, he doesn't reset vertices but moves uses the reachability strategy to S in A ;
- On $\mathcal{G} \cap S$, Steven resets register k .

Observe that this strategy is positional in \mathcal{G} because all the edges taken in \mathcal{G} are those that dictated by the attractor decomposition and the strategy is independent of the register configuration.

Correctness of the strategy. Let us prove now that the strategy defined above is indeed a defensive k -register strategy. We proceed by induction on the height of $\mathcal{T}_{\mathcal{H}}$.

Base Case: If the height of $\mathcal{T}_{\mathcal{H}}$ is 0 then $\mathcal{H} = \langle A, \emptyset \rangle$ and the strategy defined in this case follows the Steven reachability strategy on A , and resets the register 1 when containing value d . This is by definition winning for Steven since as long as register 1 contains d or a higher even priority, every reset of the register is when the vertex has an even priority at least as large as d and therefore a state of rank 3 is never visited and 2 is visited infinitely often.

Inductive step: Let $\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \dots, (S_j, \mathcal{H}_j, A_j) \rangle$ be a Steven attractor decomposition of \mathcal{G} , with Strahler number k and let k_i be the Strahler number of $\mathcal{T}_{\mathcal{H}_i}$ 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 $k_m = k$).

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

We will first show that this is indeed a defensive register strategy by showing that a state of rank $2k + 1$ cannot be reached from a state where the register k contains an even priority. This is because register k is only reset when visiting a vertex of priority d . This ensures that every time Steven resets register k , the strategy leads to a state of rank $2k$. Also, none of the strategies τ_i resets register k .

Consider now any cycle in $\mathcal{R}^k(\mathcal{G})$ using the strategy constructed above. If this cycle contains a state with a vertex of priority d , the highest rank visited in $\mathcal{R}^k(\mathcal{G})$ is $2k$ since the register k is reset at vertices of priority d , since from then on, no state of rank $2k + 1$ can be visited using this strategy, this cycle must be even. Otherwise, the cycle is in $\mathcal{R}^k(\mathcal{G} \cap S_i)$,

where Steven follows a strategy τ_i . We have established that this strategy is also a defensive k -register strategy.

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

We will again show that by following the strategy defined above from a state where register k contains an even priority at least as large as d , then a state of rank $2k + 1$ is not reached in $\mathcal{R}^k(\mathcal{G})$. To show this, observe that register k is reset in the strategy constructed under two circumstances. Either register k is reset at some vertex in S_m or at a vertex of priority d in A . In S_m , by induction hypothesis, we know that as long as register k contains an even priority which is at least as large as $d - 2$, a state of rank $2k + 1$ is not visited and the ranks of the states visited satisfy the parity condition. But between any two visits of S_m in a path in strategy, a vertex of priority d is visited in the underlying parity game. Therefore, we can conclude that the strategy above does not visit a state of rank $2k + 1$ with the mentioned conditions on the start state.

Consider a cycle in the strategy constructed above. If it contains a state with a vertex of priority d , we know then that the registers in the corresponding state would now have an even value at least d in register k . Therefore, from this vertex, as shown above a state of rank $2k + 1$ cannot be reached in the strategy and the highest rank in the cycle is $2k$. If not, then the cycle lies entirely within $\mathcal{R}^k(\mathcal{G} \cap S_i)$ for some i and by induction, this cycle is even. ◀

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 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.*

Before we embark on the proof of this result, we introduce the natural counterpart of Steven defensive dominion strategies: an *Audrey offensive dominion strategy* on a set of states in $\mathcal{R}^k(\mathcal{G})$ is an Audrey strategy that traps Steven in the set, and such that on every infinite path in it, either rank $2k + 1$ occurs at least once, or the largest rank that occurs infinitely many times is odd. Note that Steven defensive dominion strategies can be thought of as winning strategies in a game in which the winning criterion is a conjunction of a parity criterion and a safety criterion, and that Audrey offensive dominion strategies can be thought of as winning strategies in a game in which the winning criterion is a disjunction of the parity and the reachability criteria that are complementary to the corresponding parity and safety criteria for Steven. From determinacy of such games it follows that for every state in $\mathcal{R}^k(\mathcal{G})$, it is either in the largest Steven dominion (on which Steven has a defensive dominion strategy) or it is in the largest Audrey dominion strategy (on which Audrey has an offensive dominion strategy).

A careful reader will also notice that 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, when discussing Audrey strategies here, we explicitly consider the parity criterion on infinite paths. Indeed, in the proof of Theorem 8, we find it convenient to consider non-positional Audrey strategies (even if positional strategies could perhaps be obtained with extra effort), which in turn makes reasoning about the winning criteria on infinite paths more suitable.

In order to prove Theorem 8, we introduce the following definition.

► **Definition 9.** A d -Steven attractor decomposition $\mathcal{H} = \langle A, (S_i, \mathcal{H}_i, A_i)_{i=1}^\ell \rangle$ is tight if for every $i = 1, 2, \dots, \ell$,

- Audrey has a $[Str(\mathcal{T}_{\mathcal{H}_i}) - 1]$ -register dominion strategy on \mathcal{G}'_i ;
- Audrey has an offensive $[Str(\mathcal{T}_{\mathcal{H}_i})]$ -register strategy on $\mathcal{G}'_i \setminus S_i$;

where we define A'_i to be the set of vertices in which Audrey has a reachability strategy to a vertex of priority $d - 1$ in the subgame \mathcal{G}_i and $\mathcal{G}'_i = \mathcal{G}_i \setminus A'_i$. We import the definition of \mathcal{G}_i s from the definition of an attractor decomposition.

► **Lemma 10.** If \mathcal{H} is a tight Steven attractor decomposition of \mathcal{G} , then Audrey has an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy from at least one vertex in \mathcal{G} .

Proof. Let $\mathcal{H} = \langle A, (S_i, \mathcal{H}_i, A_i)_{i=1}^\ell \rangle$ be a d -attractor decomposition. We define A'_i and \mathcal{G}'_i as in Definition 9.

Case 1: $Str(\mathcal{T}_{\mathcal{H}}) = Str(\mathcal{T}_{\mathcal{H}_i})$ for some unique i in $\{1, \dots, \ell\}$. In this case, we show that in \mathcal{G}_i , Audrey has an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy. Since \mathcal{G}_i is a trap for Steven in \mathcal{G} , this gives a strategy for Audrey in \mathcal{G} . Consider the following strategy in \mathcal{G}_i :

- In $\mathcal{G}_i \cap A'_i$, Audrey plays the reachability strategy to a vertex of priority $d - 1$;
- In \mathcal{G}'_i , Audrey plays a $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register dominion strategy. One such strategy exists from the definition of a tight decomposition and the assumption that $Str(\mathcal{T}_{\mathcal{H}}) = Str(\mathcal{T}_{\mathcal{H}_i})$. This strategy is indeed an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy because any play either visits a vertex in A'_i infinitely often or eventually remains in \mathcal{G}'_i . In case A'_i is visited infinitely often, so is a vertex with priority $d - 1$ and therefore Audrey wins. For the latter case, since a $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register dominion strategy is used by Audrey in \mathcal{G}'_i , either a state of rank $2 \cdot Str(\mathcal{T}_{\mathcal{H}}) - 1$ is visited or the ranks of the states visited satisfy the parity condition for Audrey. This ensures that this strategy is indeed an offensive $[Str(\mathcal{T}_{\mathcal{H}})]$ -register strategy.

Case 2: There are two distinct i, j in $\{1, \dots, \ell\}$, $i < j$ such that $Str(\mathcal{T}_{\mathcal{H}_i}) = Str(\mathcal{T}_{\mathcal{H}_j}) = Str(\mathcal{T}_{\mathcal{H}}) - 1$.

We will construct an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy for Audrey from some vertex in \mathcal{G}_j . Observe that S_i is a Steven dominion with priorities at most $d - 2$ and is therefore disjoint from A'_i . Let $Y_i := \mathcal{G}'_i \setminus S_i$. Since \mathcal{H} is tight, Audrey has an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy in Y_i , which we refer to as ω .

We will now describe the offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy for Audrey in \mathcal{G}_i and therefore in \mathcal{G} since \mathcal{G}_i is a trap for Steven in \mathcal{G} :

- In \mathcal{G}_j , as long as register $k - 1$ contains d or a higher even priority, Audrey plays a $[Str(\mathcal{T}_{\mathcal{H}}) - 2]$ -register dominion strategy using the first $[Str(\mathcal{T}_{\mathcal{H}}) - 2]$ registers;
- If the content of register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ is at most $d - 2$:
 - In A'_i , Audrey plays the reachability strategy to a vertex of priority $d - 1$.
 - In Y_i , Audrey plays the offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy ω .
- In \mathcal{G}_i , if register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ contains $d - 1$, Audrey plays an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 2]$ -register dominion strategy using the first $Str(\mathcal{T}_{\mathcal{H}}) - 2$ registers.

Let us consider plays starting in \mathcal{G}_j . If register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ contains d in it, Audrey uses a $[Str(\mathcal{T}_{\mathcal{H}}) - 2]$ -register dominion strategy in \mathcal{G}_j . If Steven never resets register $Str(\mathcal{T}_{\mathcal{H}}) - 1$, then Audrey wins. Once register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ has been reset, the value contained in it is at most $d - 1$. We can therefore reason about plays starting at \mathcal{G}_j such that register $k - 1$ contains priorities at most $d - 1$. We consider a case where the play is at $\mathcal{G}_j \subseteq (Y_i \cup A'_i)$ and register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ contains a value that is at most $d - 2$. Suppose the play never visits A'_i , then the play is at Y_i , where Audrey plays an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register strategy, again making the play winning for Audrey. If not, the play visits A_i and therefore register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ would contain priority $d - 1$. If register $Str(\mathcal{T}_{\mathcal{H}}) - 1$ does contain $d - 1$,

7:10 The Strahler number of a parity game

Audrey plays an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 2]$ -register dominion strategy in \mathcal{G}_i . If Steven never resets register $Str(\mathcal{T}_{\mathcal{H}}) - 1$, then Audrey wins. If Steven resets register $Str(\mathcal{T}_{\mathcal{H}}) - 1$, then a state of rank $2 \cdot Str(\mathcal{T}_{\mathcal{H}}) - 1$ is visited. This shows that the described strategy is an offensive $[Str(\mathcal{T}_{\mathcal{H}}) - 1]$ -register dominion strategy. \blacktriangleleft

► **Corollary 11.** *If there is a tight Steven attractor decomposition \mathcal{H} of \mathcal{G} , then the Lehtinen number of \mathcal{G} and the Strahler number of \mathcal{G} are both equal to $Str(\mathcal{T}_{\mathcal{H}})$.*

Proof. This follows directly from Lemma 7 and Lemma 10. \blacktriangleleft

► **Lemma 12.** *Every Steven dominion has a tight attractor decomposition.*

Proof. Given a game \mathcal{G} with Lehtinen number k , we will construct a tight attractor decomposition inductively. Let d be an even value that is larger than all priorities such that $\pi^{-1}(\{d, d-1\}) \neq \emptyset$.

If $d = 0$, this is trivial as the decomposition is just $\langle A, \emptyset \rangle$ where A is the set of all vertices.

If $d > 1$, let A be the Steven attractor of all the vertices $\pi^{-1}(d)$. Suppose $\mathcal{G}_0 = \mathcal{G} \setminus A = \emptyset$, then $\langle A, \emptyset \rangle$ is a trivial decomposition for \mathcal{G} . If not, then \mathcal{G}_0 is a non-empty trap for Steven in \mathcal{G} and therefore \mathcal{G}_0 has a Lehtinen number that is at most that of \mathcal{G} . Let A' be the Audrey attractor of all the vertices of priority $d-1$ in the sub-game \mathcal{G}_0 and let $\mathcal{G}' = \mathcal{G}_0 \setminus A'$.

Given a positive integer b , we define N^b to be the largest dominion in \mathcal{G}' where Steven has a defensive b -register strategy. We define ℓ to be the smallest value such that $N^\ell \neq \emptyset$ and let $N = N^\ell$. Let \mathcal{H}_0 be the $(d-2)$ -decomposition of N that is tight and obtained by induction.

We show that if $\mathcal{G}_0 \neq \emptyset$ then $N \neq \emptyset$ and ℓ is at most k . To prove this, we will construct that for any b , an offensive b -register strategy on \mathcal{G}_0 when $N^b = \emptyset$. Since Audrey cannot have an offensive k -register strategy in \mathcal{G}_0 , this is equivalent to showing that N is not empty and ℓ , the Lehtinen number of N , is at most k .

The offensive b -register strategy on \mathcal{G}_0 assuming $N^b = \emptyset$ is as follows:

- In A' , Audrey plays the reachability strategy to vertices of priority $d-1$;
- In \mathcal{G} , Audrey plays an offensive b -register strategy on \mathcal{G}' . One such strategy exists since Steven has no defensive b -register strategy in \mathcal{G}' and $N^b = \emptyset$.

Any play following the above strategy and visiting A' infinitely often would visit vertices of priority $d-1$ infinitely often making the play winning for Audrey. If not, then the play eventually stays in \mathcal{G}' and Audrey uses an offensive b -register strategy.

Recall that \mathcal{H}_0 is a tight $(d-2)$ attractor decomposition of N obtained inductively.

Let A_0 be the Steven attractor to N in \mathcal{G}_0 . Now consider $\mathcal{G}_1 = \mathcal{G}_0 \setminus A_0$, which is a trap for Steven and therefore has Lehtinen number at most k . Let $\mathcal{H}' = \langle \emptyset, (S_i, \mathcal{H}_i, A_i)_{i=1}^t \rangle$ be a tight d -attractor decomposition of \mathcal{G}_1 obtained by induction.

We claim $\mathcal{H} = \langle A, (N, \mathcal{H}_0, A_0), (S_i, \mathcal{H}_i, A_i)_{i=1}^t \rangle$ is a tight decomposition of \mathcal{G} . Since \mathcal{H}' is a tight decomposition, we only need to show the following to show that \mathcal{H} is tight:

- Audrey has a $[Str(\mathcal{T}_{\mathcal{H}_0}) - 1]$ -register dominion strategy on \mathcal{G}' ;
- Audrey has an offensive $[Str(\mathcal{T}_{\mathcal{H}_0})]$ -register strategy on $\mathcal{G}' \setminus N$;

From Lemma 10, Audrey has an offensive $[Str(\mathcal{H}_0) - 1]$ -register strategy in N . This shows that $Str(\mathcal{H}_0)$ is at most the Lehtinen number of N . So, $\ell \geq Str(\mathcal{T}_{\mathcal{H}_0})$. From the construction of N , we have the following:

- Audrey has an offensive $(\ell - 1)$ -register strategy on \mathcal{G}' ;
- Audrey has an offensive ℓ -register strategy on $\mathcal{G}' \setminus N$;

We further claim that since Audrey has an offensive $(\ell - 1)$ -register dominion strategy on \mathcal{G}' , she also has an $(\ell - 1)$ -register dominion strategy, if she uses the same strategy. Any infinite play using an offensive $(\ell - 1)$ register strategy either visits a state of rank $2\ell - 1$ infinitely often, or the ranks of the states visited satisfy the parity condition for Audrey.

This shows that we have a tight decomposition for any Steven dominion \mathcal{G} . \blacktriangleleft

► **Corollary 13.** *If Steven has a k -register dominion strategy on \mathcal{G} then he also has one that is based on a positional strategy in \mathcal{G} .*

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 [20] and progress measure lifting algorithm [18, 19] 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 Łazić [19], whose sizes therefore grow faster with size parameters, leading to slower algorithms.

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 $\langle T_1, T_2, \dots, T_k \rangle$ can be embedded in $\langle T'_1, T'_2, \dots, T'_\ell \rangle$ if there are indices i_1, i_2, \dots, i_k such that $1 \leq i_1 < i_2 < \dots < i_k \leq \ell$ and for every $j = 1, 2, \dots, k$, 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. Firstly, define the trivial trees of non-negative heights h by the straightforward induction $I_0 = \langle \rangle$ and $I_h = \langle I_{h-1} \rangle$ for $h \geq 1$. Then, for all positive integers n , we define trees $U_{n,h}^k$ (for all h and k such that $h \geq k \geq 0$) and $V_{n,h}^k$ (for all h and k such that $h \geq k \geq 1$) by mutual induction:

- if $n = 1$, $h = 0$, or $k = 0$ then let $U_{n,h}^k = I_h$;
- if $n = 1$ and $h \geq k \geq 1$ then let $V_{n,h}^k = I_h$;
- if $n \geq 2$ and $h \geq k \geq 1$ then $V_{n,h}^k = V_{\lfloor n/2 \rfloor, h}^k \cdot \langle U_{n,h-1}^{k-1} \rangle \cdot V_{\lfloor n/2 \rfloor, h}^k$;
- if $n \geq 2$ and $h = k \geq 1$ then $U_{n,h}^k = V_{n,h}^k \cdot \langle U_{n,h-1}^{k-1} \rangle \cdot V_{n,h}^k$;
- if $n \geq 2$ and $h > k \geq 1$ then $U_{n,h}^k = V_{n,h}^k \cdot \langle U_{n,h-1}^k \rangle \cdot V_{n,h}^k$.

► **Lemma 14.** *Trees $U_{n,h}^k$ are k -Strahler (n, h) -universal.*

Proof. The proof is by induction on $n + h + k$, and the inductive hypothesis is strengthened to also include *weak k -Strahler (n, h) -universality* of trees $V_{n,h}^k$, where we say that a tree is weakly k -Strahler (n, h) -universal if we can embed in it every (n, h) -small tree in which every tree rooted in a child of the root has Strahler number at most $k - 1$.

7:12 The Strahler number of a parity game

Let T be an (n, h) -small tree of Strahler number at most k and height g . If $n = 1$, $g = 0$, or $k = 0$, then $T = I_g$, and hence T can be embedded in $U_{n,h}^k = I_h$. Likewise, if $n = 1$ then $T = I_g$ and hence, for all $h \geq k \geq 1$, tree T can be embedded in $V_{n,h}^k = I_h$.

Suppose that $T = \langle T_1, \dots, T_j \rangle$ for some positive j . We consider two cases: either $\text{Str}(T_i) \leq k - 1$ for all $i = 1, \dots, j$, or there is p such that $\text{Str}(T_p) = k$. Note that by Proposition 3, the latter case can only occur if $h > k$.

If $\text{Str}(T_i) \leq k - 1$ for all $i = 1, \dots, j$, then we argue that T can be embedded in $V_{n,h}^k$, and hence also in $U_{n,h}^k$, because $V_{n,h}^k$ can be embedded in $U_{n,h}^k$ by definition. Let p (a pivot) be such that both trees $T' = \langle T_1, \dots, T_{p-1} \rangle$ and $T'' = \langle T_{p+1}, \dots, T_j \rangle$ 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 $V_{\lfloor n/2 \rfloor, h}^k$, and tree T_p can be embedded in $U_{n, h-1}^{k-1}$, and hence $T = T' \cdot \langle T_p \rangle \cdot T''$ can be embedded in $V_{n,h}^k = V_{\lfloor n/2 \rfloor, h}^k \cdot \langle U_{n, h-1}^{k-1} \rangle \cdot V_{\lfloor n/2 \rfloor, h}^k$.

If $\text{Str}(T_p) = k$ for some p (the pivot), then we argue that T can be embedded in $U_{n,h}^k$. Note that each of the two trees $T' = \langle T_1, \dots, T_{p-1} \rangle$ and $T'' = \langle T_{p+1}, \dots, T_j \rangle$ is (n, h) -small and all trees T_1, \dots, T_{p-1} and T_{p+1}, \dots, T_j have Strahler numbers at most $k - 1$. It then follows by an argument like in the previous paragraph that each of the two trees T' and T'' can be embedded in $V_{n,h}^k$. Moreover, tree T_p is $(n, h - 1)$ -small and hence, by the inductive hypothesis, tree T_p can be embedded in $U_{n, h-1}^k$. It follows that tree $T = T' \cdot \langle T_p \rangle \cdot T''$ can be embedded in $U_{n,h}^k = V_{n,h}^k \cdot \langle U_{n, h-1}^k \rangle \cdot V_{n,h}^k$. ◀

► **Lemma 15.** For $n \geq 1$ and $h \geq k \geq 1$, we have $\text{leaves} \left(U_{n,h}^k \right) \leq 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h}{k}$.

Proof. The proof is by structural induction, where the inductive hypothesis contains both:

$$\text{leaves} \left(U_{n,h}^k \right) \leq 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h}{k}. \quad (1)$$

and the analogous bound on the number of leaves of trees $V_{n,h}^k$ for all $n \geq 1$ and $h \geq k \geq 1$:

$$\text{leaves} \left(V_{n,h}^k \right) \leq 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k-1}. \quad (2)$$

Firstly, we establish (1) and (2) for $n = 1$ and $h \geq k \geq 1$ by observing that then $U_{n,h}^k = V_{n,h}^k = I_h$ and hence $\text{leaves} \left(U_{1,h}^k \right) = \text{leaves} \left(V_{1,h}^k \right) = 1$.

Secondly, in order to prove (2) for $n \geq 2$ and $h \geq k = 1$, we slightly strengthen the inductive hypothesis to:

$$\text{leaves} \left(V_{n,h}^1 \right) \leq 2^{\lfloor \lg n \rfloor + 1} - 1, \quad (3)$$

which we prove by induction on n . For $n = 1$, we have $V_{n,h}^1 = I_h$ and hence $\text{leaves} \left(V_{1,h}^1 \right) = 1 = 2^{\lfloor \lg 1 \rfloor + 1} - 1$. For $n \geq 2$, we have:

$$\begin{aligned} \text{leaves} \left(V_{n,h}^1 \right) &= \text{leaves} \left(U_{n, h-1}^0 \right) + 2 \cdot \text{leaves} \left(V_{\lfloor n/2 \rfloor, h}^1 \right) \\ &\leq 1 + 2 \left(2^{\lfloor \lg \lfloor n/2 \rfloor \rfloor + 1} - 1 \right) \leq 2^{\lfloor \lg n \rfloor + 1} - 1, \end{aligned}$$

where the first inequality follows from $U_{n, h-1}^0 = I_{h-1}$ and from the strengthened inductive hypothesis.

Thirdly, we establish (1) for $n \geq 2$ and $h = k = 1$:

$$\text{leaves} \left(U_{n,1}^1 \right) = \text{leaves} \left(U_{n,0}^0 \right) + 2 \cdot \text{leaves} \left(V_{n,1}^1 \right) \leq 1 + 2 \left(2^{\lfloor \lg n \rfloor + 1} - 1 \right) \leq 2^{\lfloor \lg n \rfloor + 2},$$

where the first inequality follows from (3).

Fourthly, we establish (1) for $n \geq 2$ and $h > k = 1$:

$$\begin{aligned} \text{leaves}(U_{n,h}^1) &= \text{leaves}(U_{n,h-1}^1) + 2 \cdot \text{leaves}(V_{n,h}^1) \\ &< 2^{\lfloor \lg n \rfloor + 2} \binom{\lfloor \lg n \rfloor}{0} \binom{h-1}{1} + 2 \cdot 2^{\lfloor \lg n \rfloor + 1} = 2^{\lfloor \lg n \rfloor + 2} (h-1+1) = 2^{\lfloor \lg n \rfloor + 2} \binom{\lfloor \lg n \rfloor}{0} \binom{h}{1}, \end{aligned}$$

where the first inequality follows by the inductive hypothesis and by (3).

Fifthly, we establish (2) for $n \geq 2$ and $h \geq k \geq 2$:

$$\begin{aligned} \text{leaves}(V_{n,h}^k) &= \text{leaves}(U_{n,h-1}^{k-1}) + 2 \cdot \text{leaves}(V_{\lfloor n/2 \rfloor, h}^k) \\ &\leq 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 2}{k-2} \binom{h-1}{k-1} + 2 \cdot 2^{\lfloor \lg \lfloor n/2 \rfloor \rfloor + k} \binom{\lfloor \lg \lfloor n/2 \rfloor \rfloor + k - 1}{k-1} \binom{h-1}{k-1} \\ &\leq 2^{\lfloor \lg n \rfloor + k} \left[\binom{\lfloor \lg n \rfloor + k - 2}{k-2} + \binom{\lfloor \lg n \rfloor + k - 2}{k-1} \right] \binom{h-1}{k-1} \\ &= 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k-1}, \end{aligned}$$

where the first inequality follows from the inductive hypothesis and the last equality follows from Pascal's identity.

Finally, we conclude by proving (1) for $n \geq 2$ and $h > k \geq 2$:

$$\begin{aligned} \text{leaves}(U_{n,h}^k) &= \text{leaves}(U_{n,h-1}^k) + 2 \cdot \text{leaves}(V_{n,h}^k) \\ &\leq 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k} + 2 \cdot 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k-1} \\ &\leq 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \left[\binom{h-1}{k} + \binom{h-1}{k-1} \right] \\ &= 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h}{k}, \end{aligned}$$

where the first inequality follows from the inductive hypothesis and the last equality follows from Pascal's identity; and for $n \geq 2$ and $h = k \geq 2$:

$$\begin{aligned} \text{leaves}(U_{n,h}^k) &= \text{leaves}(U_{n,h-1}^{k-1}) + 2 \cdot \text{leaves}(V_{n,h}^k) \\ &\leq 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 2}{k-2} \binom{h-1}{k-1} + 2 \cdot 2^{\lfloor \lg n \rfloor + k} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k-1} \\ &< 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h-1}{k-1} = 2^{\lfloor \lg n \rfloor + k + 1} \binom{\lfloor \lg n \rfloor + k - 1}{k-1} \binom{h}{k}, \end{aligned}$$

where the first inequality follows from the inductive hypothesis and the second inequality holds because the sequence $\binom{\lfloor \lg n \rfloor + i}{i}$ for $i = 0, 1, 2, \dots$ is increasing. \blacktriangleleft

► **Theorem 16.** For $k \leq \lg n$, there are k -Strahler (n, k) -universal trees whose number of leaves 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, for all positive integers n and h , the tree $U_{n,h}^{\lfloor \lg n \rfloor}$ is (n, h) -universal. Theorem 16 implies that the number of leaves of $U_{n,h}^{\lfloor \lg n \rfloor}$ is $n^{\lg(h/\lg n) + O(1)}$, which

matches the asymptotic number of leaves of (n, h) -universal trees of Jurdziński and Lazić [19, Lemma 6]. In particular, if $h = O(\log n)$ then $\lg(h/\lg n) = O(1)$, and hence the number of leaves of $U_{n,h}^{\lfloor \lg n \rfloor}$ is polynomial in n .

Proof of Theorem 16. We analyze in turn the three terms $2^{\lfloor \lg n \rfloor + k + 1}$, $\binom{\lfloor \lg n \rfloor + k - 1}{k-1}$, and $\binom{h}{k}$ in Lemma 15. Firstly, we note that $2^{\lfloor \lg n \rfloor + k + 1}$ is $O(n^{p_1(n,k)})$, where $p_1(n, k) = 1 + k/\lg n$, because $2^k = n^{k/\lg n}$. Secondly, $k \leq \lg n$ implies that $\lfloor \lg n \rfloor + k \leq 2\lg n$, therefore we have $\binom{\lfloor \lg n \rfloor + k - 1}{k-1} \leq 2^{2\lg n} = n^2$, and hence $\binom{\lfloor \lg n \rfloor + k - 1}{k-1}$ is $O(n^{p_2(n,k)})$, where $p_2(n, k) \leq 2$. Thirdly, applying the inequality $\binom{i}{j} \leq (ei/j)^j$ to the binomial coefficient $\binom{h}{k}$, we obtain $\binom{h}{k} \leq (eh/k)^k = 2^{k \lg(eh/k)}$, and hence $\binom{h}{k}$ is $O(n^{p_3(n,h,k)})$, where $p_3(n, h, k) = k \lg(eh/k)/\lg n = k \lg(h/k)/\lg n + k \lg e/\lg n$.

Note that if we let $p(n, h, k) = p_1(n, k) + p_2(n, k) + p_3(n, h, k)$ then the number of leaves in trees $U_{n,h}^k$ is $O(n^{p(n,h,k)})$. Since $k \leq \lg n$ implies $k/\lg n \leq 1$ and $k \lg e/\lg n \leq \lg e$, we obtain $p(n, h, k) \leq k \lg(h/k)/\lg n + 4 + \lg e < k \lg(h/k)/\lg n + 5.45$, and hence the number of leaves in trees $U_{n,h}^k$ is $n^{k \lg(h/k)/\lg n + O(1)}$. If we further assume that $k = o(\log n)$ then the constant 5.45 can be straightforwardly reduced to $3 + o(1)$ because then $k/\lg n$ and $k \lg e/\lg n$ are $o(1)$. Moreover, the estimate $\binom{\lfloor \lg n \rfloor + k - 1}{k-1} = O(n^2)$ can be improved with further assumptions about k as a function of n ; for example, if $k = O(1)$ then $\binom{\lfloor \lg n \rfloor + k - 1}{k-1}$ is only polylogarithmic in n and hence $\binom{\lfloor \lg n \rfloor + k - 1}{k-1}$ is $n^{o(1)}$, bringing $3 + o(1)$ down to $1 + o(1)$. ◀

Efficiently navigating 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 $\langle (a_1, \mathcal{L}_1), (a_2, \mathcal{L}_2), \dots, (a_k, \mathcal{L}_k) \rangle$, where $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k$ are A -labelled ordered trees, and a_1, a_2, \dots, a_k are distinct elements of a linearly ordered set (A, \leq) and $a_1 < a_2 < \dots < a_k$ in that linear order. We define the *unlabelling* of a labelled ordered tree $\langle (a_1, \mathcal{L}_1), (a_2, \mathcal{L}_2), \dots, (a_k, \mathcal{L}_k) \rangle$, by straightforward induction, to be the ordered tree $\langle T_1, T_2, \dots, T_k \rangle$, where T_i is the unlabelling of \mathcal{L}_i for every $i = 1, 2, \dots, k$. 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 $T = \langle T_1, \dots, T_k \rangle$, again by a straightforward induction, to be the \mathbb{N} -labelled tree $\langle (1, \mathcal{L}_1), \dots, (k, \mathcal{L}_k) \rangle$, where $\mathcal{L}_1, \dots, \mathcal{L}_k$ are the natural labellings of trees T_1, \dots, T_k .

For an A -labelled tree $\langle (a_1, \mathcal{L}_1), \dots, (a_k, \mathcal{L}_k) \rangle$, 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 $i = 1, \dots, k$, and where the symbol \cdot denotes concatenation of sequences. For example, the natural labelling of tree $\langle \langle \circ^3 \rangle, \circ^4, \langle \langle \circ \rangle \rangle^2 \rangle$ 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 binary strings: for each binary digit $b \in \{0, 1\}$, 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 all $n \geq 1$ and $h \geq k \geq 1$, we define $\{0, 1\}^*$ -labelled ordered trees $\mathcal{B}_{n,h}^k$ and $\mathcal{C}_{n,h}^k$ as follows:

- $\mathcal{B}_{n,h}^k$ is the tree generated by sequences $\langle \beta_h, \beta_{h-1}, \dots, \beta_1 \rangle$ in which the number t of non-empty bit strings among $\beta_h, \beta_{h-1}, \dots, \beta_1$ is at most k , and the number of bits used in the bit strings $\beta_h, \beta_{h-1}, \dots, \beta_1$ overall is at most $t + \lfloor \lg n \rfloor$;

- $\mathcal{C}_{n,h}^k$ is the tree generated by sequences $\langle \beta_h, \beta_{h-1}, \dots, \beta_1 \rangle$ in which the number t' of non-empty bit strings among $\beta_{h-1}, \beta_{h-2}, \dots, \beta_1$ is at most $k-1$, and the number of bits used in the bit strings $\beta_h, \beta_{h-1}, \dots, \beta_1$ overall is at most $t' + \lfloor \lg n \rfloor$.

► **Lemma 18.** *For all $n \geq 1$ and $h \geq k \geq 0$, the unlabelling of tree $\mathcal{B}_{n,h}^k$ is equal to tree $U_{n,h}^k$.*

Proof. The proof is by induction on $n+h+k$, in which we strengthen the inductive hypothesis by the statement that for all $n \geq 1$ and $h \geq k \geq 1$, the unlabelling of tree $\mathcal{C}_{n,h}^k$ is equal to tree $V_{n,h}^k$.

We define the *trivial* $\{0,1\}^*$ -labelling of tree I_h for any non-negative integer h to be its labelling in which every label is the empty bit string ε . If $n = 1$ then both trees $\mathcal{B}_{n,h}^k$ and $\mathcal{C}_{n,h}^k$ are equal to the trivial $\{0,1\}^*$ -labelling of I_h , and hence their unlabellings are equal to trees $U_{n,h}^k$ and $V_{n,h}^k$, respectively. The same argument applies to $\mathcal{B}_{n,h}^k$ and $U_{n,h}^k$ if $h = 0$ or $k = 0$.

We say that a $\{0,1\}^*$ -labelled ordered tree has a *pivot* if the empty bit string ε is a label of one of the children of the root. Before we consider the three inductive cases (one for $\mathcal{C}_{n,h}^k$ and two for $\mathcal{B}_{n,h}^k$) we define the $(0, \varepsilon, 1)$ -decomposition of any $\{0,1\}^*$ -labelled ordered tree that has a pivot. Recall that in the linear ordering on bit strings that we consider here, all bit strings with a leading 0 are smaller than the empty string ε , and the empty string ε is smaller than all bit strings with a leading 1. For a $\{0,1\}^*$ -labelled ordered tree that has a pivot $\mathcal{L} = \langle (0\beta'_1, \mathcal{L}'_1), \dots, (0\beta'_i, \mathcal{L}'_i), (\varepsilon, \mathcal{L}_p), (1\beta''_1, \mathcal{L}''_1), \dots, (1\beta''_j, \mathcal{L}''_j) \rangle$, its $(0, \varepsilon, 1)$ -decomposition is the triple $(\mathcal{L}', \mathcal{L}_p, \mathcal{L}'')$, where $\mathcal{L}' = \langle (\beta'_1, \mathcal{L}'_1), \dots, (\beta'_i, \mathcal{L}'_i) \rangle$ and $\mathcal{L}'' = \langle (\beta''_1, \mathcal{L}''_1), \dots, (\beta''_j, \mathcal{L}''_j) \rangle$.

For $n \geq 2$ and $h \geq k \geq 1$, tree $\mathcal{C}_{n,h}^k$ has a pivot and hence it has a $(0, \varepsilon, 1)$ -decomposition $(\mathcal{L}', \mathcal{L}_p, \mathcal{L}'')$. We argue that then labelled trees \mathcal{L}' and \mathcal{L}'' are copies of $\mathcal{C}_{\lfloor n/2 \rfloor, h}^k$, and \mathcal{L}_p is a copy of $\mathcal{B}_{n, h-1}^{k-1}$.

Indeed, it follows from the definition of $\mathcal{C}_{n,h}^k$ and from the definition of the $(0, \varepsilon, 1)$ -decomposition that both \mathcal{L}' and \mathcal{L}'' are the $\{0,1\}^*$ -labelled ordered trees generated by sequences $\langle \beta_h, \beta_{h-1}, \dots, \beta_1 \rangle$ in which the number t' of non-empty bit strings among $\beta_{h-1}, \beta_{h-2}, \dots, \beta_1$ is at most $k-1$, and the number of bits used in the bit strings $\beta_h, \beta_{h-1}, \dots, \beta_1$ overall is at most $t' + \lfloor \lg n \rfloor - 1 = t' + \lfloor \lg \lfloor n/2 \rfloor \rfloor$, that is, both \mathcal{L}' and \mathcal{L}'' are copies of $\mathcal{C}_{\lfloor n/2 \rfloor, h}^k$. Similarly, it follows from the definition of $\mathcal{C}_{n,h}^k$ and from the definition of the $(0, \varepsilon, 1)$ -decomposition that \mathcal{L}_p is the $\{0,1\}^*$ -labelled ordered tree generated by sequences $\langle \beta_{h-1}, \beta_{h-2}, \dots, \beta_1 \rangle$ in which the number t' of non-empty bit strings among $\beta_{h-1}, \beta_{h-2}, \dots, \beta_1$ is at most $k-1$, and the number of bits used in the bit strings $\beta_{h-1}, \beta_{h-2}, \dots, \beta_1$ overall is at most $t' + \lfloor \lg n \rfloor$, that is, tree \mathcal{L}_p is a copy of $\mathcal{B}_{n, h-1}^k$.

By the definition of a $(0, \varepsilon, 1)$ -decomposition, the unlabelling of $\mathcal{C}_{n,h}^k$ is the ordered tree $L' \cdot \langle L_p \rangle \cdot L''$, where L', L'' , and L_p are the unlabellings of $\mathcal{L}', \mathcal{L}''$, and \mathcal{L}_p , respectively. By the inductive hypothesis, L' and L'' are copies of tree $V_{\lfloor n/2 \rfloor, h}^k$, and L_p is a copy of tree $U_{n, h-1}^{k-1}$. It then follows that the unlabelling of $\mathcal{C}_{n,h}^k$ is the tree $V_{\lfloor n/2 \rfloor, h}^k \cdot \langle U_{n, h-1}^{k-1} \rangle \cdot V_{\lfloor n/2 \rfloor, h}^k = V_{n,h}^k$.

The arguments for the unlabelling of tree $\mathcal{B}_{n,h}^k$ in two cases $n \geq 2$ and $h = k \geq 1$, and $n \geq 2$ and $h > k \geq 1$, are analogous, and hence we omit them here. ◀

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 $\langle \beta_h, \beta_{h-1}, \dots, \beta_1 \rangle$ in the tree and given a number p , such that $1 \leq p \leq h$, compute the $<_{\text{lex}}$ -smallest leaf $\langle \beta'_h, \beta'_{h-1}, \dots, \beta'_1 \rangle$ in the tree, such that $\langle \beta_h, \dots, \beta_p \rangle <_{\text{lex}} \langle \beta'_h, \dots, \beta'_p \rangle$. As (implicitly) explained by Jurdziński and Lazić [19, 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 19.** *If $k \leq \lg n$ then every leaf in tree $\mathcal{B}_{n,h}^k$ can be represented using $O(\log h \cdot \log n)$ bits and for every $p = 1, 2, \dots, h$, the level- p successor of a leaf in tree $\mathcal{B}_{n,h}^k$ can be computed in time $O(\log h \cdot \log n)$.*

Proof. Consider the following representation of a leaf $\langle \beta_h, \dots, \beta_1 \rangle$ in $\mathcal{B}_{n,h}^k$: for each of the at most $k + \lfloor \lg n \rfloor$ bits used in the bit strings β_h, \dots, β_1 overall, store the value of the bit itself and the number, written in binary, of the component in the h -tuple that this bit belongs to. Altogether, the number of bits needed is $O((k + \lg n) \cdot (\lg h + 1)) = O(\log n \cdot \log h)$.

We now consider computing the level- p successor of a leaf $\ell = \langle \beta_h, \dots, \beta_1 \rangle$ in tree $\mathcal{B}_{n,h}^k$. For every $r = 1, 2, \dots, h$, we let t_r be the number of non-empty bit strings among β_h, \dots, β_r , and we let b_r be the number of bits used in bit strings β_h, \dots, β_r overall. Recall that ℓ is indeed a leaf in $\mathcal{B}_{n,h}^k$ if $t_1 \leq k$ and $b_1 \leq t_1 + \lfloor \lg n \rfloor$.

We split the task of computing the level- p successor ℓ' of leaf ℓ into the following two steps:

- find the lowest ancestor $\langle \beta_h, \dots, \beta_q \rangle$ of $\langle \beta_h, \dots, \beta_p \rangle$ (that is, smallest q satisfying $q \geq p$) that has the next sibling $\langle \beta_h, \dots, \beta_{q+1}, \beta'_q \rangle$ in $\mathcal{B}_{n,h}^k$;
- find the smallest leaf $\ell' = \langle \beta_h, \dots, \beta_{q+1}, \beta'_q, \beta'_{q-1}, \dots, \beta'_1 \rangle$ in the subtree of $\mathcal{B}_{n,h}^k$ rooted at node $\langle \beta_h, \dots, \beta_{q+1}, \beta'_q \rangle$.

For node $\ell_r = \langle \beta_h, \dots, \beta_r \rangle$, where $q \leq r \leq h$, we determine whether it has the next sibling $\ell'_r = \langle \beta_h, \dots, \beta_{r+1}, \beta'_r \rangle$ in $\mathcal{B}_{n,h}^k$ and find it, by considering the following cases.

- If $\beta_r = \varepsilon$ and $t_r = k$ then node ℓ_r does not have the next sibling.
- If $\beta_r = \varepsilon$ and $t_r < k$ then node ℓ_r does have the next sibling and it is obtained by setting $\beta'_r = 10 \cdots 0$, where the number of 0's is such as to make the number of bits used in all bit strings in ℓ'_r equal to $t_r + 1 + \lfloor \lg n \rfloor$.
- If $\beta_r = 1^j$ for some positive j and $b_r = t_r + \lfloor \lg n \rfloor$ then ℓ_r does not have the next sibling.
- If $\beta_r = 1^j$ for some positive j and $b_r < t_r + \lfloor \lg n \rfloor$ then node ℓ_r does have the next sibling and it is obtained by setting $\beta'_r = \beta_r 10 \cdots 0$, where the number of 0's is such as to make the number of bits used in all bit strings in ℓ'_r equal to $t_r + \lfloor \lg n \rfloor$.
- If $\beta_r = \beta 01^j$ for some bit string β and where j is non-negative, then node ℓ_r does have the next sibling and it is obtained by setting $\beta'_r = \beta$.

Let t'_q be the number of non-empty bit strings among $\beta_h, \dots, \beta_{q+1}, \beta'_q$, and let b'_q be the number of bits used by those bit strings overall. Note that for $\langle \beta_h, \dots, \beta_{q+1}, \beta'_q \rangle$ to be a node in $\mathcal{B}_{n,h}^k$, we must have $b'_q \leq t'_q + \lfloor \lg n \rfloor$. We consider the following two cases.

- If $t'_q = k$ then we let $\ell' = \langle \beta_h, \dots, \beta_{q+1}, \beta'_q, \varepsilon, \dots, \varepsilon \rangle$.
- If $t'_q < k$ then we let $\ell' = \langle \beta_h, \dots, \beta_{q+1}, \beta'_q, 0 \cdots 0, \varepsilon, \dots, \varepsilon \rangle$, where the number of 0's in the component $q-1$ is such as to make the number of bits used in all bit strings in ℓ' overall equal to $t'_q + 1 + \lfloor \lg n \rfloor$.

To argue that the above case analyses can be implemented to work in time $O(\log h \cdot \log n)$ is tedious and hence we eschew it. ◀

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 $\langle m_{d-1}, m_{d-3}, \dots, m_1 \rangle$, and for every priority $p \in \{0, 1, \dots, d\}$, we define the p -truncation of $\langle m_{d-1}, m_{d-3}, \dots, m_1 \rangle$, denoted by $\langle m_{d-1}, m_{d-3}, \dots, m_1 \rangle|_p$, to be the sequence $\langle m_{d-1}, \dots, m_{p+2}, m_p \rangle$ if p is odd and $\langle m_{d-1}, \dots, m_{p+3}, m_{p+1} \rangle$ if p is even. We use the lexicographic order \leq_{lex} to linearly order the set $A^* = \bigcup_{i=0}^{\infty} A^i$.

A *Steven progress measure* [7, 18, 19] on a parity game \mathcal{G} is a map $\mu : V \rightarrow A^{d/2}$ 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 20** ([7, 18, 19]). *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 $(n, d/2)$ -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 20 by establishing that the Steven Strahler number and the Steven progress-measure Strahler number of a parity game nearly coincide.

► **Theorem 21.** *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.

Proof of Theorem 21. Let \mathcal{G} be a (n, d) -small parity game. To prove Theorem 21 we will prove the following two lemmas.

► **Lemma 22.** *If \mathcal{G} is a parity game where all the vertices belong to Audrey and \mathcal{G} has a Steven attractor decomposition of Strahler number k , then it has a Steven progress measure of Strahler number at most $k + 1$.*

Proof. Let \mathcal{G} be a parity game where all the vertices belong to Audrey. The proof is by induction on the height of the tree of a Steven attractor decomposition of \mathcal{G} .

Induction hypothesis: Given a d -attractor decomposition \mathcal{H} of \mathcal{G} and its tree $\mathcal{T}_{\mathcal{H}}$ of height h , there is a progress measure tree \mathcal{T} of height h and an embedding f from $\mathcal{T}_{\mathcal{H}}$ to \mathcal{T} such that all the nodes of \mathcal{T} which are not in the image of f are leaves.

Base case: If the height of \mathcal{T} is at most 0, then the d -attractor decomposition is (A, \emptyset) . Let C be the set of vertices which do not have priority d . Consider the topological order: $u < v$ if there is a path from v to u in A . We consider the tree $\langle \circ^{|C|} \rangle$ and μ which maps the vertices of priority d to its root and the vertices in C to leaves, respecting the topological order, i.e. if $u < v$ then u is mapped to a node on the right of the node v is mapped to. This defines a progress measure of Strahler number 1.

7:18 The Strahler number of a parity game

Induction step. Consider a Steven- d -attractor decomposition:

$$\mathcal{H} = \langle A, (S_1, \mathcal{H}_1, A_1), \dots, (S_j, \mathcal{H}_j, A_j) \rangle$$

and let $\mathcal{T}_{\mathcal{H}_i}$ be the tree of \mathcal{H}_i . Let \mathcal{G}_i defined by $\mathcal{G}_1 = \mathcal{G}$ and $\mathcal{G}_{i+1} = \mathcal{G}_i \setminus A_i$.

Inductively, for all i , there is a progress measure tree \mathcal{T}_i (and an associated progress measure mapping μ_i) of the same height as $\mathcal{T}_{\mathcal{H}_i}$ and an embedding f_i from $\mathcal{T}_{\mathcal{H}_i}$ to \mathcal{T}_i such that all the nodes of \mathcal{T}_i which are not in the image of f_i are leaves.

Let us construct a progress measure tree for \mathcal{G} as follows. Let $C_i = A_i \setminus S_i$ for each i and C be the set of nodes in A that have priority at most $d - 1$. Set:

$$\mathcal{T} = \langle \circ^{|C|}, \mathcal{T}_1, \circ^{|C_1|}, \dots, \mathcal{T}_j, \circ^{|C_j|} \rangle$$

Set μ to be a mapping from the set of vertices of \mathcal{G} to the nodes of \mathcal{T} which extends μ_i on vertices in S_i , maps the vertices of priority d to the root of the tree, the vertices in C to the first $|C|$ children of the root and the vertices in C_i to the corresponding $|C_i|$ children of the root which respects the topological ordering in \mathcal{G} as viewed as a graph, i.e, if for vertices u and v in C , resp. C_i , there is a path from u to v in C , resp. C_i , then u is mapped to a node that appears on the right of the node v is mapped to.

By construction and induction hypothesis, the tree \mathcal{T} embeds $\mathcal{T}_{\mathcal{H}}$ and the only nodes that are not images of nodes in $\mathcal{T}_{\mathcal{H}}$ are leaves. Moreover, \mathcal{T} is a progress measure tree with mapping μ by induction hypothesis, and the construction which is compatible with the Steven reachability strategy on A , and the A_i 's.

The lemma follows from the fact that the Strahler number of a tree increases by at most 1 when leaves are added to it. \blacktriangleleft

► **Lemma 23.** *If \mathcal{G} has a Steven progress measure of Strahler number k , then it has a Steven attractor decomposition of Strahler number at most k .*

Proof. We will prove the following by induction, which proves the lemma:

Induction Hypothesis on n : Given an (n, d) -small parity game \mathcal{G} where d is the least even integer no smaller than any priority in \mathcal{G} and a progress measure tree \mathcal{T} on \mathcal{G} , there exist a Steven attractor decomposition whose tree embeds in \mathcal{T} .

► **Remark 24.** Given a progress measure mapping μ on \mathcal{G} and its corresponding progress measure tree \mathcal{T} , and given a trap R for Audrey in \mathcal{G} , the restriction of μ to the vertices in R is a progress measure with the tree induced by the nodes images of the vertices of R by μ .

Base case: For games with one vertex, any progress measure tree on \mathcal{G} and any tree of a Steven attractor decomposition are $\langle \rangle$. Therefore the induction hypothesis is satisfied.

Induction step. Let \mathcal{G} be an (n, d) -small parity game where d is the least even integer no smaller than any priority in \mathcal{G} and let \mathcal{T} be a progress measure tree on \mathcal{G} .

Case 1: If the highest priority in \mathcal{G} is even, i.e. equal to d . Let A be the Steven attractor of the set of vertices of priority d . Let $\mathcal{G}' = \mathcal{G} \setminus A$. As \mathcal{G}' is a trap for Audrey in \mathcal{G} , the tree \mathcal{T}' induced by the nodes images of the vertices in \mathcal{G}' in \mathcal{T} is a progress measure tree of \mathcal{G}' . By induction hypotheses, there exist a Steven attractor decomposition \mathcal{H} of \mathcal{G}' whose tree $\mathcal{T}_{\mathcal{H}}$ embeds in \mathcal{T}' . By appending A to \mathcal{H} , one gets a Steven attractor decomposition of \mathcal{G} of same tree $\mathcal{T}_{\mathcal{H}}$, which then embeds in \mathcal{T} .

Case 2: If the highest priority in \mathcal{G} is odd, i.e. equal to $d - 1$.

No vertex is mapped to the root in the progress measure tree \mathcal{T} . Let $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_j$ be the subtrees, children of the root of \mathcal{T} . Let us note that vertices of priority $d - 1$ cannot be mapped to nodes in \mathcal{T}_0 as they would not have progressive outgoing edges if that was the case. Let S_0 be the set of vertices mapped to nodes in \mathcal{T}_0 and let A_0 be the Steven attractor of S_0 in \mathcal{G} . We can assume that S_0 is non empty (otherwise we remove \mathcal{T}_0 from \mathcal{T} and start again).

Let $\mathcal{G}' = \mathcal{G} \setminus A_0$. As \mathcal{G}' is a subgame, trap for Audrey, the tree \mathcal{T}' with subtrees $\mathcal{T}_1, \dots, \mathcal{T}_j$ is a progress measure tree on \mathcal{G}' . By induction, one gets a Steven attractor decomposition:

$$\mathcal{H}' = \langle \emptyset, (S_1, \mathcal{H}_1, A_1), \dots, (S_j, \mathcal{H}_j, A_j) \rangle$$

whose tree embeds in \mathcal{T}' .

Now, let us prove that S_0 is a trap for Audrey. Let u be in S_0 and v be one of its successor. For (u, v) to be progressive, v has to be mapped to a node in \mathcal{T}_0 and is then in S_0 . Since there is always an outgoing progressive edge for Steven's vertices and all edges of Audrey's vertices are progressive, we can conclude that S_0 is a trap for Audrey, is a sub-game, and \mathcal{T}_0 is a progress measure tree on it. By induction, one gets a Steven attractor decomposition \mathcal{H}_0 of S_0 , whose tree embeds in \mathcal{T}_0 .

We have proved that:

$$\mathcal{H} = \langle \emptyset, (S_0, \mathcal{H}_0, A_1), (S_1, \mathcal{H}_1, A_1), \dots, (S_j, \mathcal{H}_j, A_j) \rangle$$

is a Steven attractor decomposition of \mathcal{G} whose tree embeds in \mathcal{T} . ◀

Lemma 23 gives one direction of the theorem. For the reverse direction, consider \mathcal{G} a parity game and \mathcal{H} a Steven attractor decomposition of Strahler number k . This decomposition induces a winning strategy for Steven (with exactly one edge going out any vertex owned by Steven in \mathcal{G}). Consider the restriction of \mathcal{G} to this Steven strategy. This is a game where all the vertices belong to Audrey, and which has \mathcal{H} as a Steven attractor decomposition. We can apply Lemma 23 and obtain a Steven progress measure of Strahler number at most $k + 1$. By definition, this progress measure is also a progress measure of \mathcal{G} , which concludes the proof. ◀

7 Strahler-universal progress measure lifting algorithm

Jurdziński and Lazić [19, Section IV] have implicitly suggested that the progress-measure lifting algorithm [18] 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 [19, 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 [19, 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 $k \cdot \lg(d/k) = O(\log n)$. The algorithm does not need to be given (an upper bound on) the Strahler number of the game as input.*

Proof. By Proposition 3, we may assume that $k \leq \lg n$. 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 $\mathcal{B}_{n, d/2}^{k+1}$, which is $(k+1)$ -Strahler $(n, d/2)$ -universal by Lemmas 14 and 18. By Theorem 21 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 [19, Theorem 7], and by Lemma 19.

We now argue that if the Strahler number k is not known in advance, we can still achieve the running time claimed: it suffices to run the algorithm simultaneously for Steven and for Audrey, using i -Strahler $(n, d/2)$ -universal trees $\mathcal{B}_{n, d/2}^i$ for increasing values of $i = 0, 1, 2, \dots$, until the Steven and Audrey dominia returned by the two procedures are a partition of the set of vertices of the game. By Theorem 2, Theorem 21, Proposition 25, and by the definition of the Strahler number of a parity game, the above iteration will terminate for some $i \leq k+1$. The running time bound follows from observing that the numbers of leaves of trees $\mathcal{B}_{n, d/2}^i$ are increasing in i , and from $k \leq \lg n$. ◀

► **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 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 [19, Theorem 7] (set $k = \lg n$ and use Propositions 4 and 3 to justify it), and the $k = O(1)$ criterion of Lehtinen [23, 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, and Lemmas 18 and 19) 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 $O(\sqrt{\log n})$, i.e., d is $2^{O(\sqrt{\log n})}$. 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 [19, 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 $O(\sqrt{\log n})$ rate of growth allowed in this scenario for the Strahler number k substantially exceeds $k = O(1)$ required by Lehtinen [23, 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. Khoushainov, 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 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.
- 8 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.
- 9 A. P. Ershov. On programming of arithmetic operations. *Communications of the ACM*, 1(8):3–6, 1958.
- 10 J. Esparza, M. Luttenberger, and M. Schlund. A brief history of Strahler numbers—with a preface. Technical report, Technical University of Munich, 2016.
- 11 J. Fearnley. Exponential lower bounds for policy iteration. In *ICALP 2010*, volume 6199 of *LNCS*, pages 551–562, Bordeaux, France, 2010. Springer.
- 12 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.
- 13 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.
- 14 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.
- 15 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.
- 16 E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.
- 17 D. Hausmann and L. Schröder. Computing nested fixpoints in quasipolynomial time. arXiv:1907.07020, 2019.
- 18 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.
- 19 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.

- 20 M. Jurdziński and R. Morvan. A universal attractor decomposition algorithm for parity games. arXiv:2001.04333, 2020.
- 21 M. Jurdziński, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM Journal on Computing*, 38(4):1519–1532, 2008.
- 22 D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1973.
- 23 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.
- 24 K. Lehtinen, S. Schewe, and D. Wojtczak. Improving the complexity of Parys’ recursive algorithm, 2019. arXiv:1904.11810.
- 25 R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65(2):149–184, 1993.
- 26 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.
- 27 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.
- 28 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.
- 29 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.
- 30 W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.