

# A Collapse of the Parity Index Hierarchy of Tree Automata, Based on Cantor-Bendixson Ranks

Karoliina Lehtinen ✉ 

CNRS, Aix-Marseille Université, LIS, France

Nathan Lhote ✉ 

Aix-Marseille Université, LIS, France

---

## Abstract

Over words, nondeterministic Büchi automata and alternating weak automata are as expressive as parity automata with any number of priorities. Over trees, the Büchi acceptance condition is strictly weaker and the more priorities we allow, the more languages parity automata can recognise. We say that on words, the parity-index hierarchies of nondeterministic and alternating automata collapse to the Büchi and weak level, respectively, while both are infinite over trees.

We ask *when is Büchi enough?*, that is, on which classes of trees are nondeterministic Büchi automata as expressive as parity automata. Similarly for alternating weak automata. We work in the setting of unranked unordered trees, in which there is no order among the children of nodes.

We find that for nondeterministic and alternating automata, the parity-index hierarchy collapses to the Büchi level and weak level, respectively, for any class of trees of finitely bounded Cantor-Bendixson rank, a topological measure of tree complexity. Over trees of countable Cantor-Bendixson rank, (a.k.a. thin trees) the parity-index hierarchy of both nondeterministic and alternating automata collapses to the level  $[1, 2, 3]$ , as was already known for ordered trees. These results are in some sense optimal: on the class of trees of finite but unbounded Cantor-Bendixson rank, two priorities do not suffice to recognise all parity-recognisable languages, even for alternating automata.

**2012 ACM Subject Classification** Theory of computation → Formal languages and automata theory

**Keywords and phrases** Parity tree automata, alternating automata, Cantor-Bendixson rank

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2025.164

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

## 1 Introduction

Over infinite words, the languages recognised by nondeterministic parity automata are the same as those recognised by nondeterministic Büchi automata, namely all  $\omega$ -regular languages. For alternating automata, an even simpler acceptance condition called *weak*, in which accepting and rejecting cycles do not intersect, suffices to achieve the same expressivity. In contrast, over infinite trees, not only are parity automata more expressive than Büchi automata, but each additional priority increases the expressivity of parity automata [2, 12], both in the nondeterministic and alternating case. We say that on words, the nondeterministic and alternating parity-index hierarchies, also known as the Rabin-Mostowski hierarchies, collapse to the Büchi and weak level, respectively, while on trees, these hierarchies are infinite.

Here, we are interested in structures between words and arbitrary trees, and ask when does the parity-index hierarchy collapse, and to which level? That is, for which classes of trees do nondeterministic Büchi automata and alternating weak automata capture the same languages as parity automata? What about acceptance conditions between Büchi and parity?

There is a dual motivation for this problem. The decidability of the parity-index hierarchy – that is, whether one can decide the minimal number of priorities required to recognise a language – is a long-standing open problem for both nondeterministic and alternating tree automata, while it is straight-forward for word automata. So far, the problem of deciding



© Karoliina Lehtinen and Nathan Lhote;  
licensed under Creative Commons License CC-BY 4.0

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).

Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis

Article No. 164; pp. 164:1–164:20



Leibniz International Proceedings in Informatics

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



whether a tree language is expressible with a fixed number of priorities is only known to be decidable for the full hierarchy for deterministic automata [14], game automata [6] and guidable automata [13], which are a form of mildly nondeterministic automata. Results about these hierarchies over restricted classes are limited: over planar graphs, they are infinite [3]; over classes of ordered trees with a countable number of infinite branches, they collapse [7] (see comparison in Remark 27). Here we show the relevance of tools from the parity game literature – in particular attractor decompositions – for analysing the index hierarchy.

From a more practical perspective, defining and understanding properties that use many priorities (or alternating fixpoints in the  $\mu$ -calculus) is notoriously hard, so it is interesting to understand when a Büchi condition (one fixpoint alternation in the  $\mu$ -calculus), or a weak condition (alternation-free  $\mu$ -calculus) suffices.

The setting in which we study this problem is that of alternating and nondeterministic parity automata over unordered trees, that is, trees of unbounded and not necessarily finite arity in which the children of nodes are not ordered (nor labelled with *left* and *right*). The trees are coloured with letters from a finite alphabet, and processed by tree automata. Our automata formalism captures the expressivity of the modal  $\mu$ -calculus (see for example [15]), and the parity index coincides with the fixpoint alternation depth. We study the expressivity of these automata over classes of trees, of which the trees are coloured from a finite alphabet  $\Sigma$  in all possible ways: for instance, the singleton class containing a tree with a single infinite branch produces all words in  $\Sigma^\omega$ . We do not consider restrictions on the colouring.

We identify the key parameter that determines the collapse of the parity-index hierarchy to be the Cantor-Bendixson rank of trees. This notion originates from topology, see for example [10], and has been used to describe the structural complexity of trees. In particular thin trees (trees with countably many branches) are exactly the trees of Cantor-Bendixson rank less than  $\omega_1$ , the first uncountable ordinal.

Our main result is that on the class  $\mathcal{C}_k$  of trees of Cantor-Bendixson rank less than  $k \in \mathbb{N}$ :

1. For any nondeterministic parity automaton  $\mathcal{A}$ , there exists a nondeterministic Büchi automaton  $\mathcal{B}$  such that  $L(\mathcal{A}) \cap \mathcal{C}_k = L(\mathcal{B}) \cap \mathcal{C}_k$  (Theorem 23);
2. For any alternating parity automaton  $\mathcal{A}$ , there exists an alternating weak automaton  $\mathcal{B}$  such that  $L(\mathcal{A}) \cap \mathcal{C}_k = L(\mathcal{B}) \cap \mathcal{C}_k$  (Corollary 24).

For completeness, although a related result is already known for ordered trees [7]<sup>1</sup>, we also show that on the class  $\mathcal{C}_{\omega_1}$  of thin trees:

1. For any nondeterministic parity automaton  $\mathcal{A}$ , there exists a nondeterministic  $[1, 2, 3]$ -automaton  $\mathcal{B}$  such that  $L(\mathcal{A}) \cap \mathcal{C}_{\omega_1} = L(\mathcal{B}) \cap \mathcal{C}_{\omega_1}$  (Theorem 25);
2. For any alternating parity automaton  $\mathcal{A}$ , there exists an alternating  $[1, 2, 3]$ - and  $[0, 1, 2]$ -automaton  $\mathcal{B}$  such that  $L(\mathcal{A}) \cap \mathcal{C}_{\omega_1} = L(\mathcal{B}) \cap \mathcal{C}_{\omega_1}$  (Corollary 26).

Trees that don't have a countable Cantor-Bendixson rank actually don't have a *defined* Cantor-Bendixson rank (for countable trees) and are known to admit the full binary tree as a minor. For those trees, the parity index hierarchy is known to be infinite [1]. This completes the trichotomy on the classes of trees parameterised by their Cantor-Bendixson rank.

We also show that these results are optimal in the sense that there are languages not recognised by alternating Büchi automata over the class  $\mathcal{C}_\omega$  of trees of finite Cantor-Bendixson rank. This implies that *i*) neither alternating weak automata nor nondeterministic Büchi automata suffice to capture the whole hierarchy on the class of trees of finite Cantor-Bendixson rank, and *ii*) automata with only two priorities do not suffice to capture the whole hierarchy over trees of countable Cantor-Bendixson rank.

---

<sup>1</sup> Note that deducing the result in one setting from the other is harder than proving it afresh.

To obtain these results, we generalise the Cantor-Bendixson rank from trees to graphs, in order to study the complexity of the acceptance parity games induced by an automaton on the class of trees of interest. We give a new characterisation of the Cantor-Bendixson rank. A tree has Cantor-Bendixson rank at least  $k$  if it contains  $k$  disjoint sets of vertices  $(V_0, \dots, V_{k-1})$  such that every vertex of  $V_i$  has both a successor within  $V_i$  and a path to  $V_{i-1}$ . This property extends as is to graphs, which we believe to be an interesting measure of complexity in its own right. Our main technical result is that if  $\mathcal{C}$  has a finitely bounded Cantor-Bendixson rank, then the membership parity games induced by any automaton also have a finitely bounded rank. We then use the notion of *attractor decomposition*, which corresponds to the decomposition computed by, for instance, Zielonka-type algorithms for parity games, and which has also been used in the context of quasipolynomial algorithms for parity games [9] and automata transformations [11, 4]. We extend attractor decompositions to infinite parity games, and argue that games of finitely bounded rank have finitely bounded attractor decomposition and therefore the winning regions of parity games of finitely bounded rank are recognisable by Büchi automata. We then transform, for any  $k \in \mathbb{N}$ , any alternating parity automaton into a Büchi automaton that is equivalent over all trees of Cantor-Bendixson rank up to  $k$ . If the input automaton is nondeterministic so is the output automaton. Furthermore, if the input automaton is a coBüchi automaton, this transformation yields a weak automaton. Then, via double complementation, we get an equivalent alternating weak automaton.

Beyond proving these new results on the parity-index hierarchy of tree automata, we believe our technical developments provide some insights into the relevance of attractor decompositions for approaching the parity-index hierarchy. We also believe that the notion of rank we introduce, which extends Cantor-Bendixson rank from trees to graphs, can be a useful tool to study structural properties of graphs.

Proofs that are routine or uninformative are in the appendix.

## 2 Graphs, automata and parity games

**Notation.** We denote by  $\omega$  the first infinite countable ordinal and  $\omega_1$  the first uncountable ordinal. Given a set  $S$ , and an ordinal  $\kappa$ , we define  $S^\kappa$  the set of sequences of order-type  $\kappa$  over  $S$ . The length of a sequence  $s \in S^\kappa$  is denoted by  $|s| = \kappa$ . Given  $s$  of length  $\kappa \leq \omega$ , we use  $s[i]$  to refer to the  $i^{\text{th}}$  element of a sequence  $s$  and  $s[i..j]$  to refer to the subsequence  $s[i] \cdots s[j]$  for,  $i \leq j < \kappa$ . Finally we denote by  $S^* = \bigcup_{n < \omega} S^n$  the set of finite sequences over  $S$ . Two sequences  $u, v \in S^\omega$  are called *ultimately equal* if they only differ by a finite prefix. They are called *ultimately disjoint* if they have no element in common, up to removing finite prefixes.

**Graphs.** A *graph*  $G = (V, E)$  consists of a possibly infinite but countable set of vertices  $V$  and a set of directed edges  $E \subseteq V \times V$ . A *subgraph* of  $G$  is a graph  $G' = (V', E')$  with  $V' \subseteq V$ , and  $E' \subseteq E$ , which we denote by  $G' \subseteq G$ . The subgraph of  $G$  *induced by* a subset  $V' \subseteq V$  is  $(V', E \cap (V' \times V'))$  and denoted by  $G \upharpoonright V'$ . Given  $G_1 = (V_1, E_1)$ , and  $G_2 = (V_2, E_2)$  two induced subgraphs of  $G$ , we denote by  $G_1 \cap G_2$  the subgraph induced by  $V_1 \cap V_2$ . The subgraphs  $G \setminus V'$  and  $G \setminus E'$  for  $V' \subseteq V$  and  $E' \subseteq E$  are  $G \upharpoonright (V \setminus V')$  and  $(V, E \setminus E')$ , respectively. A subset  $V' \subseteq V$  of vertices is *closed under successors* in  $G$  if for every vertex  $v' \in V'$  and edge  $(v', v) \in E$ , we have  $v \in V'$ . A vertex is *terminal* if it has no successor. We write  $K_n$  for the clique of size  $n$ , that is, the graph with vertices  $[1 \dots n]$  and edges  $[1 \dots n]^2$ .

A path in  $G$  is a sequence  $e_0 e_1 \cdots \in E^*$  or  $E^\omega$  of successive edges (i.e.  $(v_0, v_1)(v_1, v_2) \dots$ ). We call  $\omega$ -path, or infinite path, a path which is not finite. When convenient, we also refer to a path by its sequence of vertices  $v_0 v_1 \cdots \in V^*$  or  $V^\omega$ . We also call a graph  $G = (V, E)$  an infinite path if  $V = \{v_i \mid i \in \mathbb{N}\}$  and  $E = \{(v_i, v_{i+1}) \mid i \in \mathbb{N}\}$ . Extending definitions, we say that a set of vertices  $V' \subseteq V$  is a path if  $G|V'$  is a path.

A graph is a *tree* if there exists a vertex  $r$ , called the root, such that for any vertex  $v$ , there is a unique path from  $r$  to  $v$ .

The (synchronous) product of two graphs  $G = (V, E), H = (W, F)$  is the graph  $G \times H$  with vertices  $V \times W$  and edges  $\{((v, w), (v', w')) \mid (v, v') \in E, (w, w') \in F\}$ . Given a vertex  $(v, w) \in V \times W$  we denote by  $\pi(v, w) = v$  the projection on the first coordinate. We extend this notation to vertex sets and vertex sequences in the natural way.

A  $\Sigma$ -coloured tree, or simply  $\Sigma$ -tree,  $(V, E, r, L)$ , consists of a tree  $(V, E)$ , an edge colouring  $L : E \rightarrow \Sigma$  and its root  $r \in V$ . A class  $\mathcal{C}$  of trees and an alphabet  $\Sigma$  generate the class  $\mathcal{C}^\Sigma$  consisting of all  $\Sigma$ -trees  $(V, E, r, L)$  such that  $(V, E) \in \mathcal{C}$ .

The *attractor* of a set of vertices (or edges)  $X$  in a graph  $S$ , denoted by  $\text{Attr}(X, S)$  (or  $\text{Attr}(X)$  if  $S$  is clear from context), is the set of vertices  $v$  in  $S$ , such that all maximal paths of  $S$  starting in  $v$  intersect with  $X$  (contain an edge in  $X$ ). In particular, if  $X$  is a set of vertices, it is contained in  $\text{Attr}(X, S)$ ; if  $X$  is a set of edges, neither of its endpoints is necessarily in  $\text{Attr}(X, S)$ . (This is the restriction of the typical definition of attractor to a one-player game.)

**Parity Games.** A parity game  $(V_E \uplus V_A, E, \Omega : E \rightarrow I, v_i)$  is a two-player game, played on a graph without terminal vertices, rooted at an initial position  $v_i$ , coloured by a priority assignement  $\Omega$  with non-negative integer priorities from  $I = 0, \dots, h$  or  $I = 1, \dots, h$  for some  $h \in \mathbb{N}$ , in which vertices (called positions) are partitioned between those belonging to Eve,  $V_E$  and those belonging to Adam,  $V_A$ . A play of a parity game is a path in  $G$ . An infinite play is winning for Eve if the highest priority that occurs infinitely often is even.

A strategy for Eve is a function  $\sigma : V^* V_E \rightarrow V$  such that  $(v, \sigma(wv)) \in E$ . A play  $p$  agrees with a strategy  $\sigma$  for Eve if for all  $p[i] \in V_E$ ,  $p[i+1] = \sigma(p[1..i])$ . Adam's strategies are defined similarly from  $V^* V_A$ . A strategy for a player is winning if all the plays that agree with it are winning for that player. Parity games admit positional strategies [16], that is, the winning player always has a strategy  $\sigma$  such that  $\sigma(uv) = \sigma(u'v)$  for all  $u, u' \in V^*, v \in V$ , and can therefore be written as a function  $\sigma : V_P \rightarrow V$  for  $P \in \{A, E\}$ .

**Tree automata.** An alternating parity tree automaton (APT)  $\mathcal{A} = (\Sigma, Q, \delta, \delta_i, \Omega)$  over a finite alphabet  $\Sigma$  consists of: a finite set of states  $Q$ ; a transition function  $\delta : Q \times \Sigma \rightarrow \mathbb{T}(Q)$ , where  $\mathbb{T}(Q)$  consists of *transition conditions*, that is, positive modal formulas over  $Q$ :  $f, f' \in \mathbb{T}(Q) := f \wedge f' \mid f \vee f' \mid \Box f \mid \Diamond f \mid q \in Q$ ; we further require that every occurrence of a state  $q \in Q$  in a transition condition is within the scope of exactly one modal operator from  $\{\Box, \Diamond\}$ ; and a priority assignement  $\Omega : Q \rightarrow I$  where  $I$  is a set  $[i..j]$  of consecutive integers starting from  $i \in \{0, 1\}$ .  $\delta_i$  is an initial transition condition in  $\mathbb{T}(Q)$ . We write  $|\mathcal{A}|$  for the size of  $\mathcal{A}$ , that is, the total number of subformulas in all the transition conditions in  $\delta$ .

An APT is also a Büchi automaton (ABT) if the set of priorities is  $\{1, 2\}$ . In this case, states with priority 2 are called “accepting” and with priority 1 “non-accepting”. An APT or ABT is nondeterministic (NPT or NBT) if the only conjunctions that occur are of the following form for a set  $F$  of states [8]:  $\bigwedge_{q \in F} \Diamond q \wedge \Box \bigvee_{q \in F} q$ . These correspond to disjunctive formulas of the modal  $\mu$ -calculus and are as expressive as APT [8]. Here we will only use the fact that adding disjunctive choices keeps an automaton nondeterministic.

We define acceptance via parity games. Given a  $\Sigma$ -tree  $T = (V, E, r, L)$  and automaton  $\mathcal{A} = (\Sigma, Q, \delta, \delta_\varepsilon, \Omega : Q \rightarrow [i..j])$ , the *acceptance game*  $\mathcal{G}(T, \mathcal{A})$  is the parity game given by:

- A set of vertices  $(V \times \mathbb{T}(Q) \times (\Sigma \cup \{\varepsilon\})) \cup \{S_E, S_A\}$  to indicate the vertex in the graph, the state or transition condition of the automaton and the label currently processed by the automaton;  $S_E$  and  $S_A$  are sinks winning for Eve and Adam respectively.
- A set of edges (where  $v \in V$ ,  $x \in \Sigma$ ,  $q \in Q$  and  $b, b' \in \mathbb{T}(Q)$ ):
  - $(v, q, x) \rightarrow (v, \delta(q, x), \varepsilon)$ ;
  - $(v, b \wedge b', x) \rightarrow (v, b, x)$  and  $(v, b \wedge b', x) \rightarrow (v, b', x)$ ,
  - $(v, b \vee b', x) \rightarrow (v, b, x)$  and  $(v, b \vee b', x) \rightarrow (v, b', x)$ ,
  - $(v, \Box b, \varepsilon) \rightarrow (v', b, x)$  for all  $v'$  such that  $(v, v') \in E$  and  $L(v, v') = x$
  - $(v, \Box b, \varepsilon) \rightarrow S_E$  and  $S_E \rightarrow S_E$ ;
  - $(v, \Diamond b, \varepsilon) \rightarrow (v', b, x)$  for all  $v'$  such that  $(v, v') \in E$  and  $L(v, v') = x$ ;
  - $(v, \Diamond b, \varepsilon) \rightarrow S_A$  and  $S_A \rightarrow S_A$ .
- Positions  $(v, b \wedge b', x)$  and  $(v, \Box q, x)$  belong to Adam;  $(v, b \vee b', x)$  and  $(v, \Diamond b, \varepsilon)$  belong to Eve. These are of the minimal priority  $i$ . Outgoing edges from positions  $(v, q, x)$  are of priority  $\Omega(q)$ ; the self-loop on  $S_E$  is of minimal even priority; the one on  $S_A$  is of minimal odd priority. These positions have a single successor so their owner is irrelevant.
- An initial position  $(r, \delta_\varepsilon, \varepsilon)$ .

$\mathcal{A}$  accepts a  $\Sigma$ -tree  $t$  with root  $r$  if Eve has a winning strategy in  $\mathcal{G}(t, \mathcal{A})$  from  $(r, \delta_\varepsilon, \varepsilon)$ .  $\mathcal{A}$  *recognises* the language  $L(\mathcal{A})$  of  $\Sigma$ -trees accepted by  $\mathcal{A}$ . Two APTs  $\mathcal{A}$  and  $\mathcal{B}$  over an alphabet  $\Sigma$  are *equivalent* if they recognise the same language, namely if  $L(\mathcal{A}) = L(\mathcal{B})$ . They are *equivalent over a class  $\mathcal{C}$  of trees*, or *agree on  $\mathcal{C}$* , if for every rooted  $\Sigma$ -tree  $G \in \mathcal{C}^\Sigma$ , either both accept  $G$  or both reject  $G$ .

► **Remark 1.** Our alternating tree automata are equivalent to the  $\mu$ -calculus, typically defined over trees (or graphs) with arbitrary branching and unordered successors. Different versions of the  $\mu$ -calculus describe properties of edge- and vertex-coloured trees. We colour edges in trees for technical convenience, but this makes little difference.

### 3 Decompositions, patterns and Cantor-Bendixson rank

In this section we introduce three measures of graph complexity. The first is the attractor decomposition of parity graphs [9, 4] extended here to infinite graphs. The second is the presence of what we call *patterns* which is new to the best of our knowledge. The third is the Cantor-Bendixson rank, a measure of complexity of topological spaces, which can also be used for trees [10].

#### 3.1 Parity graph decompositions

A parity graph is a graph without terminal vertices, edge-coloured with priorities, namely with a set of consecutive integers  $[i..j]$ , where  $i \in \{0, 1\}$ . A parity graph is *even* if on all infinite paths, the highest priority occurring infinitely often is even. In particular, the highest priority on every cycle must be even.

An attractor decomposition of a parity graph  $G$ , if it exists, is a witness that the graph is even. It was introduced in the form we use here under the name “hierarchical decomposition” in [4, Proof of Theorem 6], and later called attractor decomposition and applied to parity games, rather than just graphs in [5, 9]. Our presentation mostly follows [5], restricted to one-player games in which all positions belong to Adam, but extended to infinite graphs. Informally, for the maximal even priority  $h$ , a  $\kappa$ -width attractor decomposition identifies

a set of vertices (region)  $A_0$  from which every maximal path visits the set  $H$  of edges of priority  $h$ ; a region  $S_1$  disjoint from  $A_0$  from which every maximal path either visits an edge of priority  $h$  or does not visit an edge with priority  $h-1$ ; a region  $A_1$  that is the attractor of  $S_1$  in  $(G \setminus H) \setminus A_0$ ; and likewise regions  $S_i$  and  $A_i$ , for every  $i < \kappa$ , that are disjoint from  $A_\ell$  for every  $\ell < i$ , where  $A_i$  is the attractor of  $S_i$  in  $(G \setminus H) \setminus A_0$ , and  $S_i$  consists of vertices from which every maximal path visits priority  $h$ ,  $S_\ell$ , for some  $\ell < i$ , or does not visit a vertex with priority  $h-1$ . Each  $S_i$  is then further decomposed with width  $\kappa$  and with respect to priority  $h-2$ , which is larger or equal to all priorities in  $S_i$ . If such a decomposition is a full partition of  $G$ , in the sense that the vertices in  $G$  are the union of the vertices in the  $A_i$  regions, then the graph is even: indeed, any infinite path must either eventually remain within some  $S_i$ , thus avoiding seeing  $h-1$  infinitely often, or reach  $h$  infinitely often; within each  $S_i$ , the same is true with respect to  $h-2$ , and so on.

► **Definition 2** (Attractor decomposition). *Given a parity graph  $G = (V, E, L)$  with maximal priority at most some even  $h$ , and  $\kappa$  an ordinal, a  $(h\text{-height}, \kappa\text{-width})$ -attractor decomposition of  $G$ , if it exists, is recursively defined to be  $D = (H, A_0, (S_i, A_i, D_i)_{0 < i < \ell})$  for  $\ell \leq \kappa$ , where:*

- $H \subseteq E$  is the set of edges in  $G$  of priority  $h$ ,
- $A_0 = \text{Attr}(H, G)$ ,
- For every  $0 < i < \ell$ , we define  $V_i = V \setminus \bigcup_{0 \leq j < i} A_j$ , and  $G_i = (G \setminus H)|V_i$  and have:
  - $S_i \subseteq V_i$  is non-empty, such that  $G|S_i$  contains no edges of priority  $h-1$  and is closed under successors in  $G_i$ .
  - $A_i$  is  $\text{Attr}(S_i, (G_i))$ ,
  - $D_i$  is a  $((h-2)\text{-height}, \kappa\text{-width})$ -attractor decomposition of  $G_i|S_i$ ,
- $V = \bigcup_{0 \leq i < \ell} A_i$ .
- A  $(0\text{-height}, \kappa\text{-width})$ - and a  $(h\text{-height}, 0\text{-width})$ -attractor decomposition is just  $(H, V)$ : the entire graph is in the attractor of the edges of highest priority.

An attractor decomposition is *greedy* if at each level, each  $S_i$  is maximal, that is, it contains all vertices of  $V_i$  that do not have a path in  $G_i$  to a vertex of priority  $h-1$  in  $V_i$ .

The next lemma (proof in Section A) says that paths in a parity graph progress monotonously through the attractor decomposition, unless the maximal even priority occurs.

► **Lemma 3.** *Given a graph  $G$  that admits an attractor decomposition  $(H, A_0, (S_i, A_i, D_i)_{0 < i < \kappa})$ , the set  $A_j$  is unreachable from  $A_i$  in  $(G \setminus H)$  for all  $i$  and  $j$  such that  $0 \leq i < j \leq \kappa$ .*

► **Proposition 4.** *A parity graph is even if and only if it admits an attractor decomposition. The attractor decomposition can be chosen to be greedy.*

**Proof.** If a parity graph  $G$ , of maximal priority at most some even  $h$ , is even, we can construct a greedy attractor decomposition recursively for it as follows. Let  $H$  be the set of edges of priority  $h$  and  $A_0$  the attractor of  $H$ .

Then, we define each  $S_i$  and  $A_i$  for  $i > 0$  inductively. First let  $V_i = V \setminus \bigcup_{\ell < i} A_\ell$  for  $i$  an ordinal.  $V_i$  is either empty, or  $G_i = (G \setminus H)|V_i$  is an even parity graph with maximal priority no larger than  $h-1$ . Let  $S_i$  consist of all positions of  $G_i$  from where  $h-1$  can not be reached. If  $V_i$  is non-empty, there must be such positions, since otherwise one could build a path which sees infinitely many  $h-1$ , contradicting that  $G_i$  is an even graph. Let  $A_i$  be the attractor of  $S_i$  in  $G_i$  and let  $D_i$  be an  $h-2$ -attractor decomposition of  $S_i$ .

Let us assume  $V_i$  is non-empty for all countable ordinals  $i$ . Since all the  $V_i$ s are disjoint, their union is uncountable, since there are uncountably many ordinals smaller than  $\omega_1$ . However we only work with countable graphs, which gives a contradiction. Thus  $V_i$  must be empty for some countable ordinal  $i$ . Hence  $V = \bigcup_{\ell < i} A_\ell$ .

Observe that this attractor decomposition is, by construction, greedy.

For the other direction, assume that a parity graph has an attractor decomposition  $(T, A_0, (S_i, A_i, D_i)_{0 < i < \kappa})$  of height  $h$ . We proceed by induction on the height of the attractor decomposition, that is, the number of priorities in  $G$ . The base case of the unique priority 0 is trivial since all paths are even.

For the induction step, we observe that any infinite path of  $G$  must either eventually remain in some  $(G \setminus H) \upharpoonright S_\ell$  for  $1 \leq \ell < \kappa$  or reach  $H$  infinitely often, due to Lemma 3. In the latter case, the path is even since all edges in  $H$  are of the highest possible priority. In the former case, since each  $S_\ell$  has an attractor decomposition of height  $h-2$ , by the induction hypothesis, we are done.  $\blacktriangleleft$

### 3.2 Patterns and rank

Here we define the central notions of the article: patterns and rank, with which we measure the complexity of graphs.

► **Definition 5.** For an ordinal  $\kappa$ , a  $\kappa$ -pattern in a graph  $(V, E)$  consists of a family  $(V_\alpha)_{\alpha < \kappa}$  of pairwise disjoint non-empty subsets of  $V$  such that for all  $\alpha \leq \beta < \kappa$ :

For all vertices  $v \in V_\beta$  there exists a vertex  $v' \in V_\alpha$ , and a path  $u = v w v' \in V_\beta V^* V_\alpha$  which can only decrease in component index, meaning such that for all  $i < |u| - 1$  there exist  $\gamma \leq \delta < \kappa$ , such that  $u[i] \in V_\delta$  and  $u[i + 1] \in V_\gamma$ .

Basically, the condition of a pattern states that from any vertex in the pattern, one can go to some vertex in any component of lower index, following a (non-empty) index decreasing path. In particular any vertex must have at least one successor in the same component. Observe that a pattern in a subgraph is a pattern in the whole graph.

► **Definition 6.** A graph has rank  $\kappa$  if it has a  $\kappa$ -pattern but no  $\kappa+1$ -pattern. A graph class  $\mathcal{C}$  has rank  $\kappa$  if  $\kappa$  is the supremum rank of graphs in  $\mathcal{C}$ .

► **Example 7.** A graph has rank 0 if and only if it has no infinite paths. The class of trees with a finite number of infinite branches has no 2-pattern. It thus has rank 1. The class of directed acyclic graphs (DAGs) of finite width, where the width is the length of the longest antichain, has  $k$ -patterns for all finite  $k$  and therefore has infinite rank, while the class of DAGs of bounded width  $k$  is of rank  $k$ .

We can now make a connection between the rank and the decomposition of a graph:

► **Lemma 8.** An even parity graph of finite rank  $k$  has a  $k+1$ -attractor decomposition.

**Proof.** Let  $G$  be an even parity graph of maximal priority an even  $h$  or an odd  $h-1$  with vertices  $V$ . Since  $G$  is an even parity graph of maximal priority at most  $h$ , it has by Proposition 4 an  $h$ -height attractor decomposition. Let  $D$  be a greedy such decomposition of width  $\ell$  and not of any width  $\ell' < \ell$  (that is, the whole width  $\ell$  is used somewhere). We will argue that  $G$  has an  $\ell-1$ -pattern.

Since  $D$  indeed uses somewhere the whole width  $\ell$ , some induced subgraph has a decomposition  $D' = (T, A_0, (S_i, A_i, D_i)_{0 < i < \ell})$ .

We show by induction that for all finite  $i < \ell$ , the graph  $G' \upharpoonright (A_1 \cup \dots \cup A_i)$ , rooted at any vertex in  $A_i$ , has an  $i$ -pattern.  $G'$  therefore has an  $\ell-1$ -pattern if  $\ell$  is finite, and an  $i$ -pattern for all finite  $i$  otherwise.

First, we consider the base case  $i = 1$ . Observe that  $S_1$  has no terminal vertices as  $A_0$  is an attractor and  $G'$  has no terminal vertices. It follows that every vertex in  $S_1$  is on an infinite path. Then, as every vertex of  $A_1$  is in the attractor of  $S_1$ , from any vertex of  $A_1$  there is a path in  $A_1$  to an infinite path in  $S_1$ . It follows that in the graph  $G' \upharpoonright A_1$ , a 1-pattern is accessible from every position of  $A_1$ .

For the induction step, we similarly observe that as  $S_i$  has no terminal vertices, every vertex of  $S_i$  is on an infinite path in  $S_i$ . From greediness of  $D'$ , each vertex of  $S_i$  also has a path into  $A_{i-1}$  in  $G'$ ; indeed, a vertex without such a path could be in  $S_{i-1}$ , contradicting greediness. Then, a  $i-1$ -pattern in  $A_{i-1}$  is accessible infinitely often on every path in  $S_i$ ; it follows that there is an  $i$ -pattern from every position in  $S_i$ , and by extension in  $A_i$ . Thus a greedy  $\ell$ -width attractor decomposition witnesses that the rank of the graph is at least  $\ell-1$ .

Then, if an even parity graph is of rank  $k$ , it must have a greedy attractor decomposition, from Proposition 4, but this greedy attractor decomposition can have width at most  $k+1$ . ◀

### 3.3 Cantor Bendixson rank

In this subsection we introduce a well-known notion of descriptive set theory: the Cantor-Bendixson rank (see [10]). We then show that the Cantor-Bendixson rank of a tree is equal to its rank, as defined in the previous section.

The notion of Cantor-Bendixson rank and derivative is originally defined over topological spaces. As in [10]<sup>2</sup> we adapt the definition to trees so that it coincides with the usual notion for the metric space of  $\omega$ -branches where branches with a long common prefix are close.

► **Definition 9.** Let  $t = (V, E)$  be a tree, let  $u \in V$  be a vertex of  $t$ . We say that  $u$  is a path vertex of  $t$  if  $u$  lies on an infinite path of  $t$ , i.e. there is an infinite path  $up \in V^\omega$ . We say that  $u$  is a limit vertex of  $t$  if there is a converging sequence of pairwise distinct paths which go through  $u$ , i.e. there is a path  $up \in V^\omega$  such that for all  $i$ , there is a path  $up[0..i]q$  which is ultimately disjoint from  $p$ .

The trimmed subtree of  $t$ , denoted  $\text{trim}(t)$  is the subgraph of  $t$  induced by the set of path vertices. The derivative of  $t$ , denoted  $\text{der}(t)$  is the subgraph of  $t$  induced by the set of limit vertices.

Note that all limit vertices are path vertices, and both  $\text{trim}(t)$  and  $\text{der}(t)$  are upward closed subtrees of  $t$ , since any ancestor of a path (resp. limit) vertex is a path (resp. limit) vertex.

► **Definition 10.** We define inductively the ordinal derivatives of a tree:

- $\text{der}^0(t) = \text{trim}(t)$ ,
- $\text{der}^{\kappa+1}(t) = \text{der}(\text{der}^\kappa(t))$ , and
- $\text{der}^\kappa(t) = \text{trim}(\bigcap_{\alpha < \kappa} \text{der}^\alpha(t))$ , for  $\kappa$  a limit ordinal.

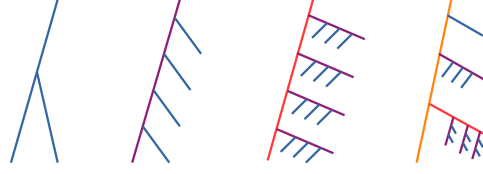
We say that  $t$  has Cantor-Bendixson rank (CB rank for short)  $\kappa$  if  $\kappa$  is the least ordinal such that  $\text{der}^\kappa(t)$  is empty.

Note that we can show inductively on ordinals that  $\text{der}^\kappa(t)$  is a decreasing (inclusion-wise) sequence of subtrees of  $t$ .

The Cantor-Bendixson rank of a tree happens to coincide with its rank in terms of patterns. The reason we consider patterns is because they are more flexible and apply to arbitrary graphs, while the CB-rank is not well-defined on other kinds of structures.

► **Proposition 11 (CB-rank).** Let  $\kappa$  be an ordinal. A tree has rank  $\kappa$  iff it has CB rank  $\kappa$ .

<sup>2</sup> Note that our definition is slightly different since we consider trees with possibly infinite branching.



■ **Figure 1** From left to right, trees of CB-rank 1 (blue), 2 (purple), 3 (red) and  $\omega+1$  (orange).

## 4 Over trees of finitely bounded Cantor-Bendixson rank

In this technical core we show that nondeterministic Büchi automata and alternating weak automata are as expressive as parity automata over trees of finitely bounded rank.

### 4.1 Bounding the rank

The main result of the subsection is the Product Lemma: the product of a tree of rank  $k$  with a graph with  $n$  vertices has rank at most  $kn$ . This is a key technical lemma for proving our main theorem in the next section as it will allow us to show that the acceptance parity game of an automaton over trees of bounded rank also has bounded rank.

► **Definition 12.** Let  $G$  be a graph, a  $k$ -pattern  $V = (V_0, \dots, V_{k-1})$  is called *tree-like* if the following holds for all  $i < j < k$ :

- there are no paths in  $G$  from  $V_i$  to  $V_j$ ,
- $V_i = \biguplus_{\ell \in \mathbb{N}} V_i^\ell$  where  $V_i^\ell$  is an infinite path, for  $\ell \in \mathbb{N}$ ,
- if a vertex can be reached in  $G$  from  $V_i^\ell$  and  $V_i^{\ell'}$  then  $\ell = \ell'$ .

If  $(V_1, \dots, V_{k-1})$  is tree-like, then  $V$  is called *almost tree-like*.

Tree-like patterns behave much like trees, which has two advantages: first, it is easier to extract a tree subgraph from the pattern and second, tree-like pattern can be *glued* together.

► **Lemma 13 (Glueing Lemma).** Let  $G$  be a graph and let  $V = (V_0, \dots, V_{k-1})$  be a tree-like  $k$ -pattern of  $G$ , with  $V_i = \biguplus_{\ell \in \mathbb{N}} V_i^\ell$  satisfying the corresponding properties. Let  $W^\ell = (W_0^\ell, \dots, W_{k'}^\ell)$  be tree-like  $k'+1$ -patterns of  $G$  such that  $W_{k'}^\ell = V_0^\ell$ , for all  $\ell \in \mathbb{N}$ . Then  $G$  admits a tree-like  $k+k'$ -pattern.

► **Proposition 14.** Let  $t$  be a tree, let  $n, k \in \mathbb{N}$ . If  $t \times K_n$  has a tree-like  $k$ -pattern, then  $t$  has a  $k$ -pattern.

**Proof.** Let  $t = (V, E)$ , and let  $\pi : V \times \{1, \dots, n\} \rightarrow V$  be the projection of vertices of  $t \times K_n$  to their  $t$  coordinate. Let us assume that  $(V_0, \dots, V_{k-1})$  is a tree-like pattern of  $t \times K_n$ . We show that two distinct vertices of the pattern must have different  $t$  coordinates. This is enough to ensure that  $(\pi(V_0), \dots, \pi(V_{k-1}))$  is a  $k$ -pattern of  $t$ . We first show that two vertices in different components of the pattern have different  $t$  coordinates, then we show that it also holds for two vertices in the same component of the pattern.

Two vertices that have the same  $t$  coordinate can reach the same vertices in  $t \times K_n$ . Then, two vertices in different components of the pattern cannot have the same  $t$  coordinate, since the pattern is tree-like.

Let  $i \in \{0, \dots, k-1\}$ , let  $V_i = \biguplus_{\ell \in \mathbb{N}} V_i^\ell$  such that the  $V_i^\ell$ s are paths with disjoint reachability sets. If two distinct vertices have the same  $t$  coordinate, they have to lie in the same  $V_i^\ell$ , for some  $\ell \in \mathbb{N}$ . However given two vertices of  $V_i^\ell$ , one can reach the other as  $V_i^\ell$  is a path, which means that the vertices have different  $t$  coordinate. ◀

## 164:10 When Is the Parity Index Hierarchy Finite?

► **Definition 15.** Given a graph  $G = (V, E)$ , a path  $p \in V^\omega$  in  $G$  and a set  $X \subseteq V$  of vertices of  $G$ , we say that  $p$  goes near  $X$  infinitely often if for all  $i \geq 0$ , there is  $j > i$  and  $x \in X$  such that  $(p[j], x) \in E$ .

► **Proposition 16.** Let  $t$  be a tree, let  $n \in \mathbb{N}$ . Let  $(V_0, \dots, V_{n+1})$  be an  $n+2$ -pattern of  $t \times K_n$  such that  $V_{n+1}$  is an infinite path. Then  $t \times K_n$  has an almost tree-like 3-pattern  $(V_0, W_1, V_{n+1})$

**Proof.** We have that  $V_{n+1}$  is an infinite path  $p$  in  $t \times K_n$ . We construct a sequence  $W_1^0, W_1^1, \dots$  and define  $W_1 = \bigcup_i W_1^i$ . We will show that we can construct this sequence inductively with the following properties for  $i, j \geq 0$ :

- there is a path in  $t \times K_n$  from  $p[i]$  to any vertex of  $W_1^i$ ,
- $W_1^i$  is an infinite path which goes near  $V_0$  infinitely often,
- if  $v$  is reachable in  $t \times K_n$  from  $W_1^i$  and  $W_1^j$  then  $i = j$
- there is no path in  $t \times K_n$  from any vertex of  $W_1^i$  to  $V_{n+1}$ .

The first condition ensures that  $(W_1, V_{n+1})$  is a pattern and the second that  $(V_0, W_1)$  is a pattern. The last two conditions ensure that  $(W_1, V_{n+1})$  is tree-like. Thus we only have left to construct the sequence  $W_1^0, W_1^1, \dots$

Initial case: Let  $w \in V_{n+1}$  and let  $wq_{n+1}w_nq_n \dots q_1w_1$  be a path such that  $q_i \in V_i^*$ ,  $w_i \in V_i$  for  $i \in [1, n+1]$ . Let us consider infinite paths  $w_i r_i \in V_i^\omega$  for  $i \in [1, n]$ . Then, the projections  $\pi(p), \pi(r_1), \dots, \pi(r_n)$  on  $t$  of these paths cannot all be eventually equal since the  $V_i$ s are disjoint and there are only  $n$  copies of each vertex of  $t$  in  $t \times K_n$ . Let  $i$  be an index such that  $\pi(r_i)$  and  $\pi(p)$  are eventually disjoint, meaning they end up in different branches of  $t$ . Let  $u$  be the first vertex of  $wq_{n+1} \dots q_{i+1}w_i r_i$  such that  $\pi(u)$  is not on the branch  $\pi(p)$ . Note that there is an edge from a vertex of  $p$  to  $u$  but there is no path from  $u$  to any vertex of  $p$ . From  $u$  we choose a path  $uq$  in  $V_n^* \dots V_2^* V_1^\omega$  which goes near  $V_0$  infinitely often, which is possible by the definition of pattern. The set  $W_1^0$  is defined as the set of vertices of  $uq$ .

Assume we have constructed  $W_1^0, \dots, W_1^i$ , with the desired properties. We choose a vertex  $w \in V_{n+1}$  such that  $w$  cannot reach any of the  $W_1^j$ s, for  $j \in [0, i]$ . Such a  $w$  has to exist, since the projections of the  $W_1^j$ s are ultimately disjoint from  $\pi(p)$ .

Similarly to the initial case, we can find a path  $uq \in V_n^* \dots V_1^\omega$  which goes near  $V_0$  infinitely often, such that there is a path in  $V_{n+1}$  from  $w$  to a predecessor of  $u$ , and there is no path from  $u$  to a vertex of  $V_{n+1}$ . Let  $W_1^{i+1}$  be the set of vertices of  $uq$ . Since  $w$  cannot reach  $W_1^j$ s, the vertices of  $W_1^{i+1}$  cannot reach  $W_1^j$ s, *a fortiori*. It remains to show that there is no path from  $W_1^j$  to  $W_1^{i+1}$  for  $j \in [0, i]$ . Assume the contrary, then there is a vertex  $v$  in  $W_1^{i+1}$  and path from  $p[0]$  which does not go through  $w$  to  $v$ , by going through  $W_1^j$ . Furthermore, there is also a path which goes through  $w$ . Let  $w' \in V_{n+1} \cup \bigcup_{0 \leq j \leq i+1} W_1^j$  be a vertex distinct from  $w$ . If  $w' \in V_{n+1}$ , then  $\pi(w) \neq \pi(w')$ , since  $V_{n+1}$  is a path. If  $w' \notin V_{n+1}$ , since from the induction assumption there is no path from  $\bigcup_{0 \leq j \leq i+1} W_1^j$  to  $V_{n+1}$ , we have  $\pi(w) \neq \pi(w')$ . Then there are two distinct paths in  $t$  from the same vertex to  $\pi(v)$ , a contradiction. ◀

From a graph  $G$  of rank  $k$ , the product  $G \times K_n$  with the clique over  $n$  vertices, has rank at least  $kn$ . We show below that this is optimal for trees, and leave open the question whether this holds in general.

► **Lemma 17 (Product Lemma).** Let  $t$  be a tree of rank  $k \in \mathbb{N}$ , let  $n \in \mathbb{N}$ . The graph  $t \times K_n$  has rank at most  $kn$ .

**Proof.** The statement is trivially true for  $n \leq 1$ , so assume  $n \geq 2$  in the following.

We show by induction on  $k$  that given  $(V_0, \dots, V_{kn})$  a  $kn+1$ -pattern of  $t \times K_n$ , such that  $V_{kn}$  is a path, one can obtain a tree-like  $k+1$  pattern  $(W_0, \dots, V_{kn})$ . From Proposition 14, this is enough to obtain the result.

Clearly this holds for  $k = 0$ . Let us assume it holds for some  $k \geq 0$ .

Let  $(V_0, \dots, V_{(k+1)n})$  be a  $(k+1)n+1$ -pattern of  $t \times K_n$ . We use Proposition 16 on the pattern  $(V_{kn-1}, \dots, V_{(k+1)n})$  and obtain  $(V_{kn-1}, W_2, V_{(k+1)n})$ , an almost tree-like pattern. We have that  $W_2$  is a union of disjoint paths  $W_2^i$ ,  $i \geq 0$ , with the property that no vertex can be reached from two distinct paths. For each  $i \in \mathbb{N}$ ,  $(V_0, \dots, V_{kn-1}, W_2^i)$  is a  $kn+1$ -pattern with  $W_2^i$  an infinite path. From the induction assumption, we can obtain a tree-like  $k$ -pattern  $(W_0^i, \dots, W_{k-1}^i, W_2^i)$ . Since  $(W_2, V_{(k+1)n})$  is tree-like, from the Glueing Lemma we obtain a tree-like  $k+1$  pattern, which concludes the induction.

To finish the proof, we only need to show that given  $(V_0, \dots, V_{kn})$  a  $kn+1$ -pattern of  $t \times K_n$  we can assume without loss of generality that  $V_{kn}$  is a path. To do this we only need to restrict  $V_{kn}$  to a path which goes near  $V_{kn-1}$  infinitely often, which can easily be done. ◀

Now that we have proved the Product Lemma, we need another simpler lemma dealing with the rank of trees obtained by what we call subdivision of vertices.

► **Definition 18.** Let  $G = (V, E)$  be a graph. An  $n$ -subdivision of vertex  $u$  in  $G$  is the graph  $G' = (V', E')$  where  $V' = V \uplus \{u_1, \dots, u_n\}$ ,  $E' = (E \cup \{(u_1, u_2), \dots, (u_{n-1}, u_n), (u_n, u)\} \cup \{(s, u_1) \mid (s, u) \in E\}) \setminus \{(s, u) \in E\}$ .

A graph  $G'$  is called an  $n$ -subdivision of  $G$  if  $G'$  can be obtained by applying to all vertices  $u$  an  $n_u$ -subdivision with  $n_u \leq n$  (a 0-subdivision of a vertex does not modify the graph).

► **Proposition 19.** Let  $t$  be a tree of rank  $k$  and let  $t'$  be an  $n$ -subdivision of  $t$ . Then  $t'$  has rank at most  $k$ .

## 4.2 From parity automata to Büchi and weak automata

The previous technical developments, in particular the Product Lemma, imply that the acceptance game of a tree cannot have a much larger rank than the original tree. This implies in turn a bound on the attractor decompositions of such games for trees of bounded CB-rank. We use this to convert parity automata into a Büchi automata over trees of bounded rank.

► **Corollary 20.** Given a tree  $t$  with rank  $k$  and an APT  $\mathcal{A}$ , the acceptance parity game  $\mathcal{G}(t, \mathcal{A})$  has rank at most  $kc$  for some constant  $c$  that depends only on  $\mathcal{A}$ .

We now build an NBT  $\mathcal{B}_{p,k}$  for finite  $p, k$  that operates over the tree unfolding of parity graphs with priorities up to  $p$ , accepts those with a  $k$ -width attractor decomposition, and rejects odd ones<sup>3</sup>. We build  $\mathcal{B}_{p,k}$  for even  $p$ , by induction on  $p$ , starting with  $\mathcal{B}_{0,k}$ , which, for all  $k$  simply consists of an accepting state  $q$  and a transition  $\delta(q, 0) = \square q$ .

Then, we build  $\mathcal{B}_{p+2,k}$  as follows. We take  $k$  copies  $\mathcal{B}_k, \dots, \mathcal{B}_1$  of  $\mathcal{B}_{p,k}$  with initial states  $\iota_k, \dots, \iota_1$ , respectively, and add a non-accepting state  $q_i$  before each  $\mathcal{B}_i$ , as well as  $q_0$  after  $\mathcal{B}_1$ , each with a nondeterministic choice over priorities up to  $p+1$  to either loop or move to the next copy  $\mathcal{B}_{i-1}$  of  $\mathcal{B}_{p,k}$ . From  $q_0$ , there is only a self-loop on priorities up to  $p+1$ . From every state of each  $\mathcal{B}_i$ , seeing the input letter  $p+1$  leads to the intermediate state  $q_{i-1}$ . Finally, we

<sup>3</sup> Note that this construction is similar in spirit to the one in [4, Lemma 9], which is made succinct by the use of small universal trees.

## 164:12 When Is the Parity Index Hierarchy Finite?

also add one accepting state  $q_b$ , to which all states transition upon seeing  $p+2$ , and from which there is a transition over all priorities up to  $p+2$  to  $q_k$ , the first new non-accepting state. A more formal definition can be found in Section B.

Finally, let  $\mathcal{B}_{p,k}$  for odd  $p$  be as  $\mathcal{B}_{p+1,k}$ , except without any transitions on priority  $p+1$ .

► **Remark 21.**  $\mathcal{B}_{1,k}$  is a weak automaton since the only accepting cycles are self-loops in the copies of  $\mathcal{B}_{0,k}$  and the only rejecting cycles are self-loops on the intermediate states  $q_i$ .

► **Lemma 22.** *For every  $k \in \mathbb{N}$  and  $p \in \mathbb{N}$ , the automaton  $\mathcal{B}_{p,k}$  accepts parity graphs with priorities up to  $p$ , represented by their tree unfolding, that have a  $k$ -width attractor decomposition and rejects odd graphs.*

**Proof.** We prove the statement for even  $p$  by induction on  $p$ ; the statement for odd  $p$  follows easily. In the base case, where  $p = 0$ , for every  $k \in \mathbb{N}$ , all parity graphs have a  $k$ -width attractor decomposition, and indeed by the construction of  $\mathcal{B}_{0,k}$ , it accepts everything.

In the induction step, we assume the claim for  $p$ , and prove it for  $p+2$ . If a parity graph  $G$  with priorities up to  $p+2$  has a  $k$ -width attractor decomposition  $(T, A_0, (S_i, A_i, D_i)_{0 < i < j})$ , then Eve can use the following strategy in the acceptance game  $\mathcal{G}(G, \mathcal{B}_{p+2,k})$ : At a position  $(v, q_i)$  with  $v \in A_0 \cup \dots \cup A_i$ , Eve remains in  $q_i$  until she reaches a position  $(v', q_i)$  where  $v' \in S_\ell$  for some  $\ell \leq i$ ; from there, she moves to  $\iota_i$  and follows a strategy that is winning in the acceptance game of  $\mathcal{B}_\ell$  and the tree rooted at  $v'$  and restricted to priorities up to  $p$ , from the IH. Then, if no priority equal or larger than  $p+1$  occurs, the play will remain within  $\mathcal{B}_\ell$ , and is winning by IH. If priority  $p+2$  occurs, the game moves to some  $(v'', q_b)$ , of priority 2; if  $p+1$  occurs, the game moves to some  $(v'', q_{i-1})$  where  $v'' \in A_0 \cup \dots \cup A_{i-1}$ . This is the case since  $S_\ell$  does not contain priority  $p+1$  and any path from  $S_\ell$  that does not contain  $p+2$  but contains  $p+1$  progresses to some  $A_j$  with  $j < \ell$  (Lemma 3). If  $v'' \in A_0$ , then the priority  $p+2$  will eventually occur, and the game moves to some  $(v'', q_b)$ , of priority 2.

We now argue that this strategy is winning for Eve. Indeed, a play can either eventually remain in some  $\mathcal{B}_\ell$  component, in which case it is winning by induction hypothesis, or eventually exits every  $\mathcal{B}_\ell$ . In the latter case, the only way for Eve to lose would be to get stuck in  $q_0$  and never see the priority  $p+2$ . However, since the  $G$ -component of the play moves to some  $A_j$  with  $j < \ell$  whenever it exists a component  $\mathcal{B}_\ell$ , the play can only reach  $(v, q_0)$  where  $v \in A_0$ , in which case  $p+2$  will eventually occur.

Furthermore, if Eve has a winning strategy in the acceptance game, then the highest priority on all infinite paths of  $G$  is even because all accepting cycles in  $\mathcal{B}_{p,k}$  read words of which the highest priority is even; hence  $\mathcal{B}_{p+2,k}$  does not accept any odd graphs.

The statement for odd  $p$  follows as  $\mathcal{B}_{p,k}$  is then as  $\mathcal{B}_{p+1,k}$  without any  $p+1$  transitions. ◀

We now build, from any APT  $\mathcal{A}$  of priorities up to  $p$  an ABT  $\mathcal{B}$  that agrees with  $\mathcal{A}$  over trees of rank at most  $k$  by composing it with  $\mathcal{B}_{p,k'}$  where  $k'$  only depends on  $k$  and  $\mathcal{A}$ .

► **Theorem 23.** *Given  $k \in \mathbb{N}$  and an alternating parity automaton  $\mathcal{A}$ , one can effectively construct an alternating Büchi automaton  $\mathcal{B}$  that agrees with  $\mathcal{A}$  over trees of Cantor-Bendixson rank up to  $k$ . If  $\mathcal{A}$  is nondeterministic, so is  $\mathcal{B}$  and if  $\mathcal{A}$  is coBüchi then  $\mathcal{B}$  is weak.*

**Proof.** Consider an APT  $\mathcal{A}$  with priorities up to  $p$ . Let  $k' = kc + 1$  for the constant  $c$  in Corollary 20; then we know that the game  $\mathcal{G}(t, \mathcal{A})$  has rank strictly less than  $k'$  for all trees  $t$  of rank up to  $k$ .

By Lemma 22, the ABT  $\mathcal{B}_{p,k'}$  accepts all parity graphs with priorities up to  $p$  with a  $k'$ -width attractor decomposition, and rejects odd parity graphs. By construction,  $\mathcal{B}_{p,k'}$  only has  $\Box$  modalities. Then, we build the required ABT  $\mathcal{B}$  by taking the synchronous composition

$\mathcal{A} \times \mathcal{B}_{p,k'}$  of  $\mathcal{A}$  and  $\mathcal{B}_{p,k'}$ . This is an ABT that accepts a tree  $t$  if and only if  $\mathcal{B}_{p,k'}$  accepts some tree induced by a strategy for Eve in  $\mathcal{G}(t, \mathcal{A})$ . The details of this composition are as expected, detailed in Section B.

In words,  $\mathcal{B}$  is an automaton that uses  $\mathcal{A}$  to process the input tree, and at each step,  $\mathcal{B}_{p,k'}$  processes the priority that  $\mathcal{A}$  would produce, which adds some nondeterministic choices. We argue  $\mathcal{B}$  accepts any tree  $t$  such that in  $\mathcal{G}(t, \mathcal{A})$  Eve has a winning strategy that induces a graph with a  $k'$ -width attractor decomposition, and rejects  $t \notin L(\mathcal{A})$ .

Assume  $t$  is such that in  $\mathcal{G}(t, \mathcal{A})$  Eve has a winning strategy  $\sigma$  that induces an even graph  $G_\sigma$  with a  $k'$ -width attractor decomposition. Let  $\sigma'$  be Eve's winning strategy in  $\mathcal{G}(G_\sigma, \mathcal{B}_{p,k'})$ . We can then compose these strategies to obtain a winning strategy in  $\mathcal{G}(t, \mathcal{B})$ . Indeed, positions of  $\mathcal{G}(t, \mathcal{B})$  can be seen as a position of  $\mathcal{G}(t, \mathcal{A})$  paired with states of  $\mathcal{B}_{p,k'}$ , to which are added, after the resolution of the boolean combinations in transition conditions of  $\mathcal{A}$ , the resolution of the boolean combinations in the transition condition of  $\mathcal{B}_{p,k'}$  corresponding to a transition on the priority of the next state in  $\mathcal{A}$ . In the composition strategy,  $\sigma$  determines the choices between successors in  $t$  at  $\Diamond$ -modalities and choices at disjunctions in the transition conditions of  $\mathcal{A}$ , while  $\sigma'$  determines choices at disjunctions in the transition conditions of  $\mathcal{B}_{p,k'}$ . The existence of  $\sigma$  guarantees the existence of  $\sigma'$ , and that the composition strategy remains in the restriction of  $\mathcal{G}(t, \mathcal{B})$  to  $\mathcal{G}_\sigma$  in the  $\mathcal{G}(t, \mathcal{A})$ -part of the game. Since the priorities of  $\mathcal{B}$  come from the priorities of  $\mathcal{B}_{p,k'}$ , the fact that  $\sigma'$  is winning for Eve ensures that the composition of  $\sigma$  and  $\sigma'$  is winning for Eve.

Conversely, if  $t \notin L(\mathcal{A})$ , then Adam has a winning strategy  $\tau$  in  $\mathcal{G}(t, \mathcal{A})$ . He can use  $\tau$  directly in  $\mathcal{G}(t, \mathcal{B})$  as  $\mathcal{B}$  does not add any universal choices ( $\Box$  or  $\wedge$ ) beyond those of  $\mathcal{A}$ . Let  $w$  be the sequence of priorities induced by the projection of a  $\tau$ -consistent play of  $\mathcal{G}(t, \mathcal{B})$  on the  $\mathcal{G}(t, \mathcal{A})$  component; the highest priority seen infinitely often on  $w$  is odd. The projection of the same play on  $\mathcal{B}_{p,k}$  is a path of  $\mathcal{B}_{p,k}$  that processes  $w$ . By construction, no accepting cycle of  $\mathcal{B}_{p,k}$  processes a word of which the highest maximal priority is odd, so this path is rejecting, and  $\tau$  is a winning strategy for Adam in  $\mathcal{G}(t, \mathcal{B})$ .

For all  $t$  of rank up to  $k$ ,  $\mathcal{G}(t, \mathcal{A})$  has rank strictly less than  $k'$ , if  $t \in L(\mathcal{A})$  then Eve has a winning strategy that induces an even graph that also has rank less than  $k'$ . From Lemma 8, it has a  $k'$ -width attractor decomposition, and therefore  $\mathcal{B}$  accepts  $t$ ; if  $t \notin L(\mathcal{A})$  then  $\mathcal{B}$  rejects  $t$ , so  $\mathcal{B}$  agrees with  $\mathcal{A}$  on such trees. By construction, if  $\mathcal{A}$  is nondeterministic, so is  $\mathcal{B}$  as  $\mathcal{B}_{p,k}$  only has nondeterministic choices; if  $\mathcal{A}$  is coBüchi, then  $\mathcal{B}$  is weak since  $\mathcal{B}_{1,k'}$  is weak. ◀

► **Corollary 24.** *Let  $k \in \mathbb{N}$ , let  $\mathcal{A}$  be an alternating parity automaton, one can effectively construct an alternating weak automaton  $\mathcal{B}$  that agrees with  $\mathcal{A}$  over trees of Cantor-Bendixson rank at most  $k$ .*

**Proof.** Given  $\mathcal{A}$ , from Theorem 23 there is an alternating Büchi automaton  $\mathcal{B}$  that agrees with  $\mathcal{A}$  over trees of Cantor-Bendixson rank at most  $k$ . Let  $\mathcal{C}$  be the standard syntactic coBüchi complement of  $\mathcal{B}$ , obtained by exchanging  $\wedge$  and  $\vee$ , and  $\Diamond$  and  $\Box$ , and replacing priority 1 with 0 and 2 with 1. Then  $\mathcal{C}$  accepts a tree of rank at most  $k$  if and only if  $\mathcal{A}$  rejects it. Again from Theorem 23, there is an alternating weak automaton  $\mathcal{C}'$  that agrees with  $\mathcal{C}$  on all trees of rank at most  $k$ . Then, the syntactic complement  $\mathcal{B}'$  of  $\mathcal{C}'$ , is an alternating weak automaton that agrees with  $\mathcal{A}$  on all trees of rank at most  $k$ . ◀

## 5 Over trees of countable Cantor-Bendixson rank

► **Theorem 25.** *Given an alternating parity automaton  $\mathcal{A}$ , one can effectively construct an alternating  $\{1, 2, 3\}$ -automaton  $\mathcal{C}$  that agrees with  $\mathcal{A}$  over trees of countable Cantor-Bendixson rank. Furthermore, if  $\mathcal{A}$  is nondeterministic, so is  $\mathcal{C}$ .*

**Proof.** Recall from the proof of Theorem 23 that the synchronous product  $\mathcal{A} \times \mathcal{B}_{p,k}$ , for some  $k$  dependent on  $\mathcal{A}$ , is equivalent to  $\mathcal{A}$  over trees of rank 1 for  $\mathcal{B}_{p,k}$ , the nondeterministic Büchi automaton from Lemma 22. For clarity, we will describe  $\mathcal{B}$  as having its additional 3-priority on transitions.

Let  $\mathcal{B}$  be as  $\mathcal{B}_{p,k}$ , but with additional nondeterministic transitions of priority 3 from every state to the initial state  $q_l$ . (To have priorities on states, it suffices to use a copy of  $q_l$  of priority 3.) We now argue that  $\mathcal{A} \times \mathcal{B}$  is equivalent to  $\mathcal{A}$  over thin trees.

One direction is clear: all accepting cycles of  $\mathcal{B}$ , like of  $\mathcal{B}_{p,k}$  read words of which the highest priority is even, so for  $\mathcal{A} \times \mathcal{B}$  to accept  $t$ , there must be an accepting run of  $\mathcal{A}$  over  $t$ .

For the other direction, we show that if a thin tree  $t$  is accepted by  $\mathcal{A}$ , then it is also accepted by  $\mathcal{A} \times \mathcal{B}$ . We build a run of  $\mathcal{A} \times \mathcal{B}$  over  $t$  with an accepting run of  $\mathcal{A}$  over  $t$  in the first component. For the second component, we must resolve the non-determinism in  $\mathcal{B}$ . To do so, let  $T_\alpha$  be the set of vertices of  $t$  of CB-rank  $\alpha$ . Observe that any subtree of  $t$  induced by  $T_\alpha$  for some  $\alpha$  itself is a tree of rank 1. Then, we resolve the nondeterminism of  $\mathcal{B}$  as follows: when the position of the run in  $t$  changes CB-rank, the run takes the available 3-edge that resets  $\mathcal{B}$  to its initial state.

Since the rank along a path can only decrease, the rank along a path in  $t$  of countable rank can only change a finite number of times, hence this strategy will only use the 3-transition a finite number of times on a tree of countable rank.

Then, from some  $(q, q_l)$  and position  $v$  in  $T_\alpha$  for some  $\alpha$  and while the run remains in  $T_\alpha$ , the run follows an accepting run in  $\mathcal{A} \times \mathcal{B}_{p,k}$  from  $(q, q_l)$  over the subtree induced by  $v$  in  $T_\alpha$ . Such a run exists since  $\mathcal{A} \times \mathcal{B}_{p,k}$  from  $(q, q_l)$  agrees with  $\mathcal{A}$  from  $q$  over trees of rank 0. ◀

► **Corollary 26.** *Given an alternating parity automaton  $\mathcal{A}$ , one can construct an alternating  $\{0, 1, 2\}$ -automaton that agrees with  $\mathcal{A}$  over thin trees.*

**Proof.** Given an alternating parity automaton  $\mathcal{A}$ , let  $\mathcal{C}$  be an alternating  $\{1, 2, 3\}$ -automaton that agrees with the complement of  $\mathcal{A}$  over trees of countable Cantor-Bendixson rank, obtained from Theorem 25. The syntactic complement of  $\mathcal{C}$ , obtained by exchanging  $\diamond$  and  $\square$ ,  $\wedge$  and  $\vee$ , and shifting priorities by  $-1$ , is a  $\{0, 1, 2\}$ , then agrees with  $\mathcal{A}$  over thin trees. ◀

► **Remark 27.** While we believe that this section is helpful in understanding how our techniques extend beyond finite Cantor-Bendixson rank, analogous results on *ordered* thin trees were already shown in [7]. While some of our reasoning is similar, there are also some differences that make transferring results between the two settings non-trivial. For instance, our trees are of unbounded, potentially infinite arity. Furthermore, they are interested in recognising languages of thin trees *among all forests*, that is, in automata that can both check whether a tree is thin and in the language. Crucially, over unordered trees, our automata (which are bisimulation-invariant, like the  $\mu$ -calculus) do not suffice to check that the rank of a tree is finite or countable, while their automata have the expressivity of MSO.

To exemplify how the two setting differ with respect to the parity index hierarchy, we observe that nondeterministic Büchi automata are as expressive as parity automata on the class of ordered trees in which every branch goes left a finite number of times. It is harder to find a similar example of a class of *unordered* trees of unbounded Cantor-Bendixson rank over which non-deterministic Büchi automata are as expressive as parity automata.

## 6 Over trees of finite Cantor-Bendixson rank

In this section we argue that the previously described trichotomy result cannot be improved upon: two priorities do not capture all regular tree languages over trees of finite rank.

► **Theorem 28.** *The language of trees where letter  $a$  occurs a finite number of times on every branch is not recognisable by alternating Büchi automata over the class of trees of finite rank.*

Before we proceed with the proof of Theorem 28, we prove the following technical lemma that describes how adding a back-edge in a tree and unfolding it back into a tree only increases the rank of the tree by at most 1.

► **Definition 29** (back-edge unfolding). *Let  $t = (V, E)$  be a tree, let  $u, v \in V$  such that  $u$  is an ancestor of  $v$ . We define  $t'$  the tree obtain by unfolding the graph  $G = (V, E \cup \{(v, u)\})$ . We say that  $t'$  is a back-edge unfolding of  $t$ .*

► **Lemma 30.** *Given a tree  $t$  of rank  $\alpha$ , the tree  $t'$ , obtained as a back-edge unfolding of  $t$ , is of rank at most  $\alpha+1$ .*

**Proof.** In the tree  $t'$ , all vertices except those on the branch with the infinitely repeating segment from  $u$  to  $v$  have induced subtrees that appear in  $t$ , and therefore all those subtrees are of rank at most  $\alpha$ . The  $\alpha$ th derivative of  $t'$  is thus either empty or contains this single infinite branch. Hence  $t'$  has rank at most  $\alpha+1$ . ◀

► **Proposition 31.** *Let  $\mathcal{C}$  be a non-empty class of trees closed under back-edge unfolding. Then the language of trees in which the letter  $a$  occurs a finite number of times on every branch is not recognisable by an alternating Büchi automata over the class  $\mathcal{C}$ .*

**Proof.** Assume, towards a contradiction, that there is an alternating automaton  $\mathcal{A}$  that recognises the language  $L$  of trees in which  $a$  occurs a finite number of times on every branch over the class of trees of finite rank.

Consider  $t$  a finitely branching tree of rank  $n$  strictly larger than  $2^m$  where  $m$  is the size of  $(Q \cup \mathbb{T}(Q)) \times (\Sigma \cup \{\varepsilon\}) + 2$  (the size of the  $\mathcal{A}$ -dependent component of the acceptance game of  $\mathcal{A}$ ), in which edges between vertices of different rank are labelled  $a$  and other edges are labelled  $b$ . This tree is in  $L$  so there is a positional winning strategy  $\sigma$  for Even in  $\mathcal{G}(t, \mathcal{A})$ . We now show that there is a branch in  $t$  with  $n-1$  occurrences of  $a$  such that every play that agrees with  $\sigma$  sees a Büchi state between each occurrence. Indeed, first take a branch  $\pi_n$  of constant rank  $n$ , starting from the root, which we call  $v_n$ . All plays along this branch that agree with the strategy  $\sigma$  must eventually see a Büchi state, so by König's lemma at some vertex, all such plays have seen a Büchi state. Since this vertex is of finite rank  $n$ , it has some descendant  $v_{n-1}$  of rank  $n-1$ . We build  $\pi_{i-1}$  and  $v_{i-2}$  similarly from  $v_{i-1}$ . We call  $\pi$  the branch from the root that includes each  $v_i$  and eventually follows  $\pi_1$ . Since  $\pi$  changes rank  $n-1$  times, it has  $n-1$  occurrences of  $a$ , and, by construction, every play that agrees with  $\sigma$  sees a Büchi state between two such occurrences.

By pigeonhole principle, along  $\pi$ , we can find two vertices  $v$  and  $v'$  such that:

- $a$  occurs at least once in between  $v$  and  $v'$ ;
- the edge leading to  $v$  and  $v'$  has the same label;
- $\sigma$  visits the same set of states of  $\mathcal{A}$  in both  $v$  and  $v'$ ;
- Every play that agrees with  $\sigma$  visits a Büchi state between  $v$  and  $v'$ .

The tree  $t'$ , defined as the unfolding of  $t$  with a back-edge from the parent of  $v'$  to  $v$ , has finite rank from Lemma 30, so  $\mathcal{A}$  should reject  $t'$  since it has a branch with infinitely many occurrences of  $a$ . However the strategy  $\sigma'$ , induced by  $\sigma$  on  $t'$ , is an winning strategy in  $\mathcal{G}(t', \mathcal{A})$  since plays on branches that see a finite number of times copies of  $v$ , eventually resemble plays from  $\sigma$  in  $\mathcal{G}(t, \mathcal{A})$  and are therefore winning, and plays on the unique branch that sees copies of  $v$  infinitely often visit a Büchi state in between each such occurrence and are therefore accepting. ◀

**Proof of Theorem 28.** The result is a consequence of Proposition 31. The fact that trees with finite rank are closed under back-edge unfolding is given by Proposition 30. ◀

## 7 Discussion

We have shown that over trees, there is a tight relationship between the index hierarchy of parity automata and the Cantor-Bendixson rank. While on trees we observe a trichotomy, it is open whether a similar result holds over graphs, or whether there are classes of graphs over which the parity index hierarchy collapses to each level beyond  $[1, 2, 3]$ . We expect our notion of graph rank to be helpful in exploring this question. One of the technical challenges of extending this work to graphs is generalising the Product Lemma, whose proof relies heavily on the simplicity of tree structures. We hope that studying the orthogonal cases of finite graphs and DAGs separately will help tame the challenges of the graph setting.

---

## References

- 1 André Arnold. p329 the  $\mu$ -calculus alternation-depth hierarchy is strict on binary trees. *RAIRO Theor. Informatics Appl.*, 33(4/5):329–340, 1999. doi:10.1051/ITA:1999121.
- 2 Julian C. Bradfield. The modal  $\mu$ -calculus alternation hierarchy is strict. *Theor. Comput. Sci.*, 195(2):133–153, 1998. doi:10.1016/S0304-3975(97)00217-X.
- 3 Giovanna D’Agostino and Giacomo Lenzi. The  $\mu$ -calculus alternation depth hierarchy is infinite over finite planar graphs. *Theor. Comput. Sci.*, 737:40–61, 2018. doi:10.1016/J.TCS.2018.04.009.
- 4 Laure Daviaud, Marcin Jurdzinski, and Karoliina Lehtinen. Alternating Weak Automata from Universal Trees. In *30th International Conference on Concurrency Theory (CONCUR 2019)*, 2019. doi:10.4230/LIPIcs.CONCUR.2019.18.
- 5 Laure Daviaud, Marcin Jurdziński, and K. S. Thejaswini. The Strahler Number of a Parity Game. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, 2020. doi:10.4230/LIPIcs.ICALP.2020.123.
- 6 Alessandro Facchini, Filip Murlak, and Michal Skrzypczak. Index problems for game automata. *CoRR*, abs/1506.02153, 2015. arXiv:1506.02153.
- 7 Tomasz Idziaszek, Michal Skrzypczak, and Mikolaj Bojanczyk. Regular languages of thin trees. *Theory Comput. Syst.*, 58(4):614–663, 2016. doi:10.1007/S00224-014-9595-Z.
- 8 David Janin and Igor Walukiewicz. Automata for the modal mu-calculus and related results. In Jirí Wiedermann and Petr Hájek, editors, *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS’95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, volume 969 of *Lecture Notes in Computer Science*, pages 552–562. Springer, 1995. doi:10.1007/3-540-60246-1\_160.
- 9 Marcin Jurdzinski, Rémi Morvan, and K. S. Thejaswini. Universal algorithms for parity games and nested fixpoints. In Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Rupak Majumdar, editors, *Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday*, volume 13660 of *Lecture Notes in Computer Science*, pages 252–271. Springer, 2022. doi:10.1007/978-3-031-22337-2\_12.
- 10 A. Kechris. *Classical Descriptive Set Theory*. Graduate Texts in Mathematics. Springer New York, 1995. URL: <https://books.google.fr/books?id=pPv9KCEkkl5C>.
- 11 Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS1997, Ramat-Gan, Israel, June 17-19, 1997, Proceedings*, pages 147–158. IEEE Computer Society, 1997. doi:10.1109/ISTCS.1997.595167.

- 12 Giacomo Lenzi. A hierarchy theorem for the  $\mu$ -calculus. In Friedhelm Meyer auf der Heide and Burkhard Monien, editors, *Automata, Languages and Programming, 23rd International Colloquium, ICALP96, Paderborn, Germany, 8-12 July 1996, Proceedings*, volume 1099 of *Lecture Notes in Computer Science*, pages 87–97. Springer, 1996. doi:10.1007/3-540-61440-0\_119.
- 13 Damian Niwinski and Michal Skrzypczak. On guidable index of tree automata. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPICs*, pages 81:1–81:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.MFCS.2021.81.
- 14 Damian Niwinski and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. In Ruy J. G. B. de Queiroz and Patrick Cégielski, editors, *Proceedings of the 11th Workshop on Logic, Language, Information and Computation, WoLLIC 2004, Fontainebleau, France, July 19-22, 2004*, volume 123 of *Electronic Notes in Theoretical Computer Science*, pages 195–208. Elsevier, 2004. doi:10.1016/J.ENTCS.2004.05.015.
- 15 Thomas Wilke. Alternating tree automata, parity games, and modal  $\{\mu\}$ -calculus. *Bulletin of the Belgian Mathematical Society-Simon Stevin*, 8(2):359–391, 2001.
- 16 Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.

## A

 Proofs of Section 3

► **Lemma 3.** *Given a graph  $G$  that admits an attractor decomposition  $(H, A_0, (S_i, A_i, D_i)_{0 < i < \kappa})$ , the set  $A_j$  is unreachable from  $A_i$  in  $(G \setminus H)$  for all  $i$  and  $j$  such that  $0 \leq i < j \leq \kappa$ .*

**Proof.** We proceed by transfinite induction on  $i$ . For  $j > 0$ , the set  $A_j$  is unreachable from  $A_0$  in  $G \setminus H$  since all paths from  $A_0$  lead to  $H$  without leaving  $A_0$ .

For the induction step, assume  $A_j$  is unreachable in  $G \setminus H$  from all  $A_\ell$  for  $0 < \ell < i$ . Since  $A_i$  is by definition an attractor of  $S_i$  in  $(G \setminus H) \upharpoonright V_i$ , any path from  $A_i$  in  $(G \setminus H) \upharpoonright V_i$  ends up in  $S_i$  without leaving  $A_i$ , and then  $S_i$  closed under successors in  $(G \setminus H) \upharpoonright V_i$ . Therefore, all paths from  $A_i$  in  $G \setminus H$  can only exit  $A_i$  by entering  $\bigcup_{0 < \ell < i} A_\ell$ . Then, from the induction hypothesis, such a path cannot reach  $A_i$ . ◀

► **Proposition 11 (CB-rank).** *Let  $\kappa$  be an ordinal. A tree has rank  $\kappa$  iff it has CB rank  $\kappa$ .*

▷ **Claim 32.** Let  $t$  be a tree with a  $\kappa$ -pattern  $(V_\alpha)_{\alpha < \kappa}$ , then the following holds for all  $\alpha < \kappa$ :

1.  $V_\alpha$  contains an infinite path,
2.  $V_\alpha$  contains only limit vertices, for  $\alpha > 0$ .

**Proof.** We prove the following statement: a tree has a  $\kappa$ -pattern if and only if its  $\alpha$ th derivative is non-empty, for all  $\alpha < \kappa$ . This is enough to get the proposition, and vacuously holds for  $\kappa = 0$ .

Let  $t$  be a tree with a  $\kappa$ -pattern  $(V_\alpha)_{\alpha < \kappa}$ . We prove by induction on ordinals that  $\text{der}^\alpha(t)$  is non-empty for all  $\alpha < \kappa$ .

If  $\kappa = \alpha + 1$ , then  $\text{der}^\kappa(t) = \text{der}(\text{der}^\alpha(t))$ . We show by induction on  $\beta \leq \alpha$  that  $V_\beta$  is a subset of the vertices of  $\text{der}^\beta(t)$ . It holds for  $\beta = 0$ . Assume it holds for some  $\beta$ . Then  $V_\beta$  is a subset of the vertices of  $\text{der}^\beta(t)$ , which means that  $(V_\beta, V_{\beta+1})$  is a 2-pattern of  $\text{der}^\beta(t)$ , hence  $V_{\beta+1}$  contains only limit vertices of  $\text{der}^\beta(t)$  from Claim 32. Thus  $\text{der}(\text{der}^\beta(t))$  contains all the vertices of  $V_{\beta+1}$ . Similarly, if  $\beta$  is a limit ordinal, then for all  $\gamma < \beta$ ,  $V_\gamma$  is a subset of  $\text{der}^\gamma(t)$ , which means that the vertices of  $V_\beta$  are limit vertices of  $\text{der}^\gamma(t)$ , for all  $\gamma$ , thus  $V_\beta$  is included in the vertex set of  $\text{der}^\beta(t)$ . In particular  $\text{der}^\alpha(t)$  contains all the vertices of  $V_\alpha$  which contains an infinite path. Hence  $\text{der}^\alpha(t)$  is not empty.

## 164:18 When Is the Parity Index Hierarchy Finite?

If  $\kappa$  is a limit ordinal, then by the induction assumption,  $t$  has an  $\alpha+1$ -pattern for any  $\alpha < \kappa$ . Thus by the induction assumption, the  $\alpha$ th derivative of  $t$  is non-empty, which concludes the proof.

Conversely, let  $t$  be a tree such that for all  $\alpha < \kappa$ ,  $\text{der}^\alpha(t)$  is non-empty. We prove by induction on  $\kappa$  that  $(V_\alpha)_{\alpha < \kappa}$ , defined by  $V_\alpha = \text{der}^\alpha(t) \setminus \text{der}^{\alpha+1}(t)$  is a  $\kappa$ -pattern. For  $\kappa = 0$  this clearly holds. If  $\kappa$  is a limit ordinal, then by induction assumption we have that  $(V_\beta)_{\beta < \alpha+1}$  is an  $\alpha+1$ -pattern, for any  $\alpha < \kappa$ . Thus  $(V_\alpha)_{\alpha < \kappa}$  is a  $\kappa$ -pattern. If  $\kappa = \alpha+1$ , then by induction assumption,  $(V_\beta)_{\beta < \alpha}$  is an  $\alpha$ -pattern. The vertices of  $V_\alpha$  are the vertices of  $\text{der}^\alpha(t)$  which are not limit vertices. Note that  $\text{der}^\alpha(t)$  does not contain any terminal vertex. Thus vertices in  $V_\alpha$  all have a successor in  $V_\alpha$ . We have left to show that any vertex in  $V_\alpha$  can reach a smaller component by a component decreasing path. Assume to the contrary that there is a vertex  $v \in V_\alpha$  and  $\beta < \alpha$  such that there is no path in  $vV_\alpha^* \cdot (\bigcup_{\beta \leq \beta' < \alpha} V_{\beta'})$ . Since  $V_\alpha$  is the set of vertices of  $\text{der}^\alpha(t)$  which are not limit vertices (and since  $\text{der}^\alpha(t)$  has not terminal vertex), this means that it is descendant-closed in  $\text{der}^\alpha(t)$ . Hence, in  $\text{der}^\beta(t)$ , all paths from  $v$  stay in  $V_\alpha$ , thus  $v$  is not a limit vertex of  $\text{der}^\beta(t)$ , which gives a contradiction. Hence the family  $(V_\alpha)_{\alpha < \kappa}$  is a  $\kappa$ -pattern.  $\triangleleft$

### B Proofs of Section 4

► **Lemma 13 (Glueing Lemma).** *Let  $G$  be a graph and let  $V = (V_0, \dots, V_{k-1})$  be a tree-like  $k$ -pattern of  $G$ , with  $V_i = \biguplus_{\ell \in \mathbb{N}} V_i^\ell$  satisfying the corresponding properties. Let  $W^\ell = (W_0^\ell, \dots, W_{k'}^\ell)$  be tree-like  $k'+1$ -patterns of  $G$  such that  $W_{k'}^\ell = V_0^\ell$ , for all  $\ell \in \mathbb{N}$ . Then  $G$  admits a tree-like  $k+k'$ -pattern.*

**Proof.** We define  $U = (U_0, \dots, U_{k+k'-1})$ . For  $i \in \{0, \dots, k-1\}$ ,  $U_{k'+i} = V_i$ , and for  $i \in \{0, \dots, k'-1\}$ ,  $U_i = \biguplus_{\ell \in \mathbb{N}} W_i^\ell$ . Note that the unions are indeed disjoint, since the  $V_0^\ell$ s have disjoint reachability sets. Let us check that  $U$  is indeed a pattern. The fact that the  $U_i$ s are disjoint is given by the two following properties: 1) for distinct  $\ell$ s, the  $V_0^\ell$ s have different reachability sets, thus the different patterns  $W^\ell$ s are disjoint, 2) for each pattern  $W^\ell$ , the components are disjoint, by definition.

We only have left to check that any vertex can reach any component of lower index by an index decreasing path. If the two indices are at least  $k'$ , then this is given by  $V$  being a pattern. If the two indices are at most  $k'$ , then any vertex is in pattern  $W^\ell$  for some  $\ell$ , and can reach any lower component of  $U$ . To reach a component of index at most  $k'$  from a component at least  $k'$ , then one can simply reach component  $k'$  first (since  $V$  is a pattern) and end up in some  $V_0^\ell$  and then reach the component of smaller index, since  $W^\ell$  is a pattern.  $\triangleleft$

► **Proposition 19.** *Let  $t$  be a tree of rank  $k$  and let  $t'$  be an  $n$ -subdivision of  $t$ . Then  $t'$  has rank at most  $k$ .*

**Proof.** Note that an  $n$ -subdivision can be obtained as  $n$  successive 1-subdivisions, hence we only have to show that the result holds for  $n = 1$ .

Let  $t$  be a tree and let  $t'$  be a 1-subdivision of  $t$ . Let us assume that  $t'$  has a  $k$ -pattern  $(V_0, \dots, V_{k-1})$ . We can partition the vertices of  $t'$  in  $V \uplus X$ , where  $V$  is the set of vertices of  $t$  and  $X$  is the set of vertices added in subdivided vertices. Note that vertices in  $X$  have only one successor, which means that any vertex in  $X$  in a  $V_i$  must have its only successor in  $V_i$ . Let  $u$  be a vertex in  $V \cap V_i$  and let  $v$  be a successor of  $u$  in  $V_i$  in  $t'$ . If  $v \in V$ , then  $u$  has a successor in  $V_i \cap V$  in  $t$ . Otherwise if  $v \in X$ , then it has a successor  $w \in V_i$  such that

$(u, w)$  is an edge in  $t$ . In both cases  $u$  has a successor in  $V_i$  in  $t$ . With the same reasoning, we can show that if  $u$  has a successor in  $V_{i-1}$  in  $t'$  then it has a successor in  $V_{i-1}$  in  $t$ . Thus we have that  $(V_0 \cap V, \dots, V_{k-1} \cap V)$  is a  $k$ -pattern in  $t$ .  $\blacktriangleleft$

► **Corollary 20.** *Given a tree  $t$  with rank  $k$  and an APT  $\mathcal{A}$ , the acceptance parity game  $\mathcal{G}(t, \mathcal{A})$  has rank at most  $kc$  for some constant  $c$  that depends only on  $\mathcal{A}$ .*

**Proof.** Observe that the number of successive positions in  $\mathcal{G}(t, \mathcal{A})$  with the same  $t$ -component is bounded by a constant  $b$  that depends on the depth of Boolean combinations in  $\mathcal{A}$ . By padding these with trivial disjunctions or conjunctions, we can ensure that the number of successive positions with the same  $t$ -component is in fact constant throughout  $\mathcal{G}(t, \mathcal{A})$ . Then, there is a subdivision  $t'$  of  $t$ , also of rank  $k$  (from Proposition 19), such that  $\mathcal{G}(t, \mathcal{A})$  is a subgraph of the product  $t' \times K_c$  where  $c$  is  $(|Q| + |\mathbb{B}(Q)|) \times (1 + |\Sigma|) + 2$ , obtained from counting the positions of  $\mathcal{G}(t, \mathcal{A})$ . From Lemma 17 since  $\mathcal{G}(t, \mathcal{A})$  is a sub-graph of  $t' \times K_c$ , it has rank at most  $kc$ .  $\blacktriangleleft$

### Full definition of $\mathcal{B}_{p,k}$

We build  $\mathcal{B}_{p,k}$  for even  $p$ , by induction on  $p$ , starting with  $\mathcal{B}_{0,k}$ , which, for all  $k$  simply consists of an accepting state  $q$  and a transition  $\delta(q, 0) = \Box q$ .

Then, we build  $\mathcal{B}_{p+2,k}$  as follows. We take  $k$  copies  $\mathcal{B}_k, \dots, \mathcal{B}_1$  of  $\mathcal{B}_{p,k}$  with initial states  $\iota_k, \dots, \iota_1$ , respectively, and add a non-accepting state  $q_i$  before each  $\mathcal{B}_i$ , as well as  $q_0$  after  $\mathcal{B}_1$ , each with a nondeterministic choice over priorities up to  $p+1$  to either loop or move to the next copy of  $\mathcal{B}_{p,k}$ . From  $q_0$ , there is only a self-loop on priorities up to  $p+1$ . From every state of each  $\mathcal{B}_i$ , seeing the input letter  $p+1$  leads to the intermediate state  $q_{i-1}$ . Finally, we also add one accepting state  $q_b$ , to which all states transition upon seeing  $p+2$ , and from which there is a transition over all priorities up to  $p+2$  to  $q_k$ , the first new non-accepting state.

More formally, having  $\mathcal{B}_1, \dots, \mathcal{B}_j$ , we make  $\iota_1, \dots, \iota_k$  non-initial, and add  $k+1$  non-accepting states  $q_k, \dots, q_0$ , of which  $q_k$  is initial, an accepting state  $q_b$ , and the following transitions:

- $\delta(q_0, x) = \Box q_0$ , for  $x < p+2$ ,
- $\delta(q_i, x) = \Box q_i \vee \Box \iota_i$ , for  $x < p+2$ ,  $i > 0$ ,
- $\delta(q, p+1) = \Box q_{i-1}$  for  $q$  a state in  $\mathcal{B}_i$ ,  $i > 0$ ,
- $\delta(q, p+2) = \Box q_b$  for all  $q$ ,
- $\delta(q_b, x) = \Box q_k$  for all  $x$ ,
- transitions on  $x$  in  $\mathcal{B}_i$  are preserved for  $x < p+1$ .

Finally, let  $\mathcal{B}_{p,k}$  for odd  $p$  be as  $\mathcal{B}_{p+1,k}$ , except without any transitions on priority  $p+1$ .

► **Theorem 23.** *Given  $k \in \mathbb{N}$  and an alternating parity automaton  $\mathcal{A}$ , one can effectively construct an alternating Büchi automaton  $\mathcal{B}$  that agrees with  $\mathcal{A}$  over trees of Cantor-Bendixson rank up to  $k$ . If  $\mathcal{A}$  is nondeterministic, so is  $\mathcal{B}$  and if  $\mathcal{A}$  is coBüchi then  $\mathcal{B}$  is weak.*

**Proof.** Consider an APT  $\mathcal{A}$  with priorities up to  $p$ . Let  $k' = kc + 1$  for the constant  $c$  in Corollary 20; then we know that the game  $\mathcal{G}(t, \mathcal{A})$  has rank strictly less than  $k'$  for all trees  $t$  of rank up to  $k$ .

By Lemma 22, the ABT  $\mathcal{B}_{p,k'}$  accepts all parity graphs with priorities up to  $p$  with a  $k'$ -width attractor-decomposition, and rejects odd parity graphs. By construction,  $\mathcal{B}_{p,k'}$  only has  $\Box$  modalities and disjunctions. Then, we build the required ABT  $\mathcal{B}$  by taking the synchronous composition  $\mathcal{A} \times \mathcal{B}_{p,k'}$  of  $\mathcal{A} = (\Sigma, Q^A, \delta^A, \delta_\iota^A, \Omega^A)$  and  $\mathcal{B}_{p,k'} = ([0, p], Q^B, \delta^B, \delta_\iota^B, \Omega^B)$ . This is an ABT that accepts a tree  $t$  if and only if  $\mathcal{B}_{p,k'}$  accepts some tree induced by a strategy for Eve in  $\mathcal{G}(t, \mathcal{A})$ . The details of this composition are as expected:

- Alphabet  $\Sigma$ ; states  $Q^A \times Q^B$ ;

## 164:20 When Is the Parity Index Hierarchy Finite?

- $\delta((q_A, q_B), x) = f(\delta^A(q_A, x), q_B)$  where:
  - $f(b \wedge b', q) = f(b, q) \wedge f(b', q)$
  - $f(b \vee b', q) = f(b, q) \vee f(b', q)$
  - $f(\Box b, q) = \Box f(b, q)$
  - $f(\Diamond b, q) = \Diamond f(b, q)$
  - $f(q, b' \vee b'') = f(q, b') \vee f(q, b'')$
  - $f(q, \Box b') = (q, b')$
  - $f(q', q) = f(q', \delta^B(q, \Omega(q')))$
- $\delta_i = f(\delta_i^A, \delta_i^B);$
- $\Omega(q^A, q^B) = \Omega^B(q^B).$

In words,  $\mathcal{B}$  is an automaton that uses  $\mathcal{A}$  to process the input tree, and at each step,  $\mathcal{B}_{p,k'}$  processes the priority that  $\mathcal{A}$  would produce, which adds some nondeterministic choices. We argue  $\mathcal{B}$  accepts any tree  $t$  such that in  $\mathcal{G}(t, \mathcal{A})$  Eve has a winning strategy that induces a graph with a  $k'$ -width attractor decomposition, and rejects  $t \notin L(\mathcal{A})$ .

Assume  $t$  is such that in  $\mathcal{G}(t, \mathcal{A})$  Eve has a winning strategy  $\sigma$  that induces an even graph  $G_\sigma$  with a  $k'$ -width attractor decomposition. Let  $\sigma'$  be Eve's winning strategy in  $\mathcal{G}(G_\sigma, \mathcal{B}_{p,k'})$ . We can then compose these strategies to obtain a winning strategy in  $\mathcal{G}(t, \mathcal{B})$ . Indeed, positions of  $\mathcal{G}(t, \mathcal{B})$  can be seen as a position of  $\mathcal{G}(t, \mathcal{A})$  paired with states of  $\mathcal{B}_{p,k'}$ , to which are added, after the resolution of the boolean combinations in transition conditions of  $\mathcal{A}$ , the resolution of the boolean combinations in the transition condition of  $\mathcal{B}_{p,k'}$  corresponding to a transition on the priority of the next state in  $\mathcal{A}$ . In the composition strategy,  $\sigma$  determines the choices between successors in  $t$  at  $\Diamond$ -modalities and choices at disjunctions in the transition conditions of  $\mathcal{A}$ , while  $\sigma'$  determines choices at disjunctions in the transition conditions of  $\mathcal{B}_{p,k'}$ . The existence of  $\sigma$  guarantees the existence of  $\sigma'$ , and that the composition strategy remains in the restriction of  $\mathcal{G}(t, \mathcal{B})$  to  $G_\sigma$  in the  $\mathcal{G}(t, \mathcal{A})$ -part of the game. Since the priorities of  $\mathcal{B}$  come from the priorities of  $\mathcal{B}_{p,k'}$ , the fact that  $\sigma'$  is winning for Eve ensures that the composition of  $\sigma$  and  $\sigma'$  is winning for Eve.

Conversely, if  $t \notin L(\mathcal{A})$ , then Adam has a winning strategy  $\tau$  in  $\mathcal{G}(t, \mathcal{A})$ . He can use  $\tau$  directly in  $\mathcal{G}(t, \mathcal{B})$  as  $\mathcal{B}$  does not add any universal choices ( $\Box$  or  $\wedge$ ) beyond those of  $\mathcal{A}$ . Let  $w$  be the sequence of priorities induced by the projection of a  $\tau$ -consistent play of  $\mathcal{G}(t, \mathcal{B})$  on the  $\mathcal{G}(t, \mathcal{A})$  component; the highest priority seen infinitely often on  $w$  is odd. The projection of the same play on  $\mathcal{B}_{p,k}$  is a path of  $\mathcal{B}_{p,k}$  that processes  $w$ . By construction, no accepting cycle of  $\mathcal{B}_{p,k}$  processes a word of which the highest maximal priority is odd, so this path is rejecting, and  $\tau$  is a winning strategy for Adam in  $\mathcal{G}(t, \mathcal{B})$ .

For all  $t$  of rank up to  $k$ ,  $\mathcal{G}(t, \mathcal{A})$  has rank strictly less than  $k'$ , if  $t \in L(\mathcal{A})$  then Eve has a winning strategy that induces an even graph that also has rank less than  $k'$ . From Lemma 8, it has a  $k'$ -width attractor decomposition, and therefore  $\mathcal{B}$  accepts  $t$ ; if  $t \notin L(\mathcal{A})$  then  $\mathcal{B}$  rejects  $t$ , so  $\mathcal{B}$  agrees with  $\mathcal{A}$  on such trees. By construction, if  $\mathcal{A}$  is nondeterministic, so is  $\mathcal{B}$  as  $\mathcal{B}_{p,k'}$  only has nondeterministic choices; if  $\mathcal{A}$  is coBüchi, then  $\mathcal{B}$  is weak since  $\mathcal{B}_{1,k'}$  is weak. ◀