Unambiguous Languages Exhaust the Index Hierarchy

Michał Skrzypczak
University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
mskrzypczak@mimuw.edu.pl
https://orcid.org/0000-0002-9647-4993

Abstract

This work is a study of the expressive power of unambiguity in the case of automata over infinite trees. An automaton is called unambiguous if it has at most one accepting run on every input, the language of such an automaton is called an unambiguous language. It is known that not every regular language of infinite trees is unambiguous. Except that, very little is known about which regular tree languages are unambiguous.

This paper answers the question whether unambiguous languages are of bounded complexity among all regular tree languages. The notion of complexity is the canonical one, called the (parity or Rabin-Mostowski) index hierarchy. The answer is negative, as exhibited by a family of examples of unambiguous languages the cannot be recognised by any alternating parity tree automata of bounded range of priorities.

Hardness of the examples is based on the theory of signatures, previously studied by Walukiewicz. The technical core of the article is a definition of the canonical signatures together with a parity game that compares signatures of a given pair of parity games (of the same index).

2012 ACM Subject Classification Theory of computation → Tree languages

Keywords and phrases unambiguous automata, parity games, infinite trees, index hierarchy

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.140

Related Version For a full version see `https://arxiv.org/abs/1803.06163`.

Funding This work has been supported by Poland's NSC (NCN) grant 2016/21/D/ST6/00491.

Acknowledgements The author would like to thank Szczepan Hummel, Damian Niwiński, and Igor Walukiewicz for inspiring and fruitful discussions on the topic. Moreover, the author is grateful to Bartek Klin, Kamila Łyczek, Filip Murlak, Grzegorz Rząca, and the anonymous referees for a number of editorial suggestions about the paper.

1 Introduction

Non-determinism provides a machine with a very powerful ability to *guess* its choices. Depending on the actual model, it might enhance the expressive power of the considered machines or, while preserving the class of recognised languages, make the machines more succinct or effective. All these benefits come at the cost of algorithmic difficulties when handling non-deterministic devices. This complexity motivates a search of ways of restricting the power of non-determinism. One of the most natural among these restrictions is a semantic notion called *unambiguity*: a non-deterministic machine is called *unambiguous* if it has at most one accepting run on every input.

Unambiguity turns out to be very intriguing in the context of automata theory [7]. In the classical case of finite words it does not enhance the expressive power of the automata, still it

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 140; pp. 140:1–140:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

simplifies some decision problems [16]. The situation is more complex in the case of infinite trees: the language of infinite trees labelled $\{a, b\}$ containing a letter a cannot be recognised by any unambiguous parity automaton [15, 6]. This example makes the impression that very few regular languages of infinite trees are in fact unambiguous (i.e. can be recognised by an unambiguous automaton). However, there is only a couple of distinct examples of ambiguous regular tree languages [3]. Our understanding of how many (or which) regular tree languages are unambiguous is far from being complete, in particular it is not known how to decide if a given regular tree language is unambiguous.

Another way of understanding the power of unambiguous tree languages is aimed at estimating their descriptive complexity. The complexity can be measured either in terms of the topological complexity or of the parity index, i.e. the range of priorities needed for an alternating parity tree automaton to recognise a given language. Initially, it was considered plausible that all unambiguous tree languages are co-analytic ($\mathbf{\Pi}^1_1$); that is topologically not more complex than deterministic ones. Hummel in [11] gave an example of an unambiguous language that is $\mathbf{\Sigma}^1_1$-complete, in particular not $\mathbf{\Pi}^1_1$. Further improvements [8, 12] showed that unambiguous languages reach high into the second level of the index hierarchy. However, the question whether this is an upper bound on their index complexity was left open. In this paper we prove that it is not the case, as expressed by the following theorem.

▶ **Theorem 1.** *For every $i < k$ there exists an unambiguous tree language L that cannot be recognised by any alternating parity tree automaton (*ATA*) that uses priorities $\{i, \ldots, k\}$. In other words, L does not belong to the level (i, k) of the index hierarchy.*

The canonical examples of languages lying high in the index hierarchy [4, 1] are the languages $W_{i,k}$ dating back to [9, 21] (see e.g. the formulae W_n in [1]). Unfortunately, the languages $W_{i,k}$ are not unambiguous—one can interpret the choice problem [6] in such a way that witnessing unambiguously that $t \in W_{1,2}$ would indicate an MSO-definable choice function [10, 5]. Therefore, to prove Theorem 1 we will use the following corollary of [2].

▶ **Corollary 2** ([1, 2])**.** *Let L be a set of trees. If there is a continuous function f s.t. $W_{i,k} = f^{-1}(L)$ then L cannot be recognised by an ATA of index $(i+1, k+1)$.*

Our aim is to enrich in a continuous way a given tree t with some additional information denoted $f(t)$, such that an unambiguous automaton reading $f(t)$ can verify if $t \in W_{i,k}$. Although this method is based on the topological concept of a continuous mapping f, the construction provided in this paper is purely combinatorial; the core is a definition of a parity game $\mathcal{C}_P$ that compares the difficulty of a given pair of parity games.

2 Basic notions

We use $u \cdot w$ to represent the concatenation of the two sequences. The symbol $\preceq$ stands for the prefix order. By $\omega = \{0, 1, \ldots\}$ we denote the set of natural numbers.

A *(ranked) alphabet* is a non-empty finite set A of *letters* where each letter $a \in A$ comes with its own finite *arity*. A *tree* over an alphabet A is a partial function $t\colon \omega^* \rightharpoonup A$ where the domain $\mathrm{dom}(t)$ is non-empty, prefix-closed, and if $u \in \mathrm{dom}(t)$ is a *node* with a $k$ary letter $a = t(u)$ then $u \cdot l \in \mathrm{dom}(t)$ if and only if $l < k$, i.e. u (the *father*) has *children* $u \cdot 0$, $u \cdot 1$, $\ldots$, $u \cdot (k-1)$. The set of all trees over A is denoted Tr_A. The element $\epsilon \in \mathrm{dom}(t)$ is called the *root* of t. A *branch* of a tree t is a sequence $\alpha \in \omega^\omega$ such that for all $n \in \omega$ the finite prefix $\alpha{\restriction}n$ is a node of t. It is easy to encode ranked alphabets using alphabets of fixed arity (or even binary), however for the technical simplicity we will work with ranked ones here.

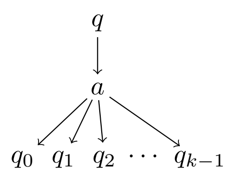

Figure 1 A representation of a transition $(q, a, q_0, \ldots, q_{k-1})$, for a $k$ary letter a.

If $u \in \mathrm{dom}(t)$ is a node of a tree t then by $t{\restriction}u$ we denote the tree $w \mapsto t(u \cdot w)$ with the domain $\{w \mid u \cdot w \in \mathrm{dom}(t)\}$. A tree of the form $t{\restriction}u$ for $u \in \mathrm{dom}(t)$ is called a *subtree* of t in u. If $a \in A$ is a $k$ary letter and $t_0, \ldots, t_{k-1}$ are trees then by $a\big(t_0, \ldots, t_{k-1}\big)$ we denote the unique tree t with $t(\epsilon) = a$ and $t{\restriction}(i) = t_i$ for $i = 0, \ldots, k-1$.

If $X \subseteq \mathrm{dom}(t)$ is a set of nodes of a tree t and $u \in \mathrm{dom}(t)$ then by $X{\restriction}u$ we denote the set $\{w \mid u \cdot w \in X\}$, which is a subset of $\mathrm{dom}(t{\restriction}u)$.

Automata. A *non-deterministic parity tree automaton* is a tuple $\mathcal{A} = \langle A, Q, \Delta, I, \Omega\rangle$, where A is a ranked alphabet; Q is a finite set of *states*; Δ is a finite set of *transitions*—tuples of the form $(q, a, q_0, \ldots, q_{k-1})$ where $a \in A$ is a $k$ary letter and $q, q_0, \ldots, q_{k-1}$ are states; $I \subseteq Q$ is a set of *initial states*; and $\Omega\colon Q \to \omega$ is a *priority* mapping.

A *run* of an automaton $\mathcal{A}$ over a tree t over the alphabet A is a function $\rho\colon \mathrm{dom}(t) \to Q$ such that $\rho(\epsilon) \in I$ and for every $u \in \mathrm{dom}(t)$ with a $k$ary letter $a = t(u)$, the tuple $\big(\rho(u), a, \rho(u \cdot 0), \ldots, \rho(u \cdot (k-1))\big)$ is a transition in Δ. A run ρ is *accepting* if for every branch α of t the lowest priority of the states appearing infinitely many times along α (i.e. $\liminf_{n\to\infty} \Omega\big(\rho(\alpha{\restriction}n)\big)$) is even. An automaton $\mathcal{A}$ *accepts* a tree t if there exists an accepting run of $\mathcal{A}$ over t. The language of an automaton $\mathcal{A}$ (denoted $\mathrm{L}(\mathcal{A})$) is the set of trees accepted by $\mathcal{A}$. A set of trees over an alphabet A is called *regular* if it is recognised by a non-deterministic parity tree automaton. For a detailed introduction to the theory of automata over infinite trees, see [18].

An automaton $\mathcal{A}$ is *unambiguous* if for every tree t there exists at most one accepting run of $\mathcal{A}$ over t. An automaton $\mathcal{A}$ is *deterministic* if $I = \{q_\mathrm{I}\}$ is a singleton and for every $q \in Q$ and $k$ary letter $a \in A$ it has at most one transition of the form $(q, a, q_0, \ldots, q_{k-1})$ in Δ. A language is *unambiguous* (resp. *deterministic*) if it can be recognised by an unambiguous (resp. deterministic) automaton. Clearly each deterministic automaton is unambiguous but the converse is not true. Due to [6] we know that there are regular tree languages that are ambiguous (i.e. not unambiguous).

Games. A *game* with players $\mathbf{1}$ and $\mathbf{2}$ is a tuple $\mathcal{G} = \langle V, E, v_\mathrm{I}, W\rangle$ where: $V = V_\mathbf{1} \sqcup V_\mathbf{2}$ is a set of *positions* split into the $\mathbf{1}$*-positions* $V_\mathbf{1}$ and $\mathbf{2}$*-positions* $V_\mathbf{2}$; $E \subseteq V \times V$ is a set of *edges*; $v_\mathrm{I} \in V$ is an *initial position*; and $W \subseteq V^\omega$ is a *winning condition*. We will denote by P the players, i.e. $P \in \{\mathbf{1}, \mathbf{2}\}$, $\bar{P}$ is the opponent of P. For $v \in V$ by $v \cdot E$ we denote the set of successors $\{v' \mid (v, v') \in E\}$. We assume that for each $v \in V$ the set $v \cdot E$ is non-empty. A non-empty finite or infinite sequence $\Pi \in V^{\leq\omega}$ is a *play* if $\Pi(0) = v_\mathrm{I}$ and for each $0 < i < |\Pi|$ there is an edge $\big(\Pi(i-1), \Pi(i)\big)$. Notice that if $\langle V, E\rangle$ is a tree then there is an equivalence between finite plays and positions $v \in V$. An infinite play Π is *winning for* $\mathbf{1}$ if $\Pi \in W$; otherwise Π is winning for $\mathbf{2}$.

A non-empty and prefix-closed set of plays Σ with no $\preceq$-maximal element (i.e. no leaf) is called a *behaviour*. We call a behaviour P*-full* if for every play $(v_0, \ldots, v_n) \in \Sigma$ with $v_n \in V_P$ and all $v' \in v \cdot E$ we have $(v_0, \ldots, v_n, v') \in \Sigma$. We call a behaviour P*-deterministic* if for every play $(v_0, \ldots, v_n) \in \Sigma$ with $v_n \in V_P$ there is a unique $v' \in v \cdot E$ such that $(v_0, \ldots, v_n, v') \in \Sigma$.

A *quasi-strategy* of a player P is a behaviour that is $\bar{P}$-full. A *strategy* of P is a quasi-strategy of P that is P-deterministic. A quasi-strategy is *positional* if the fact whether a play $(v_0, \ldots, v_n, v_{n+1})$ belongs to Σ depends only on v_n.

A *partial strategy* of P is a P-deterministic behaviour—it defines the unique choices of P but may not respond to some choices of $\bar{P}$. We say that a play $(v_0, \ldots, v_n, v_{n+1}) \notin \Sigma$ is *not reachable* by a partial strategy Σ if $(v_0, \ldots, v_n) \in \Sigma$ and $v_n \in V_{\bar{P}}$. If Σ is a (partial) strategy of P and $(v_0, \ldots, v_n, v') \in \Sigma$ with $v_n \in V_P$ then we say that Σ *moves to* v' in $(v_0, \ldots, v_n)$.

A strategy Σ of P is *winning* if every *infinite play of* Σ (i.e. Π such that $\forall n \in \omega.\ \Pi{\restriction}n \in \Sigma$) is winning for P. A game is (*positionally*) *determined* if one of the players has a (positional) winning strategy. We say that a position v of a game $\mathcal{G}$ is *winning for* P (resp. *losing for* P) if P (resp. $\bar{P}$) has a winning strategy in the game $\mathcal{G}$ with $v_{\mathrm{I}} := v$.

Topology. In this work we use only basic notions of descriptive set theory and topology, see [13, 19] for a broader introduction. The space Tr_A with the product topology is homeomorphic to the Cantor space. One can take as the basis of this topology the sets of the form $\{t \in \mathrm{Tr}_A \mid t(u_1){=}a_1, t(u_2){=}a_2, \ldots, t(u_n){=}a_n\}$ for finite sequences $(u_1, a_1, u_2, a_2, \ldots, u_n, a_n)$. The open sets in Tr_A are obtained as unions of basic open sets. A function $f\colon X \to Y$ is continuous if the pre-image of each basic open set in Y is open in X.

3 The languages

Let us fix a pair of numbers $i < k$. Our aim is to encode a general parity game with players $\mathbf{1}$ and $\mathbf{2}$ and priorities $\{i, \ldots, k\}$ as a tree over a fixed ranked alphabet $A_{(i,k)}$. That alphabet consists of: unary symbols $[i], [i{+}1], \ldots, [k]$ indicating priorities of positions; and binary symbols $\langle\mathbf{1}\rangle$ and $\langle\mathbf{2}\rangle$ which leave the choice of the subtree to the respective player.

The game induced by a tree $t \in \mathrm{Tr}_{A_{(i,k)}}$ is denoted $\mathcal{G}(t)$. The set of positions of $\mathcal{G}(t)$ is $\mathrm{dom}(t)$ and the edge relation contains pairs father—child. The initial position is ϵ and a position $v \in \mathrm{dom}(t)$ is a $\mathbf{1}$-position iff $t(v) = \langle\mathbf{1}\rangle$. An infinite play of that game is won by $\mathbf{1}$ if and only if the minimal priority j that occurs infinitely often during the play is even[1]. Since the graph of $\mathcal{G}(t)$ is a tree, we identify finite plays in $\mathcal{G}(t)$ with positions $v \in \mathrm{dom}(t)$. Therefore, (quasi / partial) strategies in $\mathcal{G}(t)$ can be seen as specific subsets $\Sigma \subseteq \mathrm{dom}(t)$ and infinite plays of these strategies as branches of t.

For a player $P \in \{\mathbf{1}, \mathbf{2}\}$ the language $\mathrm{W}_{P,(i,k)}$ contains a tree t if P has a winning strategy in $\mathcal{G}(t)$. It is easy to see that $\mathrm{W}_{\mathbf{1},(i,k)}$ is homeomorphic (i.e. topologically equivalent) to $W_{i,k}$ from [21] (the case of $P = \mathbf{2}$ is dual, we put $W_{i+1,k+1}$ then).

As it turns out, the languages $\mathrm{W}_{P,(i,k)}$ are not expressive enough to allow enrichment of a tree t into $f(t)$, see Proposition 22. To enlarge their expressive power we will extend the alphabet with a unary symbol $\sim$ that will correspond to a swap of the players in $\mathcal{G}(t)$. The enhanced alphabet will be denoted $A^{\sim}_{(i,k)}$. We will say that a tree t over the alphabet $A^{\sim}_{(i,k)}$ is *well-formed* if there is no branch with infinitely many symbols $\sim$.

Consider a node $u \in \mathrm{dom}(t)$ in a well-formed tree t over $A^{\sim}_{(i,k)}$. We will say that u is *switched* if there is an odd number of nodes $w \prec u$ such that $t(w) = \sim$. Otherwise u is *kept*. These notions represent the fact that each symbol $\sim$ swaps the players in $\mathcal{G}(t)$.

[1] We restrict our attention to the trees in which every second symbol on each branch is a unary symbol representing a priority (i.e. $[j]$ for $j \in \{i, \ldots, k\}$); every tree can implicitly be transformed into that format by padding with the maximal priority k (such a padding does not influence the winner of $\mathcal{G}(t)$).

If a tree t is well-formed then the game $\mathcal{G}(t)$ is well-defined in a similar way as above, a *kept* position $v \in \mathrm{dom}(t)$ is a $\mathtt{1}$-position iff $t(v) = \langle \mathtt{1} \rangle$; a *switched* position $v \in \mathrm{dom}(t)$ is a $\mathtt{1}$-position iff $t(v) = \langle \mathtt{2} \rangle$. The language $\mathrm{W}^{\sim}_{P,(i,k)}$ contains a well-formed tree t over $A^{\sim}_{(i,k)}$ if P has a winning strategy in $\mathcal{G}(t)$. The games $\mathcal{G}(t)$ have a parity winning condition and therefore are determined [9, 14], so we obtain:

▶ **Fact 3.** *If $t \in \mathrm{Tr}_{A^{\sim}_{(i,k)}}$ is well-formed then $t \in \mathrm{W}^{\sim}_{P,(i,k)}$ iff $t \notin \mathrm{W}^{\sim}_{\bar{P},(i,k)}$ iff $\sim(t) \in \mathrm{W}^{\sim}_{\bar{P},(i,k)}$.*

The additional information added by f will be kept under third children of new ternary variants of the symbols $\langle \mathtt{1} \rangle$ and $\langle \mathtt{2} \rangle$, denoted $\langle \mathtt{1}+ \rangle$ and $\langle \mathtt{2}+ \rangle$. Let the alphabet $A^{+\sim}_{(i,k)}$ be $A^{\sim}_{(i,k)}$ where instead of the symbols $\langle \mathtt{1} \rangle$ and $\langle \mathtt{2} \rangle$ we have $\langle \mathtt{1}+ \rangle$ and $\langle \mathtt{2}+ \rangle$ respectively.

Consider a tree r over the extended alphabet $A^{+\sim}_{(i,k)}$. By $\mathrm{shave}(r)$ we denote the tree over the non-extended alphabet $A^{\sim}_{(i,k)}$, where instead of each subtree of the form $\langle \mathtt{1}+ \rangle(t_{\mathtt{L}}, t_{\mathtt{R}}, t_2)$ one puts the subtree $\langle \mathtt{1} \rangle(t_{\mathtt{L}}, t_{\mathtt{R}})$; the same for $\langle \mathtt{2}+ \rangle$ and $\langle \mathtt{2} \rangle$. Notice that $\mathrm{dom}(\mathrm{shave}(r)) \subseteq \mathrm{dom}(r)$ and the labels of $\mathrm{shave}(r)$ correspond to the labels of r in the respective nodes (up to the additional $+$ in r). We will say that a tree r over the alphabet $A^{+\sim}_{(i,k)}$ is *well-formed* if for every its subtree $r' = r{\restriction}u$ the tree $\mathrm{shave}(r')$ is well-formed in the standard sense. In other words, r is well-formed if there is no branch of r that contains infinitely many symbols $\sim$ but only finitely many directions 2 (the direction 2 corresponds to moving outside $\mathrm{shave}(r')$).

We are now in position to define the witnesses proving Theorem 1. For that we will define an unambiguous automaton $\mathcal{U}$ recognising certain language of trees over the alphabet $A^{+\sim}_{(i,k)}$. In this section we will prove that $\mathcal{U}$ is unambiguous. In the rest of the article we show that $\mathcal{U}$ (with a restricted set of initial states) recognises a language high in the index hierarchy.

▶ **Definition 4.** The set of states of $\mathcal{U}$ is $\{0, \ldots, k{+}2\} \times \{\mathtt{1}, \mathtt{2}\} \times \{\mathtt{D}, \mathtt{L}, \mathtt{R}\}$. Let $\Omega(j, P, d) = j$. The transitions of $\mathcal{U}$ are depicted in Figure 2. The set of initial states of $\mathcal{U}$ contains all the states of the form $(0, \star, \star)$ (recall that as in Figure 2, $\star$ represents all the possible choices).

Intuitively, the first coordinate of a state q of $\mathcal{U}$ is its priority; the second coordinate is the winner for $\mathcal{G}(\mathrm{shave}(r'))$ for the current subtree r'; while the third coordinate indicates the actual strategy if there is ambiguity and $\mathtt{D}$ otherwise. The transition over $\sim$ represents that $\sim$ swaps the players; the next two transitions correspond to positions that are not controlled by a (claimed) winner P over a given subtree; and the last two transitions correspond to a position that is controlled by P. In the lower two transitions the choice of a direction $\mathtt{L}$ or $\mathtt{R}$ depends on the declared winner P in the third child of the current node.

Consider a run ρ of $\mathcal{U}$ over a tree r, let $u \in \mathrm{dom}(r)$, and assume that $\rho(u)$ is of the form $(\star, P, \star)$. In that case one can extract from the third coordinates of ρ a strategy Σ of P in $\mathcal{G}(\mathrm{shave}(r{\restriction}u))$ that will be called the *ρ-strategy in u*. This strategy is defined inductively, preserving the invariant that for each $w \in \Sigma$ the node w is *kept* in $\mathrm{shave}(r{\restriction}u)$ if and only if $\rho(u \cdot w)$ is of the form $(\star, P, \star)$. We start with Σ containing the initial position ϵ. Now consider a position w in Σ. If w is controlled by P (i.e. $r(u \cdot w) = \langle P+ \rangle$ for w *kept* and $\langle \bar{P}+ \rangle$ for w *switched*) then the strategy Σ moves to the position $w \cdot 0$ (resp. $w \cdot 1$) if the state $\rho(u \cdot w) = (\star, \star, d)$ satisfies $d = \mathtt{L}$ (resp. $d = \mathtt{R}$). In the positions $w \in \Sigma$ not controlled by P the strategy Σ has no choice and contains all the children of w in $\mathrm{shave}(r{\restriction}u)$. It is easy to check that the transitions of $\mathcal{U}$ guarantee that the invariant is preserved.

▶ **Lemma 5.** *Let ρ be a run of $\mathcal{U}$ over r. Then ρ is accepting if and only if r is well-formed and for every $u \in \mathrm{dom}(r)$ the ρ-strategy in u is winning in $\mathcal{G}\big(\mathrm{shave}(r{\restriction}u)\big)$.*

Proof. First observe that $\mathcal{U}$ uses states of priority 0 and 1 to deterministically verify that r is well-formed. Consider a well-formed tree r and a run ρ. On the branches following the

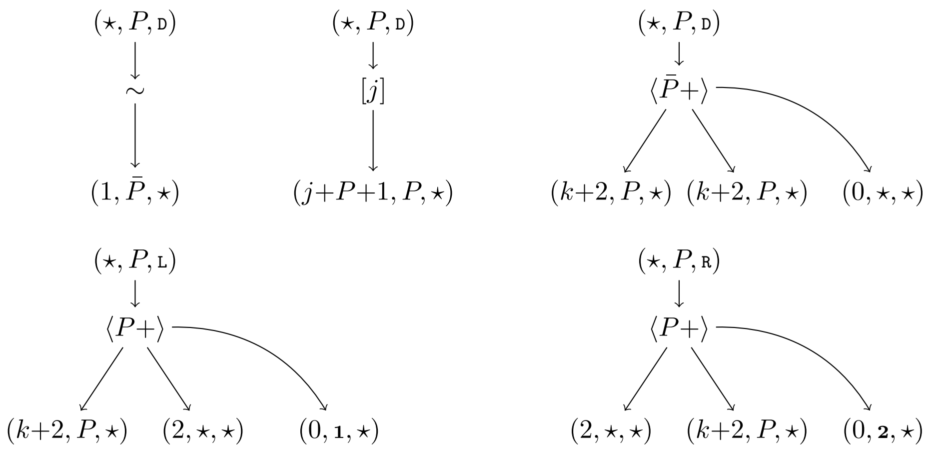

Figure 2 The transitions of the automaton $\mathcal{U}$, where $P \in \{\mathtt{1}, \mathtt{2}\}$ stands for a player; $j \in \{i, \ldots, k\}$ is a priority; and $\star$ represents all the possible choices on a given coordinate.

ρ-strategies, the priorities of ρ correspond to the priorities visited in $\mathcal{G}\big(\mathrm{shave}(r{\restriction}u)\big)$ (up to a shift by 1 or 2 depending on the current second coordinate of ρ). Thus, ρ satisfies the parity condition on all the branches of r if and only if all the ρ-strategies are winning. ◀

The following fact follows directly from the above lemma.

▶ **Fact 6.** *Assume that ρ is an accepting run of $\mathcal{U}$ over a tree r, $u \in \mathrm{dom}(r)$, and $\rho(u)$ is of the form $(\star, P, \star)$. Then P wins $\mathcal{G}\big(\mathrm{shave}(r{\restriction}u)\big)$. In particular, if ρ and ρ' are two accepting runs of $\mathcal{U}$ over the same tree r then the second coordinates of ρ and ρ' are equal.*

▶ **Lemma 7.** *If ρ and ρ' are two runs of $\mathcal{U}$ (possibly not accepting) over a tree r and the second coordinates of ρ and ρ' are equal then $\rho = \rho'$.*

Proof. The third coordinate of a run in $u \in \mathrm{dom}(r)$ depends on $r(u)$, the second coordinate of the run in u, and (if one of the lower two transitions from Figure 2 is used) the second coordinate of the run in $u \cdot 2$. Thus, the third coordinates of ρ and ρ' must agree.

The first coordinates of ρ and ρ' in the root are 0. Consider a node u and its child $u' \in \mathrm{dom}(r)$. The first coordinate of a run in u' depends on $r(u)$ and the last two coordinates of the run in u. Therefore, also the first coordinates of ρ and ρ' must agree. ◀

▶ **Definition 8.** Take $P \in \{\mathtt{1}, \mathtt{2}\}$ and let $L_{P,(i,k)}$ be the language recognised by the automaton $\mathcal{U}$ with the set of initial states restricted to the states of the form $(0, P, \star)$.

Fact 6 together with Lemma 7 imply that the languages $L_{P,(i,k)}$ are unambiguous. Thus, to complete the proof of Theorem 1, one needs to prove that the languages $L_{P,(i,k)}$ climb up the index hierarchy. This will be done using Corollary 2 in the next section.

4 Construction of f

Fix a pair $i < k$. In this section we will prove the following proposition.

▶ **Proposition 9.** *There exists a continuous function $f\colon \mathrm{Tr}_{A^{\sim}_{(i,k)}} \to \mathrm{Tr}_{A^{+\sim}_{(i,k)}}$ such that:*

- *If t is well-formed then $f(t)$ is also well-formed.*
- *For a well-formed t and a player P we have: $t \in \mathrm{W}^{\sim}_{P,(i,k)}$ if and only if $f(t) \in L_{P,(i,k)}$.*

Before proving that proposition, notice that a tree over the alphabet $A_{(i,k)}$ can be seen as a well-formed tree over $A^{\sim}_{(i,k)}$ and $\mathrm{W}^{\sim}_{\mathbf{1},(i,k)} \cap \mathrm{Tr}_{A_{(i,k)}} = \mathrm{W}_{\mathbf{1},(i,k)}$ which is topologically equivalent to $W_{i,k}$. Therefore, the above proposition implies that $f{\restriction}\mathrm{Tr}_{A_{(i,k)}}$ satisfies the assumptions of Corollary 2. As $i < k$ are arbitrary, Theorem 1 follows.

To properly define f we first need to introduce a notion that allows to compare how much a player P *prefers* one tree over another. This is achieved by assigning to each tree its P-*signature* and proving that moving to a subtree with an optimal signature guarantees that the player wins whenever possible. The theory of signatures comes from [17] and results of Walukiewicz, e.g. [20, 21]. The notion of signatures used here is more demanding than the classical ones, as we require equalities instead of inequalities in the invariants from Lemma 10.

We say that a number $j \in \omega$ is P-*losing* if $i \leq j \leq k$ and j is odd (resp. even) if $P = \mathbf{1}$ (resp. $P = \mathbf{2}$). A number $j \in \{i, \ldots, k\}$ that is not P-losing is called P-*winning*. A P-signature is either ∞ or a tuple of countable ordinals $(\theta_{i'}, \theta_{i'+2}, \ldots, \theta_{k'})$, indexed by P-losing numbers—i' is the minimal and k' is the maximal P-losing number. P-signatures that are not ∞ are well-ordered by the lexicographic order $\leq_{\mathrm{lex}}$ in which the ordinals with smaller indices are more important. Let ∞ be the maximal element of $\leq_{\mathrm{lex}}$.

A $\mathbf{1}$-signature $\sigma_{\mathbf{1}}(t) = (42)$ with $i = 0$ and $k = 2$ means that the best what player $\mathbf{1}$ can hope for is to visit at most 42 times a [1]-node (possibly interleaved with nodes of priority 2) before the first [0]-node is visited (if ever). After visiting a [0]-node, the $\mathbf{1}$-signature of the subtree may grow, starting again a counter of nodes of priority 1 to be visited. The $\mathbf{1}$-signature (ω) means that $\mathbf{2}$ can choose a finite number of [1]-nodes that will be visited; however the choice needs to be done before the first such node is seen.

The following two lemmas express the crucial properties of the signatures.

▶ **Lemma 10.** *There exists a unique point-wise minimal pair of assignments $t \mapsto \sigma_P(t)$ for $P \in \{\mathbf{1}, \mathbf{2}\}$ that assign to each well-formed tree t over $A^{\sim}_{(i,k)}$ a P-signature $\sigma_P(t)$ such that:*

1. $\sigma_P(t) = \infty$ *if and only if P loses $\mathcal{G}(t)$;*
2. $\sigma_P\big({\sim}(t)\big) = (0, \ldots, 0)$ *if P wins $\mathcal{G}\big({\sim}(t)\big)$ (i.e. $\bar{P}$ wins $\mathcal{G}(t)$);*
3. $\sigma_P\big([j](t)\big) = (\theta_{i'}, \ldots, \theta_{j-1}, 0, 0 \ldots, 0)$ *if $\sigma_P(t) = (\theta_{i'}, \ldots, \theta_{k'})$ and j is P-winning;*
4. $\sigma_P\big([j](t)\big) = (\theta_{i'}, \ldots, \theta_{j-2}, \theta_j{+}1, 0, \ldots, 0)$ *if $\sigma_P(t) = (\theta_{i'}, \ldots, \theta_{k'})$ and j is P-losing;*
5. $\sigma_P\big(\langle P\rangle(t_{\mathtt{L}}, t_{\mathtt{R}})\big) = \min\big\{\sigma_P(t_{\mathtt{L}}), \sigma_P(t_{\mathtt{R}})\big\}$*;*
6. $\sigma_P\big(\langle \bar{P}\rangle(t_{\mathtt{L}}, t_{\mathtt{R}})\big) = \max\big\{\sigma_P(t_{\mathtt{L}}), \sigma_P(t_{\mathtt{R}})\big\}$.

Let us fix the functions σ_P for $P \in \{\mathbf{1}, \mathbf{2}\}$ as above. Consider a well-formed tree t over the alphabet $A^{\sim}_{(i,k)}$. We say that a strategy Σ of a player P in $\mathcal{G}(t)$ is *optimal* (or σ-optimal) if:

- In a position $u \in \mathrm{dom}(t)$ that is *kept* in t and $t{\restriction}u = \langle P\rangle\big(t_{\mathtt{L}}, t_{\mathtt{R}}\big)$ the strategy Σ moves to a subtree of a minimal value of σ_P; i.e. Σ can move to $u \cdot 0$ if $\sigma_P(t_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(t_{\mathtt{R}})$ and to $u \cdot 1$ if $\sigma_P(t_{\mathtt{L}}) \geq_{\mathrm{lex}} \sigma_P(t_{\mathtt{R}})$. If the values $\sigma_P(t_{\mathtt{L}})$ and $\sigma_P(t_{\mathtt{R}})$ are equal then Σ can move in any of the two directions.
- In a position u that is *switched* in t and $t{\restriction}u = \langle \bar{P}\rangle\big(t_{\mathtt{L}}, t_{\mathtt{R}}\big)$ the strategy Σ uses the same rule as above but uses the function $\sigma_{\bar{P}}$ to compare the $\bar{P}$-signatures of the subtrees.

Notice that according to the above definition there might be more than one optimal strategy.

▶ **Lemma 11.** *If $t \in \mathrm{W}^{\sim}_{P,(i,k)}$ and Σ is an optimal strategy of P in $\mathcal{G}(t)$ then Σ is winning.*

The following lemma claims the combinatorial core of this article: it shows that one can compare the P-signatures using a continuous reduction to the languages $\mathrm{W}^{\sim}_{P,(i,k)}$.

▶ **Lemma 12.** *There exists a continuous function $c_P\colon \big(\mathrm{Tr}_{A^{\sim}_{(i,k)}}\big)^2 \to \mathrm{Tr}_{A^{\sim}_{(i,k)}}$ such that if $t_{\mathtt{L}}$ and $t_{\mathtt{R}}$ are well-formed then so is $c_P(t_{\mathtt{L}}, t_{\mathtt{R}})$ and additionally*

$$c_P(t_{\mathtt{L}}, t_{\mathtt{R}}) \in \mathrm{W}^{\sim}_{\mathbf{1},(i,k)} \quad \textit{if and only if} \quad \sigma_P(t_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(t_{\mathtt{R}}).$$

The rest of this section demonstrates Proposition 9. Lemma 12 is proved in the next section.

Consider a function $f\colon \mathrm{Tr}_{A^{\sim}_{(i,k)}} \to \mathrm{Tr}_{A^{+\sim}_{(i,k)}}$ defined recursively as:

$$\begin{aligned} f\big(\langle P\rangle(t_{\mathtt{L}}, t_{\mathtt{R}})\big) &= \langle P+\rangle\Big(f(t_{\mathtt{L}}), f(t_{\mathtt{R}}), f\big(c_P(t_{\mathtt{L}}, t_{\mathtt{R}})\big)\Big) && \text{for } P \in \{\mathbf{1}, \mathbf{2}\},\\ f\big(\sim(t)\big) &= \sim\big(f(t)\big),\\ f\big([j](t)\big) &= [j]\big(f(t)\big) && \text{for } j \in \{i, \ldots, k\}. \end{aligned}$$

Clearly by the definition of f we know that $\mathrm{shave}\big(f(t)\big) = t$. Additionally, $f(t)$ is defined recursively using c_P which is continuous, therefore f is also continuous. As c_P maps well-formed trees to well-formed trees, so does f.

First assume that $f(t) \in L_{P,(i,k)}$ as witnessed by an accepting run ρ of $\mathcal{U}$ over $f(t)$ with $\rho(\epsilon) = (\star, P, \star)$. Fact 6 says that P wins $\mathcal{G}\big(\mathrm{shave}(f(t))\big) = \mathcal{G}(t)$, so $t \in \mathrm{W}^{\sim}_{P,(i,k)}$.

For the converse assume that P_0 wins $\mathcal{G}(t)$ for a well-formed tree t over $A^{\sim}_{(i,k)}$.

▶ **Lemma 13.** *If t is well-formed then there exists a unique run ρ of $\mathcal{U}$ over $f(t)$ such that for every $u \in \mathrm{dom}\big(f(t)\big)$ we have*

$$\rho(u) = (\star, P, \star) \quad \textit{if and only if} \quad P \textit{ wins } \mathcal{G}\big(\mathrm{shave}(f(t){\restriction}u)\big). \tag{1}$$

Moreover, all the ρ-strategies are winning for the respective players.

Proof. The construction of ρ is inductive from the root preserving (1). The only ambiguity when choosing transitions of $\mathcal{U}$ is when we reach a node $w \in \mathrm{dom}\big(f(t)\big)$ such that $\rho(w)$ is of the form $(\star, P, \star)$ and $f(t){\restriction}w = \langle P+\rangle\big(f(t_{\mathtt{L}}), f(t_{\mathtt{R}}), f(c_P(t_{\mathtt{L}}, t_{\mathtt{R}}))\big)$. We choose either the left or the right of the two lower transitions of $\mathcal{U}$ depending on the winner in $\mathcal{G}\big(\mathrm{shave}(f(t){\restriction}w \cdot 2)\big)$ in such a way to satisfy (1) for $u = w \cdot 2$. By the symmetry assume that we used the left transition. This leaves undeclared the second coordinate of $\rho(w \cdot 1)$ (resp. $\rho(w \cdot 0)$ in the case of the right transition). Again we declare this coordinate accordingly to (1). We need to check that (1) is also satisfied for $u = w \cdot 0$ (resp. $u = w \cdot 1$) i.e. that P wins $\mathcal{G}\big(\mathrm{shave}(f(t){\restriction}w \cdot 0)\big)$. To see that, we notice that the following conditions are equivalent (★):

- $\rho(w) = (\star, \star, \mathtt{L})$ [by the form of the transitions of $\mathcal{U}$]
- $\rho(w \cdot 2) = (\star, \mathbf{1}, \star)$ [by the definition of ρ]
- $\mathbf{1}$ wins in $\mathcal{G}\big(\mathrm{shave}(f(t){\restriction}(w \cdot 2))\big)$ [by the form of $f(t){\restriction}(w \cdot 2)$]
- $\mathbf{1}$ wins in $\mathcal{G}\big(\mathrm{shave}(f(c_P(t_{\mathtt{L}}, t_{\mathtt{R}})))\big)$ [by the equality $\mathrm{shave}\big(f(t')\big) = t'$]
- $\mathbf{1}$ wins in $\mathcal{G}\big(c_P(t_{\mathtt{L}}, t_{\mathtt{R}})\big)$ [by Lemma 12]
- $\sigma_P(t_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(t_{\mathtt{R}})$.

Thus, if we choose the lower left transition of $\mathcal{U}$, we know that $\sigma_P(t_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(t_{\mathtt{R}})$. By the inductive invariant we know that $\infty >_{\mathrm{lex}} \sigma_P\big(\mathrm{shave}(f(t){\restriction}w)\big) = \sigma_P\big(\langle P\rangle(t_{\mathtt{L}}, t_{\mathtt{R}})\big)$. Therefore, Item 5 of Lemma 10 implies that $\sigma_P(t_{\mathtt{L}}) <_{\mathrm{lex}} \infty$ so in fact P wins $\mathcal{G}(t_{\mathtt{L}}) = \mathcal{G}\big(\mathrm{shave}(f(t_{\mathtt{L}}))\big) = \mathcal{G}\big(\mathrm{shave}(f(t){\restriction}w \cdot 0)\big)$. Thus, the invariant (1) is also preserved for $u = w \cdot 0$. This concludes the inductive definition of ρ. Lemma 7 implies uniqueness.

Take $u \in \mathrm{dom}\big(f(t)\big)$ with $\rho(u) = (\star, P, \star)$. Let Σ be the ρ-strategy in u and $r' = f(t){\restriction}u$. Consider a node $w \in \mathrm{dom}\big(\mathrm{shave}(r')\big)$ such that $\mathrm{shave}(r')(w) = \langle P'\rangle$ with $P' = P$ if w is *kept* and $P' = \bar{P}$ if w is *switched*. In both cases $f(t)(u \cdot w) = \langle P'+\rangle$. By the above equivalence (★) and the definition of a ρ-strategy, Σ makes a σ-optimal move in w. Therefore, Σ is optimal. Invariant (1) says that P wins $\mathcal{G}\big(\mathrm{shave}(r')\big)$, so Lemma 11 implies that Σ is winning. ◀

Fix the run ρ given by the above lemma. Since $\mathcal{G}(t) = \mathcal{G}\big(\mathrm{shave}(f(t))\big)$, $\rho(\epsilon) = (\star, P_0, \star)$. Since all the ρ-strategies are winning, ρ is accepting by Lemma 5 and $f(t) \in L_{P,(i,k)}$. This concludes the proof of Theorem 1 assuming that Lemma 12 holds.

5 Comparing signatures

We will now prove Lemma 12 by defining, given two trees p and s over $A^{\sim}_{(i,k)}$, a game $\mathcal{C}_P(p,s)$. To indicate the difference between the games $\mathcal{C}_P$ and $\mathcal{G}$, we denote the players of $\mathcal{C}_P$ as $\exists = \mathbf{1}$ and $\forall = \mathbf{2}$. The purpose of the game $\mathcal{C}_P$ will be to ensure that $\exists$ wins $\mathcal{C}_P(p,s)$ if and only if $\sigma_P(p) \leq_{\mathrm{lex}} \sigma_P(s)$. The winning condition of the game $\mathcal{C}_P$ will be a parity condition, however, the game will allow certain *lookahead* (see steps (EL) and (AL) in the definition of $\mathcal{C}_P$). Then, the function c_P will just unravel the game $\mathcal{C}_P(p,s)$ into a tree over the alphabet $A^{\sim}_{(i,k)}$.

If i' and k' are the minimal and maximal P-losing numbers; j is a P-losing number; and $\sigma = (\theta_{i'}, \theta_{i'+2}, \ldots, \theta_{k'})$ is a P-signature then $\sigma{\restriction}j$ is the tuple $(\theta_{i'}, \theta_{i'+2}, \ldots, \theta_j)$. For completeness let $\infty{\restriction}j \stackrel{\mathrm{def}}{=} \infty$. Clearly $\sigma{\restriction}k' = \sigma$. Notice that $\sigma{\restriction}j$ is also a P-signature (with $k = j$) and moreover if $\sigma \leq_{\mathrm{lex}} \sigma'$ then $\sigma{\restriction}j \leq_{\mathrm{lex}} \sigma'{\restriction}j$.

A position of the game $\mathcal{C}_P$ is a triple (p,s,ℓ) where $p,s \in \mathrm{Tr}_{A^{\sim}_{(i,k)}}$ and ℓ is a P-losing number. As $\mathcal{C}_P(p,s)$ we denote the game $\mathcal{C}_P$ with the initial position set to (p,s,k'). The game is designed in such a way to guarantee the following claim.

▶ **Claim 14.** *A position (p,s,ℓ) is winning for $\exists$ in $\mathcal{C}_P$ if and only if*

$$\sigma_P(p){\restriction}\ell \;\leq_{\mathrm{lex}}\; \sigma_P(s){\restriction}\ell. \tag{2}$$

A single round of the game $\mathcal{C}_P$ consists of a sequence of choices done by the players. It is easy to encode such a sequence using additional intermediate positions of the game. For the sake of readability, we do not specify these positions explicitly. Instead, a round moves the game from a position (p,s,ℓ) into a new position according to the following sequential steps:

(EL) $\exists$ can claim that P loses $\mathcal{G}(s)$. If she does so, the game ends and $\exists$ wins iff $s \notin \mathrm{W}^{\sim}_{P,(i,k)}$.

(AL) $\forall$ can claim that P loses $\mathcal{G}(p)$. If he does so, the game ends and $\forall$ wins iff $p \notin \mathrm{W}^{\sim}_{P,(i,k)}$.

(EI) $\exists$ can modify ℓ into another P-losing number $\ell' < \ell$. In that case the round ends and the next position is (p', s, ℓ') where $p' = [\ell'](p)$.

(AI) $\forall$ can modify ℓ into another P-losing number $\ell' < \ell$. In that case the round ends and the next position is (p, s, ℓ').

(↓) If $p = [\ell](p')$ and $s = [\ell](s')$ then the round ends and the next position is (p', s', ℓ).

(↓p) If p is not of the form $[\ell](p'')$ then a step called *Step*$\exists(p)$ is done, resulting in an immediate win of $\exists$ or a new tree p'. The round ends and the next position is (p', s, ℓ).

(↓s) Otherwise p is of the form $[\ell](p')$ and a step called *Step*$\forall(s)$ is done, resulting in an immediate win of $\forall$ or a new tree s'. The round ends and the next position is (p, s', ℓ).

The result p' of *Step*$\exists(p)$ depends on the form of p as follows:

- If $p = \langle P\rangle(p_{\mathtt{L}}, p_{\mathtt{R}})$ then $\exists$ chooses to set $p' = p_{\mathtt{L}}$ or $p' = p_{\mathtt{R}}$.
- If $p = \langle \bar{P}\rangle(p_{\mathtt{L}}, p_{\mathtt{R}})$ then $\forall$ chooses to set $p' = p_{\mathtt{L}}$ or $p' = p_{\mathtt{R}}$.
- If $p = [j](p')$ and $j > \ell$ then p' is defined and that round of $\mathcal{C}_P$ has priority $j{+}1{-}P$.
- Otherwise $\exists$ immediately wins.

Dually, the result s' of *Step*$\forall(s)$ depends on the form of s as follows:

- If $s = \langle P\rangle(s_{\mathtt{L}}, s_{\mathtt{R}})$ then $\forall$ chooses to set $s' = s_{\mathtt{L}}$ or $s' = s_{\mathtt{R}}$.
- If $s = \langle \bar{P}\rangle(s_{\mathtt{L}}, s_{\mathtt{R}})$ then $\exists$ chooses to set $s' = s_{\mathtt{L}}$ or $s' = s_{\mathtt{R}}$.
- If $s = [j](s')$ with $j > \ell$ then s' is defined and that round of $\mathcal{C}_P$ has priority $j{-}2{+}P$.
- Otherwise $\forall$ immediately wins.

The rounds of $\mathcal{C}_P$ which priority is not declared above have priority k. An infinite play Π of $\mathcal{C}_P$ is won by $\exists$ if the least priority seen infinitely often during Π is even.

▶ **Lemma 15.** *The game $\mathcal{C}_P(p,s)$ can be unravelled as a tree $c_P(p,s)$ over the alphabet $A^{\sim}_{(i,k)}$ in such a way that for well-formed trees p, s, the tree $c_P(p,s)$ is well-formed and $\exists$ wins $\mathcal{C}_P(p,s)$ if and only if $\mathtt{1}$ wins $\mathcal{G}\big(c_P(p,s)\big)$. Moreover, the function c_P is continuous.*

Proof. Notice that in both cases when a round of $\mathcal{C}_P$ has an explicitly declared priority, that priority is j or $j{-}1$ with $i \leq \ell < j \leq k$. Therefore, the priorities of $\mathcal{C}_P$ are within $\{i,\ldots,k\}$.

The condition $s \notin \mathrm{W}^{\sim}_{\mathbf{1},(i,k)}$ from (EL) boils down to checking if $\mathtt{1}$ wins $\mathcal{G}\big(\sim(s)\big)$. Similarly, $s \notin \mathrm{W}^{\sim}_{\mathbf{2},(i,k)}$ if and only if $\mathtt{1}$ wins $\mathcal{G}\big(s\big)$. The same for the condition $p \in \mathrm{W}^{\sim}_{P,(i,k)}$ from (AL). Continuity and well-formedness follow directly from the definition. ◀

Notice that the rules of the game $\mathcal{C}_P$ do not allow to move from a position with a tree (either p or s) of the form $\sim(t)$ to a position with the respective tree being t. Therefore, we never need to swap the considered player P into $\bar{P}$.

Claim 14 together with Lemma 15 prove Lemma 12. Thus, the rest of this section is devoted to a proof of Claim 14. Since the winning condition of $\mathcal{C}_P$ is a parity condition, that game is positionally determined. Thus, to prove Claim 14 it is enough to show that none of the following two cases is possible for a position (p,s,ℓ) of $\mathcal{C}_P$:

- (2) is true and $\forall$ has a positional winning strategy $\Sigma_\forall$ from (p,s,ℓ),
- (2) is false and $\exists$ has a positional winning strategy $\Sigma_\exists$ from (p,s,ℓ).

In both cases we will confront the assumed strategy with a specially designed positional quasi-strategy of the opponent ($\Sigma^\star_\exists$ and $\Sigma^\star_\forall$ respectively). The quasi-strategy $\Sigma^\star_\exists$ will be defined only in positions that satisfy (2) and the quasi-strategy $\Sigma^\star_\forall$ in the remaining positions.

The quasi-strategy $\Sigma^\star_\exists$ (resp. $\Sigma^\star_\forall$) of a player $\exists$ (resp. $\forall$) in a round starting in a position $\big(p,s,\ell\big)$ performs the following choices in sub-rounds (EL) to (AI):

- In (EL) $\Sigma^\star_\exists$ claims that P loses $\mathcal{G}(s)$ if and only if he really does.
- In (AL) $\Sigma^\star_\forall$ claims that P loses $\mathcal{G}(p)$ if and only if he really does.
- In (EI) $\Sigma^\star_\exists$ modifies ℓ into ℓ' if $\ell' < \ell$ is the minimal P-losing number such that $\sigma_P(p)\restriction\ell' <_{\mathrm{lex}} \sigma_P(s)\restriction\ell'$. If there is no such number, $\Sigma^\star_\exists$ does not declare ℓ'.
- In (AI) $\Sigma^\star_\forall$ modifies ℓ into ℓ' if $\ell' < \ell$ is the minimal P-losing number such that $\sigma_P(p)\restriction\ell' >_{\mathrm{lex}} \sigma_P(s)\restriction\ell'$. If there is no such number, $\Sigma^\star_\forall$ does not declare ℓ'.

Moreover, in *Step*$\exists(p)$ when $p = \langle P\rangle(p_{\mathtt{L}}, p_{\mathtt{R}})$ the quasi-strategy $\Sigma^\star_\exists$ chooses to set $p' = p_{\mathtt{L}}$ if and only if $\sigma_P(p_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(p_{\mathtt{R}})$. Dually, in *Step*$\forall(s)$ when $s = \langle P\rangle(s_{\mathtt{L}}, s_{\mathtt{R}})$ the quasi-strategy $\Sigma^\star_\forall$ chooses to set $s' = s_{\mathtt{L}}$ if and only if $\sigma_P(s_{\mathtt{L}}) \leq_{\mathrm{lex}} \sigma_P(s_{\mathtt{R}})$.

Now it remains to define the choices of the quasi-strategies in the steps *Step*$\forall(s)$ and *Step*$\exists(p)$ when the given tree is of the form $\langle\bar{P}\rangle(t_{\mathtt{L}}, t_{\mathtt{R}})$. This is the place where the choices of $\Sigma^\star_\exists$ and $\Sigma^\star_\forall$ are not unique and that is why these are quasi-strategies.

▶ **Definition 16.** The quasi-strategies $\Sigma^\star_\exists$ and $\Sigma^\star_\forall$ need to satisfy the following *preservation guarantees*. First, in *Step*$\forall(s)$ when $s = \langle\bar{P}\rangle(s_{\mathtt{L}}, s_{\mathtt{R}})$ then $\Sigma^\star_\exists$ can set s' as any of the two $s_{\mathtt{L}}$, $s_{\mathtt{R}}$ that satisfies $\sigma_P(s')\restriction\ell \geq_{\mathrm{lex}} \sigma_P(p)\restriction\ell$. Second, in *Step*$\exists(p)$ when $p = \langle\bar{P}\rangle(p_{\mathtt{L}}, p_{\mathtt{R}})$ then $\Sigma^\star_\forall$ can set p' as any of the two $p_{\mathtt{L}}$, $p_{\mathtt{R}}$ that satisfies $\sigma_P(p')\restriction\ell >_{\mathrm{lex}} \sigma_P(s)\restriction\ell$.

▶ **Fact 17.** *In both cases the preservation guarantee leaves at least one possible choice.*

▶ **Lemma 18.** *Consider a position $\big(p,s,\ell\big)$. If it satisfies* (2) *and $\exists$ follows her quasi-strategy $\Sigma^\star_\exists$ then either she immediately wins or the next position also satisfies* (2)*.*

Dually, if the position violates (2) *and $\forall$ follows his quasi-strategy $\Sigma^\star_\forall$ then either he immediately wins or the next position also violates* (2)*.*

Notice that each play of $\mathcal{C}_P$ can modify the value of ℓ only bounded number of times. Moreover, because of the conditions in steps ($\downarrow$), ($\downarrow p$), and ($\downarrow s$) we obtain the following fact.

▶ **Fact 19.** *If Π is an infinite play of $\mathcal{C}_P$ then exactly one of the following three cases holds:*

- *Π makes infinitely many ($\downarrow$) steps,*
- *from some point on Π makes only ($\downarrow p$) steps,*
- *from some point on Π makes only ($\downarrow s$) steps.*

Observe that each step of $\mathcal{C}_P$ of the form $\mathit{Step}\exists(p)$ or $\mathit{Step}\forall(s)$ (if it doesn't mean an immediate win) simulates in fact a round of the game $\mathcal{G}(p)$ and $\mathcal{G}(s)$ respectively. Moreover, the quasi-strategies of the players $\exists$ and $\forall$ simulate optimal strategies of P in these rounds respectively. Thus, P must win these plays, as expressed by the following lemma.

▶ **Lemma 20.** *Consider an infinite play consistent with the quasi-strategy of one of the players ($\Sigma^\star_\exists$ or $\Sigma^\star_\forall$). Then the play only finitely many times makes the ($\downarrow$) step.*

Moreover, if the play follows $\Sigma^\star_\exists$ and from some point on makes only ($\downarrow p$) steps then it is winning for $\exists$. Similarly, if the play follows $\Sigma^\star_\forall$ and from some point on makes only ($\downarrow s$) steps then it is winning for $\forall$.

Proof. First take an infinite play Π of the quasi-strategy $\Sigma^\star_\exists$ of $\exists$ starting from a position (p_0, s_0, ℓ_0). The subtrees of the tree p_0 seen during Π follow a (possibly finite) play α of an optimal strategy of P in $\mathcal{G}(p_0)$. Since Π is infinite, $\exists$ does not declare that P loses $\mathcal{G}(s_0)$. Therefore, $\sigma_P(s_0) <_{\mathrm{lex}} \infty$ and by (2) we know that also $\sigma_P(p_0) <_{\mathrm{lex}} \infty$ which implies that $p_0 \in \mathrm{W}^{\sim}_{P,(i,k)}$. By Lemma 11 the play α in $\mathcal{G}(p_0)$ follows a winning strategy of P.

We will show that the step ($\downarrow$) occurs only finitely many times in Π. Assume contrarily and let ℓ be the minimal number ℓ that occurs in the play Π. Using the above assumptions, we know that ℓ is the minimal priority that is seen in the tree p_0 infinitely many times on α. Therefore, the simulated play α in $\mathcal{G}(p_0)$ is infinite and losing for P, which is a contradiction.

Therefore, by Fact 19 the play Π either makes from some point on only ($\downarrow p$) steps, or from some point on only ($\downarrow s$) steps. In the first case it follows the play α of $\mathcal{G}(p_0)$ that is winning for P. By the choice of priorities in the step $\mathit{Step}\exists(p)$ we know that $\exists$ wins Π.

The case of the quasi-strategy $\Sigma^\star_\forall$ of $\forall$ is entirely dual: we use the assumption that (2) is violated to know that $\sigma_P(s) <_{\mathrm{lex}} \infty$ so $s_0 \in \mathrm{W}^{\sim}_{P,(i,k)}$. Moreover, the choice of priorities in the step $\mathit{Step}\forall(s)$ implies that if Π makes $\mathit{Step}\forall(s)$ infinitely many times then $\forall$ wins Π. ◀

Now we move to the proof of the first case we need to exclude, i.e. that a position (p, s, ℓ) of $\mathcal{C}_P$ satisfies (2) but $\forall$ has a positional winning strategy $\Sigma_\forall$ from that position. We will prove that such a case is not possible. The second case is dual and the proof is analogous.

By Lemma 18, the positional quasi-strategy $\Sigma^\star_\exists$ always stays within positions satisfying the invariant (2). Moreover, the quasi-strategy never reaches a position that is immediately losing for $\exists$. Similarly, $\Sigma_\forall$ never reaches a position that is immediately losing for $\forall$. Thus, all the plays consistent with both $\Sigma^\star_\exists$ and $\Sigma_\forall$ must be infinite.

Notice that the values of ℓ are non-increasing during the plays of $\mathcal{C}_P$ and therefore, there exists a position that belongs to both $\Sigma^\star_\exists$ and $\Sigma_\forall$ such that the value ℓ stays constant during all the plays from that position. Without loss of generality take this as the starting position.

We can now proceed inductively in the tree obtained by unravelling the intersection of $\Sigma^\star_\exists$ and $\Sigma_\forall$: whenever the currently considered subtree contains anywhere a ($\downarrow$) step, we change the initial position to the result of that step. By Lemma 20 no play consistent with $\Sigma^\star_\exists$ takes the ($\downarrow$) step infinitely many times. Therefore, our inductive procedure has to stop at some point with no ($\downarrow$) steps in the current subtree. Without loss of generality we can assume that the initial position (p_0, s_0, ℓ_0) is the last position from the procedure. We know that the plays consistent with both $\Sigma^\star_\exists$ and $\Sigma_\forall$ never take the ($\downarrow$) step nor modify $\ell = \ell_0$.

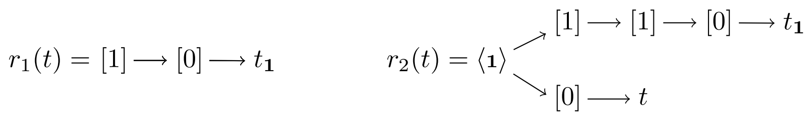

Figure 3 The pair of trees being the result of the reduction $r(t)$ from Proposition 22.

The structure of $\mathcal{C}_P$ guarantees that since the step ($\downarrow$) is not allowed, each play consistent with both $\Sigma_\exists^\star$ and $\Sigma_\forall$ takes only ($\downarrow p$) steps or takes only ($\downarrow s$) steps. Lemma 20 implies that in the former case the play would be winning for $\exists$, contradicting the assumption that $\Sigma_\forall$ is winning. Thus, all the considered plays take only ($\downarrow s$) steps. In particular $p = p_0$ is constant.

The intersection of $\Sigma_\forall$ and $\Sigma_\exists^\star$ induces a partial strategy Σ_P of P in $\mathcal{G}(s_0)$—Σ_P is partial because it does not contain positions that cannot be reached by following $\Sigma_\exists^\star$ because of the preservation guarantees, see Definition 16. The subtrees s' of s_0 in such unreachable positions satisfy $\sigma_P(s')\restriction\ell_0 <_{\text{lex}} \sigma_P(p_0)\restriction\ell_0$ by the definition of $\Sigma_\exists^\star$. In the positions on which Σ_P is defined it never visits a priority j with $j \leq \ell_0$ nor a node labelled $\sim$ because such a move is immediately losing for $\forall$ in *Step*$\forall(s)$. Because of the choice of the priorities in *Step*$\forall(s)$ and since $\Sigma_\forall$ is winning, Σ_P is winning for P on infinite plays.

Notice that since we take only ($\downarrow s$) steps, p_0 must be of the form $[\ell_0](p_0')$. Therefore, $\sigma_P(p_0)\restriction\ell_0 = (\theta_{i'}, \ldots, \theta_{\ell_0} + 1)$ for $(\theta_{i'}, \ldots, \theta_{\ell_0}) \stackrel{\text{def}}{=} \sigma_P(p_0')\restriction\ell_0$. It means that whenever the partial strategy Σ_P cannot reach a position with a subtree s', we know that in fact $\sigma_P(s')\restriction\ell_0 \leq_{\text{lex}} (\theta_{i'}, \ldots, \theta_{\ell_0})$. The following lemma says that the existence of such a partial strategy Σ_P witnesses the inequality $\sigma_P(s_0)\restriction\ell_0 \leq_{\text{lex}} (\theta_{i'}, \ldots, \theta_{\ell_0})$. By the definition of the ordinals θ_j we know that $(\theta_{i'}, \ldots, \theta_{\ell_0}) <_{\text{lex}} \sigma_P(p_0)\restriction\ell_0$, what contradicts (2) for (p_0, s_0, ℓ_0).

▶ **Lemma 21.** *Let $P \in \{\mathbf{1}, \mathbf{2}\}$, $t \in \mathrm{W}^{\sim}_{P,(i,k)}$, i' be the minimal P-losing number, and ℓ be some P-losing number. Assume that $(\theta_{i'}, \theta_{i'+2}, \ldots, \theta_\ell)$ is a tuple of ordinals and Σ_P is a partial strategy of the player P in $\mathcal{G}(t)$ such that:*

- *Σ_P never reaches a node u with $t(u) = [j]$ with $j \leq \ell$ nor a node u with $t(u) = \sim$,*
- *infinite plays of Σ_P are winning for P,*
- *if a position $u \in \mathrm{dom}(t)$ is not reachable by Σ then $\sigma_P\big(t\restriction u\big)\restriction\ell \leq_{\text{lex}} (\theta_{i'}, \ldots, \theta_\ell)$.*

Under all these assumptions $\sigma_P(t)\restriction\ell \leq_{\text{lex}} (\theta_{i'}, \ldots, \theta_\ell)$.

This finishes the proof of Claim 14. We conclude this section with a simple argument showing that a similar reduction c does not exist when we disallow the swapping symbol $\sim$, as expressed by the following proposition.

▶ **Proposition 22.** *There is no continuous function $c' \colon \big(\mathrm{Tr}_{A_{(0,k)}}\big)^2 \to \mathrm{Tr}_{A_{(0,k)}}$ such that*

$$c'(t_{\mathrm{L}}, t_{\mathrm{R}}) \in \mathrm{W}_{\mathbf{1},(0,k)} \quad \text{if and only if} \quad \sigma_{\mathbf{1}}(t_{\mathrm{L}}) \leq_{\text{lex}} \sigma_{\mathbf{1}}(t_{\mathrm{R}}).$$

Proof. Assume that such a function c' exists. Fix a tree $t_{\mathbf{1}} \in \mathrm{W}_{\mathbf{1},(0,k)}$ and consider a function $r \colon \mathrm{Tr}_{A_{(0,k)}} \to \big(\mathrm{Tr}_{A_{(0,k)}}\big)^2$ defined as $r(t) = \big(r_1(t), r_2(t)\big)$ for the pair of trees from Figure 3. Clearly r is continuous. Let $t \in \mathrm{Tr}_{A_{(0,k)}}$ and $r(t) = (t_{\mathrm{L}}, t_{\mathrm{R}})$. Notice that $\sigma_{\mathbf{1}}(t_{\mathrm{L}}) = (1, 0, \ldots)$. The value $\sigma_{\mathbf{1}}(t_{\mathrm{R}})$ is either $(0, 0, \ldots)$ if $t \in \mathrm{W}_{\mathbf{1},(0,k)}$ or $(2, 0, \ldots)$ otherwise. Therefore, $c'\big(r(t)\big) \in \mathrm{W}_{\mathbf{1},(0,k)}$ iff $\sigma_{\mathbf{1}}(t_{\mathrm{L}}) \leq_{\text{lex}} \sigma_{\mathbf{1}}(t_{\mathrm{R}})$ iff $t \notin \mathrm{W}_{\mathbf{1},(0,k)}$ iff $t \in \mathrm{W}_{\mathbf{2},(0,k)}$. Thus, $c' \circ r \colon \mathrm{Tr}_{A_{(0,k)}} \to \mathrm{Tr}_{A_{(0,k)}}$ is a continuous reduction of $\mathrm{W}_{\mathbf{2},(0,k)}$ to $\mathrm{W}_{\mathbf{1},(0,k)}$. This is a contradiction with [2, Lemma 1] (the assumption of contractivity is redundant there by Lemma 2 from the same paper). ◀

6 Conclusions

The main result of this work is the construction of the languages $L_{P,(i,k)}$ that solve the question of index bounds for unambiguous languages. Although the construction is not direct and relies heavily on an involved theory of signatures, these complications seem to be unavoidable when one wants to recognise languages like $W_{i,k}$ in an unambiguous way.

The definition of signatures given in the paper seems to be the canonical one, as witnessed by the point-wise minimality from Lemma 10. The previous ways of using signatures were mainly focused on their monotonicity and well-foundedness, thus it was enough to assume inequalities in the invariants of Lemma 10. Here, we are interested in comparing their actual values, therefore we insist on preserving these values via equalities.

References

1 André Arnold. The mu-calculus alternation-depth hierarchy is strict on binary trees. *ITA*, 33(4/5):329–340, 1999.
2 André Arnold and Damian Niwiński. Continuous separation of game languages. *Fundamenta Informaticae*, 81(1-3):19–28, 2007.
3 Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, pages 81–100, 2013.
4 Julian C. Bradfield. Simplifying the modal mu-calculus alternation hierarchy. In *STACS*, pages 39–49, 1998.
5 Arnaud Carayol and Christof Löding. MSO on the infinite binary tree: Choice and order. In *CSL*, pages 161–176, 2007.
6 Arnaud Carayol, Christof Löding, Damian Niwiński, and Igor Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Cent. Europ. J. of Math.*, 8:662–682, 2010.
7 Thomas Colcombet. Forms of determinism for automata. In *STACS*, pages 1–23, 2012.
8 Jacques Duparc, Kevin Fournier, and Szczepan Hummel. On unambiguous regular tree languages of index (0, 2). In *CSL*, pages 534–548, 2015.
9 Allen Emerson and Charanjit Jutla. Tree automata, mu-calculus and determinacy. In *FOCS'91*, pages 368–377, 1991.
10 Yuri Gurevich and Saharon Shelah. Rabin's uniformization problem. *J. Symb. Log.*, 48(4):1105–1119, 1983.
11 Szczepan Hummel. Unambiguous tree languages are topologically harder than deterministic ones. In *GandALF*, pages 247–260, 2012.
12 Szczepan Hummel. *Topological Complexity of Sets Defined by Automata and Formulas.* PhD thesis, University of Warsaw, 2017.
13 Alexander S. Kechris. *Classical descriptive set theory.* Springer-Verlag, New York, 1995.
14 Andrzej W. Mostowski. Games with forbidden positions. Technical report, University of Gdańsk, 1991.
15 Damian Niwiński and Igor Walukiewicz. Ambiguity problem for automata on infinite trees. unpublished, 1996.
16 Richard E. Stearns and Harry B. Hunt. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal of Computing*, 14(3):598–611, 1985.
17 Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81(3):249–264, 1989.
18 Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

19 Wolfgang Thomas and Helmut Lescow. Logical specifications of infinite computations. In *REX School/Symposium*, pages 583–621, 1993.

20 Igor Walukiewicz. Pushdown processes: Games and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, pages 62–74, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

21 Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275(1–2):311–346, 2002.