

On the topological complexity of weakly recognizable tree languages

Jacques Duparc¹ and Filip Murlak^{2*}

¹ Université de Lausanne, Switzerland

`jduparc@unil.ch`

² Warsaw University, Poland

`fmurlak@mimuw.edu.pl`

Abstract. We show that the family of tree languages recognized by weak alternating automata is closed by three set theoretic operations that correspond to sum, multiplication by ordinals $< \omega^\omega$, and pseudo-exponentiation with the base ω_1 of the Wadge degrees. In consequence, the Wadge hierarchy of weakly recognizable tree languages has the height of at least ε_0 , that is the least fixed point of the exponentiation with the base ω .

1 Introduction

Topological hierarchies stormed into the theory of formal languages with Klaus Wagner’s fundamental works on regular ω -languages [16, 17]. The incredible coincidence of the Wadge hierarchy and the index hierarchy for these languages encouraged further investigation of the Wadge hierarchies of wider classes of ω -languages, corresponding to more powerful recognizing devices: push-down automata and Turing machines [3, 4, 13]. It was only a matter of time before the same questions were asked for tree languages. Deterministic languages, an acclaimed “easy” subclass, were considered first. Albeit more complex, they admitted a number of techniques developed for ω -languages. Soon, the Borel hierarchy, the Wadge hierarchy, and the index hierarchy of deterministic tree languages were described and proved decidable [7–10].

The real challenge seems to be nondeterminism. The power it gives to tree automata makes them extremely difficult to tackle. Therefore, the investigation has basically concentrated on a very special sub-case – weakly recognizable languages. This class is the intersection of Büchi and co-Büchi languages [6, 12], so it is a rather small subclass of all regular tree languages. In fact, it does not even contain all deterministic languages. On the other hand, it captures some real nondeterminism, as it contains a lot of languages that cannot be recognized by deterministic automata: Skurczyński showed that weakly recognizable languages can have any finite Borel rank [14], while deterministic languages are either Π_1^1 -complete or are in Π_3^0 [9].

* The second author was supported by Polish government grant no. N206 005 31/0881.

A part of this work was done during the author’s visit at the University of Lausanne, Switzerland, financed by AutoMathA (ESF Short Visit Grant 1410).

More precisely, Skurczyński gave examples of Π_n^0 and Σ_n^0 -complete languages recognized by weak alternating automata using ranks $[0, n]$ and $[1, n + 1]$ accordingly. In this paper we extend this result by showing that weak automata using ranks $[0, n]$ can only recognize Π_n^0 languages, and dually $[1, n + 1]$ -automata can only recognize Σ_n^0 languages. (One may conjecture that the converse also holds, i. e., every weakly recognizable Π_n^0 language can be recognized by a $[0, n]$ -automaton, and dually for the additive classes.) We actually prove a bit stronger result. We consider so called weak game languages $W_{[\iota, \kappa]}$, to which all languages recognized by weak $[\iota, \kappa]$ -automata can be reduced. We show that $W_{[0, n]} \in \Pi_n^0$ and $W_{[1, n+1]} \in \Sigma_n^0$ (by Skurczyński's results, they are hard for these classes). The languages $W_{[\iota, \kappa]}$ are natural weak counterparts of strong game languages considered lately by Arnold and Niwiński. The strong game languages also form a strict hierarchy, but they are all non-Borel [1].

The main result of this paper is a lower bound for the Wadge hierarchy of weakly recognizable languages. We show that weakly recognizable languages are closed by three set-theoretic operations corresponding to the sum, multiplication by ordinals $< \omega^\omega$ and pseudo-exponentiation with the base ω_1 of the Wadge degrees. As a consequence, the hierarchy has the height of at least ε_0 , which is the least fixpoint of the exponentiation with the base ω . Again, this should be contrasted with the height of the hierarchy of deterministic tree languages, which is as low as $(\omega^\omega)^3 + 3$.

2 Weak Alternating Automata

A *tree over Σ* is a partial function $t : X^* \dashrightarrow \Sigma$ with a prefix closed domain. For the purpose of this paper we call such trees *conciliatory*. We do that to remind the reader that we are working with the trees that may have infinite and finite branches. A tree t is *full* if $\text{dom } t = X^*$. A tree is called *binary* if $X = \{0, 1\}$. Let T_Σ denote the set of full binary trees over Σ , and let \tilde{T}_Σ be the set of all conciliatory binary trees over Σ . By $t.v$ we denote the subtree of t rooted in $v \in \text{dom } t$.

A *weak alternating automaton* $\mathcal{A} = \langle \Sigma, Q_\exists, Q_\forall, q_0, \delta, \text{rank} \rangle$ consists of a finite input alphabet Σ , a finite set of states Q partitioned into existential states Q_\exists and universal states Q_\forall , an initial state q_0 , a transition relation $\delta \subseteq Q \times \Sigma \times \{0, 1, \varepsilon\} \times Q$, and a priority function $\text{rank} : Q \rightarrow [\iota, \kappa]$, where $[\iota, \kappa]$ stands for $\{\iota, \iota + 1, \dots, \kappa\}$. The transitions of the automaton are usually written as $p \xrightarrow{\sigma, d} q$, instead of $(p, \sigma, d, q) \in \delta$.

The *run* of the automaton \mathcal{A} on a conciliatory input tree $t \in \tilde{T}_\Sigma$ is a finitely branching conciliatory tree ρ_t labeled with $Q \times \{0, 1, \varepsilon\}$. The root of the tree is labeled with (q_0, ε) . Suppose we have already labeled a node X of ρ_t . Let $(p_1, d_1), (p_2, d_2), \dots, (p_m, d_m)$ be the sequence of labels on the unique path leading from the root to X . Let $v = d_1 d_2 \dots d_m$, where the ε 's occurring in the sequence $d_1 d_2 \dots d_m$ are interpreted as empty words. If $v \notin \text{dom } t$, then X is a leaf in ρ_t . Otherwise, for each transition $p_m \xrightarrow{t(v), d} q$, add a child Y to the node X and label

it with (q, d) . Note that the number of children of each node can be bounded by $3|Q|$.

The reader should not be puzzled by the fact that leaves of ρ_t do not correspond to any nodes of t . This is a notorious inconvenience in automata on finite objects: the number of states visited always exceeds by one the number of letters read. Let us imagine that cutting off a subtree produces a stub, and this is where the leaves of ρ_t dwell.

The accepting runs are defined by means of a modified weak parity game. Let ρ be a run of \mathcal{A} . The game G_ρ is played by Adam and Eve on the tree ρ . They move a token along the edges of the tree, starting from the root. The move is always made by the owner of the node: if the current node is labeled with a state from Q_\exists , it is Eve who moves the token to the next node, otherwise it is Adam. The play is infinite, unless it reaches a leaf. A play is won by Eve if the maximum of the ranks of states seen on the labels of visited nodes is even. Note that classically, when a play is finite, the owner of the last position loses. Here, we give no special rules for finite plays: the highest rank decides.

A run ρ is *accepting* if Eve has a winning strategy in the game G_ρ . A tree t is *accepted by the automaton* if ρ_t is accepting. The *language recognized by the automaton*, $L(\mathcal{A})$, is the set of accepted trees. A language is *weakly recognizable* if it is recognized by a weak alternating automaton.

While our automata work on conciliatory trees, the classical automata work on full binary trees. Instead of $L(\mathcal{A})$ one considers $L^\omega(\mathcal{A}) = L(\mathcal{A}) \cap T_\Sigma$. In order to relate the two versions we have to disguise conciliatory trees to make them look full.

Consider $T_{\Sigma \cup \{s\}}$, where s stands for “skip”. For a tree $t \in T_{\Sigma \cup \{s\}}$ we will define a conciliatory tree $u(t)$, called the *undressing* of t . Informally, we want to omit the skips in a top-down manner. Suppose we are in a node v such that $t(v) = s$. We would like to ignore this node and replace it with the next one, but in case of trees we have two nodes to choose from: $v0$ and $v1$. Let us always choose $v0$. Another problem is that we may encounter an infinite sequence of s 's. This would keep us replacing the current node with its left child, and never get to a symbol different from s . In that case, the tree $u(t)$ simply does not contain this node. Now, let us see a formal definition. Let v be the first node *not* labeled with s on the leftmost path of t (if there is no such node, $u(t)$ is empty). For each $w \in \{0, 1\}^*$ consider two possibly infinite sequences:

- $v_0 = v, w_0 = w,$
- $v_{i+1} = v_i b, w_{i+1} = w'_i$ if $w_i = b w'_i$ and $t(v_i b) \neq s,$
- $v_{i+1} = v_i 0, w_{i+1} = w_i$ if $w_i = b w'_i$ and $t(v_i b) = s.$

If $w_n = \varepsilon$ for some n , then $w \in \text{dom } u(t)$ and $u(t)(w) = t(v_n)$. Otherwise, $w \notin \text{dom } u(t)$. For a conciliatory language L , define L_s as the set of trees that belong to L after undressing, i. e., $L_s = \{t \in T_{\Sigma \cup \{s\}} : u(t) \in L\}$.

An automaton \mathcal{A} can be transformed easily into \mathcal{A}' such that $L^\omega(\mathcal{A}') = (L(\mathcal{A}))_s$. Simply, whenever you see a node labeled with s , move deterministically to the left without changing the state. In other words, it is enough to add $\{q \xrightarrow{s,0} q : q \in Q\}$ to the transition relation of \mathcal{A} .

3 Games, Hierarchies, and Topology

Let us start this section with the definition of a *conciliatory version of the Wadge game* (see [2]). For any pair of conciliatory tree languages L, M the game $G_C(L, M)$ is played by Spoiler and Duplicator. Each player builds a tree, t_S and t_D respectively. In every round, first Spoiler adds a finite number of nodes to t_S and then Duplicator adds a finite number of nodes to t_D . Nodes added by Duplicator and Spoiler must be children of nodes previously added. Both players are allowed to skip, i. e., add no nodes to their trees. Duplicator wins the game if $t_S \in L \iff t_D \in M$. Note that the resulting trees are conciliatory: they may contain finite branches, or even be finite.

For conciliatory languages L, M we use the notation $L \leq_C M$ iff Duplicator has a winning strategy in the game $G_C(L, M)$. If $L \leq_C M$ and $M \leq_C L$, we will write $L \equiv_C M$. The *conciliatory hierarchy* is the order induced by \leq_C on the \equiv_C classes of conciliatory languages.

The *classical Wadge game* $G_W(L, M)$ is defined for languages of full infinite trees, therefore a restriction on the moves is needed. The players must add both child nodes under each node they had put in the previous round. Only Duplicator is allowed to skip, and he must not skip forever. He must make infinitely many real moves, so that the tree he constructs is full.

The classical Wadge games provide a well-known criterion for continuous reducibility. T_Σ and the space of ω -words over Σ are equipped with the standard Cantor-like topology. For trees it is induced by the metric

$$d(s, t) = \begin{cases} 2^{-\min\{|x| : x \in \{0,1\}^*, s(x) \neq t(x)\}} & \text{if } s \neq t, \\ 0 & \text{if } s = t. \end{cases}$$

L is *continuously reducible* (or *Wadge reducible*) to M , if there exists a continuous function φ such that $L = \varphi^{-1}(M)$. We will write $L \leq_W M$, if L is Wadge reducible to M . Similarly we define \equiv_W and $<_W$. The *Wadge hierarchy* is the order induced on \equiv_W classes of languages.

Lemma 1 (Wadge). *For $L, M \subseteq T_\Sigma$, $L \leq_W M$ iff Duplicator has a winning strategy in $G_W(L, M)$.*

The conciliatory hierarchy embeds naturally into the Wadge hierarchy by the mapping $L \mapsto L_s$. A strategy in one game can be translated easily to a strategy in the other: arbitrary skipping in $G_C(L, M)$ gives the same power as the s labels in $G_W(L_s, M_s)$.

Lemma 2. *For all conciliatory languages L and M , $L \leq_C M \iff L_s \leq_W M_s$.*

Recall that a language L is called *self dual* if it is equivalent to its complement L^c . The conciliatory hierarchy does not contain self dual languages: a strategy for Spoiler in $G_C(L, L^c)$ is to skip in the first round, and then copy Duplicator's moves. By the lemma above, L_s is non self dual in terms of ordinary Wadge reducibility. Altogether, this shows that the conciliatory languages correspond

to certain non self dual languages. Which ones? For sets of infinite words of the finite Borel ranks – all of them.

A conciliatory word language is simply $L \subseteq \Sigma^{\leq \omega} = \Sigma^* \cup \Sigma^\omega$, i. e. a set of finite or infinite words. As for trees, we define L_s as the set of words over $\Sigma \cup \{s\}$, such that when we ignore all the s we obtain a word (finite or infinite) from L . Obviously, Lemma 2 holds also for words, but – as we have already disclosed – we get much more than that.

Theorem 1. (*Duparc [2]*) *For every $L \subseteq \Sigma^\omega$ of finite Borel rank, L is non self dual iff there exists $F \subseteq \Sigma^*$ such that $L \equiv_W (F \cup L)_s$.*

In particular, an ω -language of finite Borel rank is non self dual iff it is Wadge equivalent to a disguised conciliatory set. From the theorem above it follows that this also holds for tree languages.

Corollary 1. *For every $L \subseteq T_\Sigma$ of finite Borel rank, L is non self dual iff $L \equiv_W C_s$ for some conciliatory language C .*

Proof. First, observe that L is Wadge equivalent to L_w , which is the set of sequences obtained by writing down the trees from L level by level from left to right. The “writing down” and its inverse are suitable continuous reductions. By Theorem 1, L_w is equivalent to $(L_w \cup F)_S$ for some set of finite words F . Now, we need a conciliatory tree language C , equivalent to $L_w \cup F$. For a conciliatory tree t let $\text{fixed}(t)$ denote the sequence obtained by writing down the tree level by level from left to right until the first missing node is found. Note that $\text{fixed}(t)$ is infinite iff t is a full tree. Let $C = \{t \in \tilde{T}_\Sigma : \text{fixed}(t) \in L_w \cup F\}$. The identity function reduces L to C_s , so $C_s \geq_W L \equiv_W L_w \cup F$. Let us prove the converse inequality.

We will consider a mixed game $G(C_s, (L_w \cup F)_s)$. (Formally, instead of $(L_w \cup F)_s$ one can take a Wadge equivalent language $T(L_w \cup F)$, consisting of trees which have the leftmost path in $(L_w \cup F)_s$.) A winning strategy for Duplicator is to undress on-line the tree t_S constructed by Spoiler and write it down level by level, from left to right. When Duplicator finds a missing node, he plays s until Spoiler plays the missing node. At the end of the play, $\text{fixed}(u(t_S)) = u(w_D)$. Hence, $t_S \in C_s \iff w_D \in (L_w \cup F)_s$. \square

Let us end this section by recalling the notion of the *Wadge degree*. Since the Wadge ordering is well-founded [5], one may proceed by induction:

- $d_W(\emptyset) = d_W(\emptyset^{\mathbb{C}}) = 1$,
- $d_W(L) = \sup\{d_W(M) + 1 : M \text{ is non self dual, } M <_W L\}$ for $L >_W \emptyset$.

The *conciliatory degree* of a language is defined analogously:

- $d_C(\emptyset) = d_C(\emptyset^{\mathbb{C}}) = 1$,
- $d_C(L) = \sup\{d_C(M) + 1 : M <_C L\}$ for $L >_C \emptyset$.

By Corollary 1, for conciliatory L such that L_s has finite Borel rank, $d_C(L) = d_W(L_s)$. This observation lets us work with the conciliatory hierarchy instead of the Wadge hierarchy, as long as we restrict ourselves to non self dual sets of finite Borel ranks.

4 Up and Down the Hierarchy

In this section we present a handful of set-theoretical operations. Four of them will be the main tools in the remaining of the paper.

First, note that the choice of the alphabet Σ is of no importance. Let Σ and Σ' be finite alphabets containing at least two letters. For any language L over Σ (conciliatory or not), one can find an equivalent language L' over Σ' . Furthermore, if L is recognized by an automaton, so can be L' , and the construction of the new automaton is effective. Therefore, without loss of generality we may assume $\Sigma = \{a, b\}$.

For $L, M \subseteq \tilde{T}_\Sigma$ define $M + L$ as the set of trees $t \in \tilde{T}_\Sigma$ satisfying any of the following conditions:

- $t.0 \in L$ and $t(10^n) = a$ for all n ,
- 10^n is the first node on the path 10^* labeled with b and either $t(10^n 0) = a$ and $t.10^n 00 \in M$ or $t(10^n 0) = b$ and $t.10^n 00 \in M^{\mathcal{C}}$.

When playing a conciliatory Wadge game, being in charge of the language $M + L$ is like being in charge of L with one extra move that erases everything played so far and replaces L with M or $M^{\mathcal{C}}$. This move can be played only once during the play, and is executed by playing b on the path 10^* for the first time. By choosing the next letter on this path we make choice between M and $M^{\mathcal{C}}$.

The next operation is a generalization of the previous one. It lets a player choose from a countable collection of languages. Let $L_n \subseteq \tilde{T}_\Sigma$ for $n < \omega$. Define $\sup_{n < \omega} L_n$ as the set of trees $t \in \tilde{T}_\Sigma$ satisfying one of the following conditions:

- $t(1^n) = a$ for all n ,
- 1^n is the first node on 1^* labeled with b and $t.1^n 0 \in L_n$.

Note that $\sup_{n < \omega} L_n$ is conciliatory, even if the languages L_n contain only full trees.

The multiplication by countable ordinals is defined as an iterated sum:

- $L \cdot 1 = L$,
- $L \cdot (\alpha + 1) = L \cdot \alpha + L$,
- $L \cdot \lambda = \sup_{\gamma < \lambda} L \cdot \gamma$ for limit ordinals λ .

Finally let us define the pseudo-exponentiation. Let $L \subseteq \tilde{T}_\Sigma$. For $t \in \tilde{T}_\Sigma$ let

$$i(t)(a_1 a_2 \dots a_n) = \begin{cases} t(a_1 0 a_2 \dots 0 a_n 0) & \text{if } \forall_k t(a_1 0 a_2 \dots 0 a_n 1^k) = a \\ s & \text{if } \exists_k t(a_1 0 a_2 \dots 0 a_n 1^k) = b \end{cases} .$$

Intuitively, the rightmost path starting in $a_1 0 a_2 \dots 0 a_n$ tells us whether to skip the node $a_1 0 a_2 \dots 0 a_n 0$ or not. Let

$$\exp L = \{t \in \tilde{T}_\Sigma : u(i(t)) \in L\} .$$

A player in charge of $\exp L$ is like a player in charge of L with an extra possibility to decide that a chosen node labeled in the past (and the subtree rooted in its right child) is to be ignored.

The names of the operations and the notation used make the following theorem rather expected.

Theorem 2 (Duparc [2]). *For $L, M \subseteq \tilde{T}_\Sigma$, L_s, M_s Borel of finite rank, and a countable ordinal α it holds that $d_C(L + M) = d_C(L) + d_C(M)$, $d_C(L \cdot \alpha) = d_C(L) \cdot \alpha$ and $d_C(\exp L) = \omega_1^{d_C(L)+\varepsilon}$, where*

$$\varepsilon = \begin{cases} -1 & \text{if } d_C(L) < \omega \\ 0 & \text{if } d_C(L) = \beta + n \text{ and } \text{cof} \beta = \omega_1 \\ +1 & \text{if } d_C(L) = \beta + n \text{ and } \text{cof} \beta = \omega \end{cases} .$$

A kind of inverse operation for $\exp L$ was introduced in [2]. The operation relies on auxiliary moves and involves games where players must, along with playing letters as usual, answer questions about the future of the play. One may easily imagine that when the opponent asks questions like this, it may be much more difficult to win. For convenience, we describe this operation on infinite words, but it can easily be modified to apply to infinite trees (via the usual correspondence between infinite sequences and trees with finite branching).

Let us define the space in which the player evolves when answering questions about the future of the play. We call such a space a *question tree*. We will be working with non-labeled trees, which are simply prefix closed subsets of X^* for a fixed set X . A tree is *pruned* if it has no finite branches. For a finite or infinite word $x = x_0x_1x_2\dots$ we write $\frac{x}{2}$ for the word $x_0x_2x_4\dots = \langle x_{2i} : i < |x|/2 \rangle$.

Definition 1. *A question tree related to an alphabet Σ is a non-labeled non-empty pruned tree $\mathcal{T} \subseteq (\Sigma')^\omega$, with $\Sigma' = \Sigma \cup \{!w : w \in \Sigma\} \cup \{?\}$, satisfying for every node $u \in \mathcal{T}$:*

if $|u|$ is even, then $u = v\sigma$ for some $\sigma \in \Sigma$ (these nodes correspond to the main play),

if $|u|$ is odd, then it is an auxiliary move with two different kinds of options:

$\langle ! \rangle$ $u = v \langle !w \rangle$ for some $w \in \Sigma^*$ extending $\frac{u}{2}$. In this case we demand that any node $u' \in \mathcal{T}$ extending u satisfies $\frac{u'}{2} \sqsubseteq w$ or $w \sqsubseteq \frac{u'}{2}$. Moreover, we require that for any $v \langle !w' \rangle \in \mathcal{T}$ either $w' = w$ or $w' \not\sqsubseteq w \wedge w \not\sqsubseteq w'$.

$\langle ? \rangle$ $u = v \langle ? \rangle$. This is an option to avoid all positions of the form $v \langle !w \rangle$. Formally, for each $u' \in \mathcal{T}$ extending $v \langle ? \rangle$ and each w such that $v \langle !w \rangle \in \mathcal{T}$, $w \not\sqsubseteq \frac{u'}{2}$.

Recall that $[\mathcal{T}]$ denotes the set of branches of \mathcal{T} . Notice that for every infinite word $x \in \Sigma^\omega$ there exists a unique infinite branch $y \in [\mathcal{T}]$ such that $\frac{y}{2} = x$.

In even moves a player moving along a question tree simply plays letters from Σ , just like in an ordinary play. In odd moves, everything looks like his opponent were asking him questions of the form: ‘‘Do you intend to stay forever in this closed subset of Σ^ω ? If you are willing to exit, let me know which of the

positions I submitted to you, you intend to reach.” So, the player can choose $\langle !v' \rangle$ and say: “OK, I’m going to reach this position v ”, or taking the option $\langle ? \rangle$ he may answer: “No, I won’t reach any of these, I will rather stay out of the open set formed by the union of the basic open sets $v\Sigma^\omega$ for all positions v you are asking me about.” Taking the latter option means remaining in the complement of this open set, hence in a closed subset of Σ^ω .

Definition 2. Let $L \subseteq \Sigma^\omega$ and \mathcal{T} be a question tree on Σ' related to Σ . $L^{\mathcal{T}}$ consists of $x \in (\Sigma')^\omega$ such that $x \in [\mathcal{T}] \wedge \frac{x}{2} \in L$ or $x \notin [\mathcal{T}] \wedge \exists x' (wx' \in [\mathcal{T}] \wedge \frac{wx'}{2} \in L)$ where w is the longest prefix of x that belongs to \mathcal{T} .

The definition of $L^{\mathcal{T}}$ may seem a bit awkward. We defined it this way so that Duplicator still has a winning strategy in the restricted version of the Wadge game $G_W(L^{\mathcal{T}}, L^{\mathcal{T}})$, where Duplicator is not allowed to exit the question tree \mathcal{T} , while Spoiler can play anything he wants. Hence, a player in charge of $L^{\mathcal{T}}$ in a Wadge game has a winning strategy if and only if he has a winning strategy that always remains inside \mathcal{T} . Therefore in the sequel we always assume that a strategy involving a set of the form $L^{\mathcal{T}}$ remains in the underlying question tree; in other words it restricts its moves to the legal ones.

For any $L \subseteq \Sigma^\omega$ and any question tree \mathcal{T} related to Σ , $L^{\mathcal{T}} \leq_W L$. In particular, whenever L is Borel, $L^{\mathcal{T}}$ is Borel too.

Definition 3. Let $L \subseteq \Sigma^\omega$ be Borel, $\log L$ stands for a $<_W$ -minimal element of the form $L^{\mathcal{T}}$ where \mathcal{T} ranges over all question trees related to Σ .

In [2] it is proved that for a fixed L all $<_W$ -minimal sets of the form $L^{\mathcal{T}}$ are Wadge equivalent. However, $L \mapsto \log L$ is an operation only if we make it functional, which requires the full Axiom of Choice. Since this functional character is not needed in the proofs, we insist on the fact that $\log L$ is just a notational convenience and none of the foregoing proofs involving it requires the Axiom of Choice.

The operation \log preserves the Wadge ordering.

Proposition 1. For $L, M \subseteq \Sigma^\omega$ Borel, $L \leq_W M \implies \log L \leq_W \log M$.

For the present application the key property of the operation \log is the following.

Proposition 2. Let $L \subseteq \Sigma^\omega$,

$$\begin{aligned} L \text{ is } \Pi_{n+1}^0\text{-complete} &\iff \log L \text{ is } \Pi_n^0\text{-complete,} \\ L \text{ is } \Sigma_{n+1}^0\text{-complete} &\iff \log L \text{ is } \Sigma_n^0\text{-complete.} \end{aligned}$$

Actually, the operation is even finer: it is a precise counterpart of the pseudo-exponentiation. We state the following result for tree languages despite the fact it was proved for word languages in [2]. The extension to tree languages is straightforward.

Proposition 3. *Given L a (full) tree language, and M a conciliatory tree language, with L, M_s both Borel,*

$$\begin{aligned} \log L \leq_W M_s &\iff L \leq_W (\exp M)_s, \\ M_s \leq_W \log L &\iff (\exp M)_s \leq_W L. \end{aligned}$$

5 Weak Index vs. Borel Rank

Fix a natural number N . Let us call a game-tree, a N -ary tree T whose nodes are boxes or diamonds equipped with ranks. For $\iota = 0, 1$ and $\kappa > \iota$ let $W_{[\iota, \kappa]}$ be the set of all game-trees T with ranks inside $[\iota, \kappa]$ and such that Eve has a winning strategy in the underlying *weak* parity game.

Theorem 3. *For each n , $W_{[0, n]} \in \Pi_n^0$ and $W_{[1, n+1]} \in \Sigma_n^0$.*

Proof. By the determinacy of parity games, $(W_{[0, n]})^c \equiv_W W_{[1, n+1]}$. Hence, it is enough to prove the claim for ranks inside $[0, n]$. The proof goes by induction with respect to n .

Let us first see that $W_{[0, 1]}$ is closed. Let t_k denote the restriction of a tree t to its k first levels. By Weak König Lemma, Adam has a winning strategy in t iff for some k , Adam has a winning strategy in t_k . This means that if a tree t does not belong to $W_{[0, 1]}$, then one already knows it after looking at some finite initial part of it. This is precisely the condition that defines closed sets.

Let \mathcal{T} be a question tree defined as follows. Given any game tree t with ranks inside $[0, n+1]$, for each node u in t which is the first node on this branch with priority n , \mathcal{T} asks the question whether Adam or Eve would have a winning strategy if the game were to start from this particular node. By what we have already proved, if n is even, “Eve has a winning strategy” is a closed condition and if n is odd, “Adam has a winning strategy” is a closed condition. Hence, the question above is legal.

Fix a tree T over $\Sigma \cup \{!w : w \in \Sigma\} \cup \{?\}$, where $\Sigma = \{\diamond, \square\} \times [0, n+1]$, which answers questions asked by \mathcal{T} (strictly speaking this means the one and only play inside \mathcal{T} which corresponds to T). Let us construct a tree t' over $\{\diamond, \square\} \times [0, n]$ which is exactly the same as T except that for each node u as above: if the answer is “Adam has a winning strategy”, every node in the subtree rooted in u receives the rank n for odd n 's and $n-1$ for even n 's, and if the answer is “Eve has a winning strategy” – the other way round. This gives a continuous reduction of $W_{[0, n+1]}^{\mathcal{T}}$ to $W_{[0, n]}$.

Hence, $\log W_{[0, n+1]} \leq_W W_{[0, n+1]}^{\mathcal{T}} \leq_W W_{[0, n]}$. By induction hypothesis we have $W_{[0, n]} \in \Pi_n^0$, and in consequence $W_{[0, n+1]} \in \Pi_{n+1}^0$. \square

As a corollary we get the promised improvement of Skurczyński's result [14].

Corollary 2. *For every weak alternating automaton with ranks inside $[0, n]$ (resp. $[1, n+1]$) it holds that $L^\omega(\mathcal{A}) \in \Pi_n^0$ (resp. $L^\omega(\mathcal{A}) \in \Sigma_n^0$).*

Proof. Let \mathcal{A} be an automaton with priorities inside $[\iota, \kappa]$. For sufficiently large N we may assume without loss of generality that the runs of the automaton are N -ary trees. By assigning to an input tree the run of \mathcal{A} , one obtains a continuous function reducing $L^\omega(\mathcal{A})$ to $W_{(\iota, \kappa)}$. Hence, the claim follows from the theorem above. \square

The results described in this section give yet another argument to one of the opposing parties in the everlasting dispute between the big-endians and the little-endians of game theory. Had we defined a play to be winning for Eve if the *lowest* rank was even, the correspondence between the indices and Borel classes would be rather ugly.

6 Three Simple Constructions

Let \mathcal{A}, \mathcal{B} be weak alternating tree automata. As it was explained in the beginning of Sect. 4, we may assume without loss of generality that the automata have the same input alphabet $\Sigma = \{a, b\}$. We will construct automata recognizing languages equivalent (in the conciliatory sense) to $L(\mathcal{B}) + L(\mathcal{A})$, $L(\mathcal{A}) \cdot \omega$, and $\exp L(\mathcal{A})$.

Sum. Consider the automaton $\mathcal{B} + \mathcal{A}$ defined on Fig. 1. The diamond states are existential and the box states are universal. The circle states can be treated as existential, but in fact they give no choice to either player. The transitions leading to \mathcal{A}, \mathcal{B} and \mathcal{B}^c should be understood as transitions to the initial states of the according automata. The priority functions of \mathcal{B} and \mathcal{B}^c might need shifting up, so that they were not using the value 0. It is easy to check that $L(\mathcal{B} + \mathcal{A}) = L(\mathcal{B}) + L(\mathcal{A})$.

Multiplication by ω . The automaton $\mathcal{A} \cdot \omega$ is shown on Fig. 1. The language recognized by $\mathcal{A} \cdot \omega$ consists of trees having no b 's on the path 1^* or satisfying the following conditions for some $0 < i \leq k$ and n :

- 1^k is the first node labeled with b on the path 1^* ,
- i is minimal such that for all $i < j \leq k$ the path $1^j 0^+$ contains no b 's,
- $1^i 0^n$ is the first node labeled with b lying on the path $1^i 0^+$,
- either $t(1^i 0^n 0) = a$ and $t.1^i 0^n 00 \in L(\mathcal{A})$ or $t(1^i 0^n 0) = b$ and $t.1^i 0^n 00 \in L(\mathcal{A})^c$.

Let L_k denote the set of trees satisfying the four conditions above for a fixed k . Observe that $L_k \equiv_W \emptyset + L(\mathcal{A}) \cdot k$. Intuitively, we cannot use the subtrees rooted in $10, 110, \dots, 1^k 0$ together, because making the first nontrivial move in the subtree rooted in $1^i 0$ (putting the first b on the path $1^i 0^+$) makes the subtrees rooted in $10, 110, \dots, 1^{i-1} 0$ irrelevant. The best we can do is to use them one by one, and this gives exactly the power of $\emptyset + L(\mathcal{A}) \cdot k$.

Now, consider $G_C(L(\mathcal{A}) \cdot \omega, L(\mathcal{A} \cdot \omega))$. By definition, $L(\mathcal{A}) \cdot \omega$ consists of trees having no b on the rightmost path, or such that 1^k is the first node on

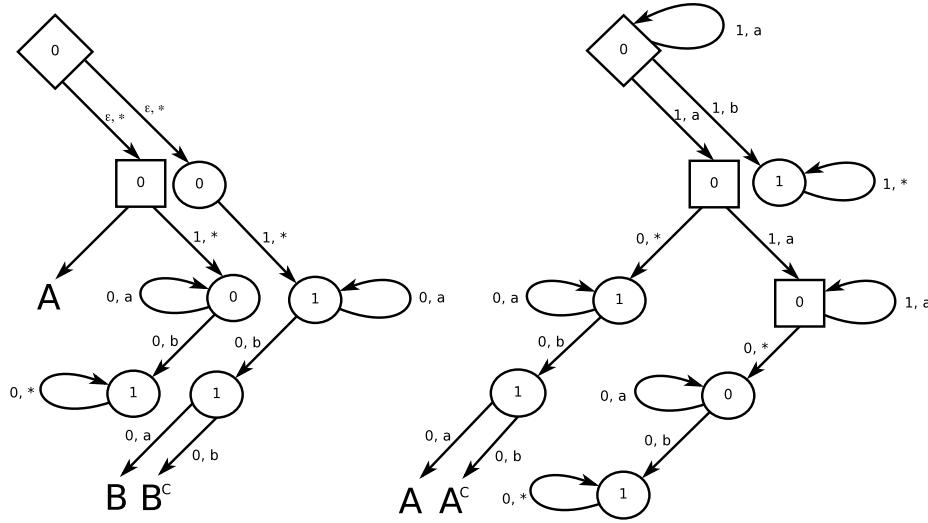


Fig. 1. The automata $\mathcal{B} + \mathcal{A}$ and $\mathcal{A} \cdot \omega$.

this path labeled with b , and $t.1^k0 \in L(\mathcal{A}) \cdot k$. Consider the following strategy for Duplicator. First, only observe the rightmost path of Spoiler's tree t_S . While Spoiler plays a 's, keep playing a 's in t_D (Duplicator's tree). If Spoiler never plays a b , Duplicator wins. Suppose Spoiler plays his first b in the node 1^k . Duplicator should also play b in the node 1^k . Now, the result of the play depends only on whether $t_S.1^k0 \in L(\mathcal{A}) \cdot k \iff t_D \in L_k$, and Duplicator should simply use the strategy from $G_C(L(\mathcal{A}) \cdot k, L_k)$.

In the game $G_C(L(\mathcal{A} \cdot \omega), L(\mathcal{A}) \cdot \omega)$ the only difference is that Duplicator should play one more a : if Spoiler plays the first b on the rightmost path in the node 1^k , then Duplicator should put his first b in 1^{k+1} , so that he can later use the strategy from $G_C(L_k, L(\mathcal{A}) \cdot (k + 1))$.

Pseudo-exponentiation. Both previous constructions were performed by combining two or three automata with a particularly chosen gadget. The automaton $\text{exp } \mathcal{A}$ is a bit more tricky. This time, we have to change the whole structure of the automaton. Instead of adding one gadget, we replace each state of \mathcal{A} by a different gadget.

The gadget for a state p is shown on Fig. 2. By replacing p with the gadget we mean that all the transitions ending in p should now end in p' and all the transitions starting in p should start in p'' . Note that the state p'' is the place where the original transition is chosen, so p'' should be existential iff p is existential. It is not difficult to see that $\text{exp } \mathcal{A}$ recognizes exactly $\text{exp } L(\mathcal{A})$.

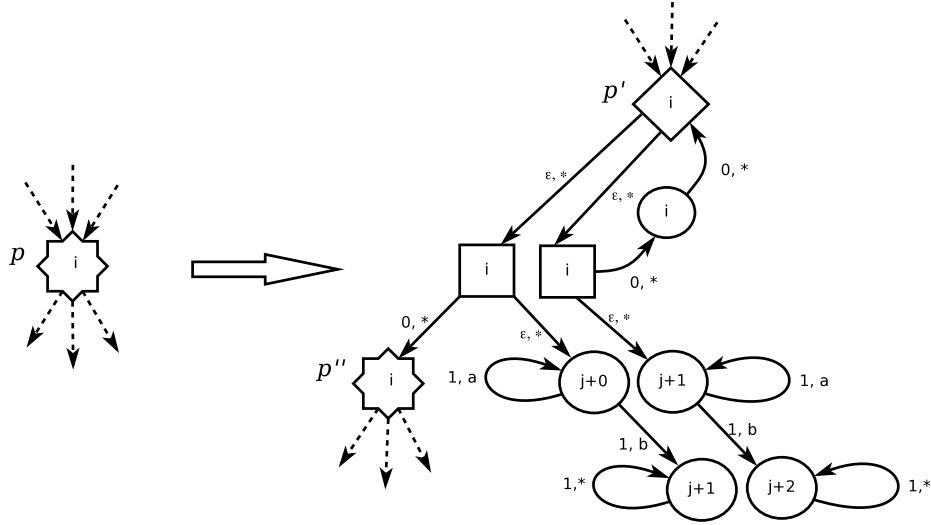


Fig. 2. The gadget to replace p in the construction of $\exp \mathcal{A}$. The state p'' is existential iff p is existential, $i = \text{rank } p$, and j is the least even number greater or equal to i .

7 A Lower Bound

In the previous section we have shown that weakly recognizable languages are closed by sum, multiplication by ω , and pseudo-exponentiation with the base ω_1 . By iterating finitely many times sum and multiplication by ω we obtain closure by multiplication by ordinals of the form $\omega^n k_n + \dots + \omega k_1 + k_0$, i.e., all ordinals less than ω^ω . In other words, we can find a weakly recognizable language of any conciliatory degree from the closure of $\{1\}$ by ordinal sum, multiplication by ordinals $< \omega^\omega$ and pseudo-exponentiation with the base ω_1 . It is easy to see that the order type of this set is not changed if we replace pseudo-exponentiation with ordinary exponentiation $\alpha \mapsto \omega_1^\alpha$. This in turn is isomorphic with the closure of $\{1\}$ by ordinal sum, multiplication by ordinals $< \omega^\omega$, and exponentiation with the base ω^ω . This last set is obviously ε_0 , the least fixpoint of the exponentiation with the base ω .

By Lemma 2 and the final remark of Sect. 2 it follows that the mapping $L \mapsto L_s$ embeds the conciliatory hierarchy of weakly recognizable languages into the Wadge hierarchy of weakly recognizable languages of full trees. Hence, we obtain the main result of this paper.

Theorem 4. *The Wadge hierarchy of weakly recognizable tree languages has the height of at least ε_0 .*

Our intuition tells us the bound is tight, but we have no evidence for that. The question of the exact height of the hierarchy for weak automata remains open.

Acknowledgments

The first author would like to express his gratitude to David Janin who incidentally made him initiate this study of the hierarchy of weak alternating tree automata. We also thank Damian Niwiński and the anonymous referees for very helpful comments.

References

1. A. Arnold, D. Niwiński. Continuous separation of game languages. Manuscript, submitted, 2006.
2. J. Duparc. Wadge hierarchy and Veblen hierarchy. Part I: Borel sets of finite rank. *The Journal of Symbolic Logic* **66** (2001).
3. J. Duparc. A hierarchy of deterministic context-free ω -languages. *Theoret. Comput. Sci.* **290** (2003) 1253–1300.
4. O. Finkel. Wadge Hierarchy of Omega Context Free Languages. *Theoret. Comput. Sci.* **269** (2001) 283–315.
5. A. S. Kechris. *Classical Descriptive Set Theory*. Graduate Texts in Mathematics Vol. 156, 1995.
6. D. E. Muller, A. Saoudi, P. E. Schupp. Alternating automata. The weak monadic theory of the tree, and its complexity. *Proc. ICALP '86*, LNCS 226 (1986), 275–283.
7. F. Murlak. On deciding topological classes of deterministic tree languages. *Proc. CSL'05*, LNCS 3634 (2005) 428–441.
8. F. Murlak. The Wadge hierarchy of deterministic tree languages. *Proc. ICALP '06*, Part II, LNCS 4052 (2006), 408–419.
9. D. Niwiński, I. Walukiewicz. A gap property of deterministic tree languages. *Theoret. Comput. Sci.* **303** (2003) 215–231.
10. D. Niwiński, I. Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Proc. WoLLiC 2004*, *Electronic Notes in Theoret. Comp. Sci.* 195–208, 2005.
11. D. Perrin, J.-E. Pin. *Infinite Words. Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics Vol. 141, Elsevier, 2004.
12. M. O. Rabin. Weakly definable relations and special automata. *Mathematical Logic and Foundations of Set Theory*, North-Holland, 1970, 1–70.
13. V. Selivanov. Wadge Degrees of ω -languages of deterministic Turing machines. *Theoret. Informatics Appl.* **37** (2003) 67–83.
14. J. Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoret. Comput. Sci.* **112** (1993) 413–418.
15. T. F. Urbański. On deciding if deterministic Rabin language is in Büchi class. *Proc. ICALP 2000*, LNCS 1853 (2000) 663–674.
16. K. Wagner. Eine topologische Charakterisierung einiger Klassen regulärer Folgenmengen. *J. Inf. Process. Cybern. EIK* **13** (1977) 473–487.
17. K. Wagner. On ω -regular sets. *Inform. and Control* **43** (1979), 123–177.