On the topological complexity of tree languages

André Arnold[1]
Jacques Duparc[2]
Filip Murlak[3]
Damian Niwiński[3]

[1] Laboratoire Bordelais de Recherche en Informatique
Université Bordeaux 1
351 cours de la Libération
33405 Talence cedex, France
andre.arnold@club-internet.fr

[2] École des Hautes Études Commerciales
Université de Lausanne
1015 Dorigny, Switzerland
Jacques.Duparc@unil.ch

[3] Institute of Informatics
Uniwersytet Warszawski
Banacha 2
02-097 Warszawa, Poland
{fmurlak,niwinski}@mimuw.edu.pl

Abstract

The article surveys recent results in the study of topological complexity of recognizable tree languages. Emphasis is put on the relation between topological hierarchies, like the Borel hierarchy or the Wadge hierarchy, and the hierarchies resulting from the structure of automata, as the Rabin-Mostowski index hierarchy. The topological complexity of recognizable tree languages is seen as an evidence of their structural complexity, which also induces the computational complexity of the verification problems related to automata, as the non-emptiness problem. Indeed, the topological aspect can be seen as a rudiment of the infinite computation complexity theory.

1 Introduction

Since the discovery of irrational numbers, the issue of impossibility has been one of the driving forces in mathematics. Computer science brings forward a related problem, that of difficulty. The mathematical expression of difficulty is complexity, the concept which affects virtually all subjects in computing science, taking on various contents in various contexts.

In this paper we focus on infinite computations, and more specifically on finite-state recognition of infinite trees. It is clearly not a topic of clas-

Erich Grädel, Jörg Flum, Thomas Wilke (eds.). *Logic and Automata: History and Perspectives.* Texts in Logic and Games 2, Amsterdam University Press 2007, pp. 9–29.

sical complexity theory which confines itself to computable functions and relations over integers or words, and measures their complexity by the—supposedly finite—time and space used in computation. However, infinite computations are meaningful in computer science, as an abstraction of some real phenomena as, e.g., interaction between an open system and its environment. The finite and infinite computations could be reconciliated in the framework of descriptive complexity, which measures difficulty by the amount of logic necessary to describe a given property of objects, were they finite or infinite. However the automata theory has also developed its own complexity measures which refer explicitly to the dynamics of infinite computations.

From yet another perspective, infinite words (or trees) are roughly the real numbers, equipped with their usual metric. Classification of functions and relations over reals was an issue in mathematics long before the birth of computer science. The history goes back to Émil Borel and the circle of semi-intuitionists around 1900, who attempted to restrict the mathematical universe to mentally constructible (*définissables*) objects, rejecting set-theoretic pathologies as unnecessary. This program was subsequently challenged by a discovery made by Mikhail Suslin in 1917: the projection of a Borel relation may not be Borel anymore (see [Mos80], but also [Add04] for a brief introduction to definability theory). It is an intriguing fact that this phenomenon is also of interest in automata theory. For example, the set of trees recognized by a finite automaton may be non-Borel, even though the criterion for a path being successful is so. One consequence is that the Büchi acceptance condition is insufficient for tree automata.

Classical theory of definability developed two basic topological hierarchies: Borel and projective, along with their recursion-theoretic counterparts: arithmetical and analytical. These hierarchies classify the relations over both finite (integers) and infinite (reals, or ω^ω) objects. Although the classical hierarchies are relevant to both finite and infinite computations, it is not in the same way.

Classical complexity theory borrows its basic concepts from recursion theory (reduction, completeness), and applies them by analogy, but the scopes of the two theories are, strictly speaking, different. Indeed, complexity theory studies only a fragment of computable sets and functions, while recursion theory goes far beyond computable world. Finite-state recognizability (regularity) forms the very basic level in complexity hierarchies (although it is of some interest for circuit complexity).

In contrast, finite state automata running over infinite words or trees exhibit remarkable expressive power in terms of the classical hierarchies. Not surprisingly, such automata can recognize uncomputable sets if *computable* means *finite time*. Actually, the word automata reach the second level of the Borel hierarchy, while the tree automata can recognize Borel sets on any finite level, and also — as we have already remarked — some non-Borel sets. So, in spite of a strong restriction to finite memory, automata can reach the very level of complexity studied by the classical definability theory. Putting it the other way around, the classical hierarchies reveal their finite state hardcore.

In this paper we overview the interplay between automata on infinite trees and the classical definability hierarchies, along with a subtle refinement of the Borel hierarchy, known as the hierarchy of Wadge. The emerging picture is not always as expected. Although, in general, topological complexity underlines the automata-theoretic one, the yardsticks are not always compatible, and at one level automata actually refine the Wadge hierarchy. A remarkable application exploits the properties of complete metric spaces: in the proof of the hierarchy theorem for alternating automata, the diagonal argument follows directly from the Banach fixed-point theorem.

2 Climbing up the hierarchies

It is sufficiently representative to consider binary trees. A full binary tree over a finite alphabet Σ is a mapping $t : \{1,2\}^* \to \Sigma$. As a motivating example consider two properties of trees over $\{a,b\}$.

- L is the set of trees such that, on each path, there are infinitely many b's (in symbols: $(\forall \pi \in \{1,2\}^\omega)(\forall m)(\exists n \geq m)\ t(\pi \restriction n) = b$).
- M is the set of trees such that, on each path, there are only finitely many a's (in symbols: $(\forall \pi \in \{1,2\}^\omega)(\exists m)(\forall n \geq m)\ t(\pi \restriction n) = b$).

(In the above, $\pi \restriction n$ denotes the prefix of π of length n.) At first sight the two properties look similar, although the quantifier alternations are slightly different. The analysis below will exhibit a huge difference in complexity: one of the sets is definable by a Π^0_2 formula of arithmetics, while the other is not arithmetical, and even not Borel.

We have just mentioned two views of classical mathematics, where the complexity of sets of trees can be expressed: topology and arithmetics. For the former, the set T_Σ of trees over Σ is equipped with a metric

$$\mathrm{d}(t_1,t_2) = \begin{cases} 0 & \text{if } t_1 = t_2 \\ 2^{-n} \text{ with } n = \min\{|w| : t_1(w) \neq t_2(w)\} & \text{otherwise} \end{cases}$$

For the latter, trees can be encoded as functions over natural numbers ω. The two approaches are reconciliated by viewing trees as elements of the Cantor discontinuum $\{0,1\}^\omega$. Indeed, by fixing a bijection $\iota : \omega \to \{1,2\}^*$, and an injection $\rho : \Sigma \to \{0,1\}^\ell$ (for sufficiently large ℓ), we continuously embed

$$t \mapsto \rho \circ t \circ \iota$$

T_Σ into $(\{0,1\}^\omega)^\ell$, which in turn is homeomorphic to $\{0,1\}^\omega$. It is easy to see that we have a homeomorphism $T_\Sigma \approx \{0,1\}^\omega$, whenever $2 \leq |\Sigma|$.

On the other hand, as far as computability is concerned, the functions in ω^ω can be encoded as elements of $\{0,1\}^\omega$. Assuming that ι above is computable, we can apply the recursion-theoretic classification to trees.

We now recall classical definitions. Following [Hin78], we present topological hierarchies as the relativized versions of recursion-theoretic ones. Thus we somehow inverse the historical order, as the projective hierarchy (over reals) was the first one studied by Borel, Lusin, Kuratowski, Tarski, and others (see [Add04]). However, from computer science perspective, it is natural to start with Turing machine. Let $k, \ell, m, n, \ldots$ range over natural numbers, and $\alpha, \beta, \gamma, \ldots$ over infinite words in $\{0,1\}^\omega$; boldface versions stand for vectors thereof. We consider relations of the form $R \subseteq \omega^k \times (\{0,1\}^\omega)^\ell$, where (k,ℓ) is the *type* of R. The concept of *(partially) recursive relation* directly generalizes the familiar one (see, e.g., [Hin78, Rog67]). In terms of Turing machines, a tuple $\langle \mathbf{m}, \boldsymbol{\alpha} \rangle$ forms an entry for a machine, with $\boldsymbol{\alpha}$ spread over infinite tapes. Note that if a Turing machine gives an answer in finite time, the assertion $R(\mathbf{m}, \boldsymbol{\alpha})$ depends only on a finite fragment of $\boldsymbol{\alpha}$. Consequently the complement $\overline{R}$ of a recursive relation R is also recursive.

The first-order projection of an arbitrary relation R of type $(k+1,\ell)$ is given by

$$\exists^0 R \quad = \quad \{\langle \mathbf{m}, \boldsymbol{\alpha} \rangle : (\exists n)\, R(\mathbf{m}, n, \boldsymbol{\alpha})\}$$

and the second-order projection of a relation R of type $(k, \ell+1)$ is given by

$$\exists^1 R \quad = \quad \{\langle \mathbf{m}, \boldsymbol{\alpha} \rangle : (\exists \beta)\, R(\mathbf{m}, \boldsymbol{\alpha}, \beta)\}$$

The *arithmetical hierarchy* can be presented by

$$\begin{aligned} \Sigma^0_0 &= \text{the class of recursive relations} \\ \Pi^0_n &= \{\overline{R} : R \in \Sigma^0_n\} \\ \Sigma^0_{n+1} &= \{\exists^0 R : R \in \Pi^0_n\} \\ \Delta^0_n &= \Sigma^0_n \cap \Pi^0_n \end{aligned}$$

The relations in the class $\bigcup_{n<\omega} \Sigma^0_n = \bigcup_{n<\omega} \Pi^0_n$ are called *arithmetical*. Note that $\overline{R}$ is arithmetical if so is R.

The *analytical hierarchy* can be presented by

$$\begin{aligned}\Sigma^1_0 &= \text{the class of arithmetical relations}\\ \Pi^1_n &= \{\overline{R} : R \in \Sigma^1_n\}\\ \Sigma^1_{n+1} &= \{\exists^1 R : R \in \Pi^1_n\}\\ \Delta^1_n &= \Sigma^1_n \cap \Pi^1_n.\end{aligned}$$

The two hierarchies have their relativized counterparts usually distinguished by the boldface notation. For a relation R of type $(k, \ell+1)$ and $\beta \in \{0,1\}^\omega$, let

$$R[\beta] \quad = \quad \{\langle \mathbf{m}, \alpha\rangle : R(\mathbf{m}, \alpha, \beta)\}$$

Then, for $i = 0, 1$, we define

$$\begin{aligned}\boldsymbol{\Sigma}^i_n &= \{R[\beta] : R \in \Sigma^i_n,\ \beta \in \{0,1\}^\omega\}\\ \boldsymbol{\Pi}^i_n &= \{R[\beta] : R \in \Pi^i_n,\ \beta \in \{0,1\}^\omega\}\\ \boldsymbol{\Delta}^i_n &= \boldsymbol{\Sigma}^0_n \cap \boldsymbol{\Pi}^i_n\end{aligned}$$

The crucial observation is that the $\boldsymbol{\Sigma}^0_1$ relations (of type $(0, \ell)$) coincide with *open* relations on $\{0,1\}^\omega$ with the Cantor topology. To see this, note that an open set in $\{0,1\}^\omega$ can be presented by $\bigcup_{v \in B} v\{0,1\}^\omega$, for some $B \subseteq \{0,1\}^*$, and hence we can present it by $(\exists n)\, R(n, \alpha, \beta)$, where the parameter β lists the elements of B, and the recursive relation verifies, given $n = \langle k, m\rangle$ that the k^{th} prefix of α coincides with the m^{th} element of B. (The other direction is straightforward.) Next it is easy to see that relations in $\boldsymbol{\Sigma}^0_{n+1}$ coincide with the countable unions of relations in $\boldsymbol{\Pi}^0_n$ (of suitable type). Therefore the classes $\boldsymbol{\Sigma}^0_n$, $\boldsymbol{\Pi}^0_n$ form the initial segment of the *Borel hierarchy* over $\{0,1\}^\omega$.

Similarly, the classes $\boldsymbol{\Sigma}^1_n$, $\boldsymbol{\Pi}^1_n$, form the so-called *projective* hierarchy over $\{0,1\}^\omega$.

Like in computation/complexity theory, the problems can be compared *via* reductions. We say that a continuous mapping of topological spaces, $\varphi : \mathcal{T}_1 \to \mathcal{T}_2$, *reduces* a set $A \subseteq \mathcal{T}_1$ to a set $B \subseteq \mathcal{T}_2$, if $A = \varphi^{-1}(B)$; in this case we say that A is *Wadge reducible* to B, in symbols $A \leq_{\mathrm{W}} B$. A set B is *complete* in a class $\mathcal{C} \subseteq \wp(\mathcal{T})$ if $B \in \mathcal{C}$ and $(\forall A \in \mathcal{C})\ A \leq_{\mathrm{W}} B$.

A remarkable point is that complete sets may have very simple structure.

Example 2.1. The singleton $\{0^\omega\}$ is in Π^0_1, and it is complete for $\boldsymbol{\Pi}^0_1$. The membership in Π^0_1 is seen by presentation of the complement by $(\exists n)\, \alpha(n) \neq 0$. Now let L be any closed subset of ω^ω. Define $\hat{f} : \omega^* \to \omega^*$ by

$$\hat{f}(xy) = 0^{|x|} y$$

where x is the longest prefix of xy being also a prefix of u, for some $u \in L$. Then it is easy to see that the mapping $f : \omega^\omega \to \omega^\omega$ given by

$$f(u)(n) = \hat{f}(u \restriction n+1)(n)$$

is a desired reduction (where $u = u_0 u_1 \dots$ and $u \restriction n+1 = u_0 u_1 \dots u_n$).

It can be seen that, in fact, any singleton $\{\alpha\}$ is complete in $\mathbf{\Pi}^0_1$, although in general it need not be in Π^0_1.

The reader may be puzzled by triviality of this example compared to the construction of complete sets of natural numbers in Π^0_1 or in Σ^0_1. Intuitively, the second-order objects (trees or words) are "less sensitive" to first-order quantification.
In a similar vein, one can show

Example 2.2. The set $\{0,1\}^* 0^\omega$ is in Σ^0_2, and it is complete in $\mathbf{\Sigma}^0_2$.

We now revisit our motivating example from beginning of this section.

Example 2.3. It is not hard to see that the set L is in class Π^0_2. Although the original definition has used a second-order quantifier (for all paths), a simpler definition can be given by exploiting arithmetic (like encoding finite sets of nodes by single numbers):

> $t \in L \iff$ for all $v \in \{1,2\}^*$, there is a finite maximal antichain B below v with $(\forall w \in B)\, t(w) = b$.

On the other hand, the set M, which is by definition in Π^1_1, is also complete in $\mathbf{\Pi}^1_1$ w.r.t. continuous reductions, hence not Borel. The completeness can be seen by reduction of the set W of the suitably encoded wellfounded (non-labeled) trees $T \subseteq \omega^*$ (see, e.g., [NW03]), which is well-known to be $\mathbf{\Pi}^1_1$-complete [Kec95].

3 The power of game languages

The properties of Example 2.3 have a powerful generalization, which is best understood by viewing sequences in $\{a,b\}^\omega$ as outcomes of some infinite two-player game, where one of the players wants to see b infinitely often, while the other does not. To make this game more general/symmetric, we assume that each player has her or his favorite set of letters, and to make the result definite, we assume a priority order on letters. This gives rise to parity games (introduced by Emerson and Jutla [EJ91], and independently by A.W. Mostowski [Mos91a]), the concept highly relevant to the μ-calculus-based model checking and to automata theory (see [Tho97]). We briefly recall it now.

A *parity game* is a perfect information game of possibly infinite duration played by two players, say Eve and Adam. We present it as a tuple

$\langle V_\exists, V_\forall, \mathrm{Move}, p_0, \mathrm{rank}\rangle$, where $V_\exists$ and $V_\forall$ are (disjoint) sets of positions of Eve and Adam, respectively, $\mathrm{Move} \subseteq V \times V$ is the relation of possible moves, with $V = V_\exists \cup V_\forall$, $p_0 \in V$ is a designated initial position, and $\mathrm{rank} : V \to \omega$ is the ranking function.

The players start a play in the position p_0 and then move the token according to relation Move (always to a successor of the current position), thus forming a path in the graph (V, Move). The move is selected by Eve or Adam, depending on who is the owner of the current position. If a player cannot move, she/he looses. Otherwise, the result of the play is an infinite path in the graph, $v_0, v_1, v_2, \ldots$ Eve wins the play if $\limsup_{n\to\infty} \mathrm{rank}(v_n)$, is even, otherwise Adam wins. A crucial property of parity games is the *positional determinacy*: any position is winning for one of the players, and moreover a winning strategy of player θ can be chosen *positional*, i.e., represented by a (partial) function $\sigma : V_\theta \to V$. We simply say that Eve *wins* the game if she has a winning strategy, the similar for Adam. (See [GTW02] for more detailed introduction to parity games.)

Here we are interested in several groups of tree languages related to the parity games.

For $\iota \in \{0, 1\}$ and $\iota \le \kappa < \omega$, let

$$\begin{aligned} \Sigma_{(\iota,\kappa)} &= \{\iota, \iota+1, \ldots, \kappa\} \\ M_{(\iota,\kappa)} &= \{u \in \Sigma^\omega_{(\iota,\kappa)} : \limsup_{n\to\infty} u_n \text{ is even }\} \\ T_{(\iota,\kappa)} &= \{t \in T_{\Sigma_{(\iota,\kappa)}} : (\forall \pi \in \{1,2\}^\omega) t \restriction \pi \in M_{(\iota,\kappa)}\}, \end{aligned}$$

where $t \restriction \pi$ stands for the restriction of t to the path π. That is, $T_{(\iota,\kappa)}$ is the set of trees over $\Sigma_{(\iota,\kappa)}$ such that, on each path, the highest label occurring infinitely often is even. The sets L and M of Example 2.3 can be readily identified with $T_{(1,2)}$ and $T_{(0,1)}$, respectively.

We now present an important game variation of sets $T_{(\iota,\kappa)}$; these will be tree languages over alphabet $\{\exists, \forall\} \times \Sigma_{(\iota,\kappa)}$.

With each tree t in $T_{\{\exists,\forall\}\times\Sigma_{(\iota,\kappa)}}$, we associate a parity game $G(t)$, as described in the previous section, with

- $V_\exists = \{v \in \{1,2\}^* : t(v)\downarrow_1 = \exists\}$,
- $V_\forall = \{v \in \{1,2\}^* : t(v)\downarrow_1 = \forall\}$,
- $\mathrm{Move} = \{(w, wi) : w \in \{1,2\}^*,\ i \in \{1,2\}\}$,
- $p_0 = \varepsilon$ (the root of the tree),
- $\mathrm{rank}(v) = t(v)\downarrow_2$, for $v \in \{1,2\}^*$.

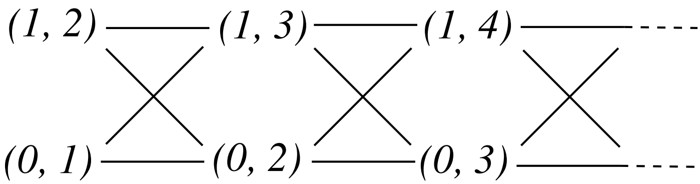

FIGURE 1. The Mostowski–Rabin index hierarchy.

The set $W_{(\iota,\kappa)}$ consists of those trees for which Eve wins the game $G(t)$. Note that this means that Eve can force the resulting path π to satisfy $(t \restriction \pi)\downarrow_2 \in M_{(\iota,\kappa)}$.

Finally, we introduce the *weak version* of all the concepts above, which is obtained by replacing everywhere lim sup by sup. We denote by $L^\flat$ the weak version of L. So, in particular $M^\flat_{(\iota,\kappa)} = \{u \in \Sigma^\omega_{(\iota,\kappa)} : \sup_{n\to\infty} u_n \text{ is even }\}$. Similarly, the *weak parity games* differ from the games defined above in that Eve wins a play if the *highest* rank occurring in the play is even.

It is useful to have a partial ordering on pairs (ι,κ), with $\iota \in \{0,1\}$, which we call *Mostowski-Rabin indices*. We let $(\iota,\kappa) \sqsubseteq (\iota',\kappa')$ if either $\iota' \le \iota$ and $\kappa \le \kappa'$ (i.e., $\{\iota,\ldots,\kappa\} \subseteq \{\iota',\ldots,\kappa'\}$) or $\iota = 0$, $\iota' = 1$, and $\kappa + 2 \le \kappa'$ (i.e., $\{\iota+2,\ldots,\kappa+2\} \subseteq \{\iota',\ldots,\kappa'\}$). We consider the indices $(1,\kappa)$ and $(0,\kappa-1)$ as *dual*, and let $\overline{(\iota,\kappa)}$ denote the index dual to (ι,κ). Note that $\overline{\overline{(\iota,\kappa)}} = (\iota,\kappa)$. The ordering is represented on Figure 1.

Clearly, in each of the above-defined families, the ordering on Mostowski-Rabin indices induces inclusion of corresponding sets.

Now the crucial observation is the following. If $\mathcal{T}$ is a complete metric space then no *contracting* reduction can reduce a set $A \subseteq \mathcal{T}$ to its complement $\overline{A}$. Indeed, otherwise, by the Banach Fixed-Point Theorem, we would have

$$a \in A \iff f(a) \in \overline{A} \iff a \in \overline{A} \quad \text{(contradiction)},$$

for the fixed point $a = f(a)$.
It immediately implies the following.

Lemma 3.1. No contracting mapping reduces $W_{\overline{(\iota,\kappa)}}$ to $W_{(\iota,\kappa)}$, or $W^\flat_{\overline{(\iota,\kappa)}}$ to $W^\flat_{(\iota,\kappa)}$.

Proof. Although $W_{\overline{(\iota,\kappa)}}$ and $W_{(\iota,\kappa)}$ are over different alphabets, we have an isometry of $T_{\Sigma_{(\iota,\kappa)}}$ and $T_{\Sigma_{\overline{(\iota,\kappa)}}}$, induced by the re-labeling of symbols which exchanges quantifiers and alters the ranks by ± 1. This isometry reduces $\overline{W_{(\iota,\kappa)}}$ to $W_{\overline{(\iota,\kappa)}}$, so the claim follows from the observation above. The argument for weak version is similar. Q.E.D.

It turns out that we can strengthen the above lemma by removing the hypothesis of contractivity. This is because, in general, any continuous reduction of $W_{(\iota,\kappa)}$ to some L can be improved to a contracting one, by composing it with a "stretching" reduction of $W_{(\iota,\kappa)}$ to itself. The details can be found in [AN]. Thus we obtain the following.

Theorem 3.2. The game languages form a hierarchy w.r.t. the Wadge reducibility, i.e.,

$$\begin{aligned}(\iota,\kappa) \sqsubseteq (\iota',\kappa') \quad &\text{iff} \quad W_{(\iota,\kappa)} \leq_{\mathrm{W}} W_{(\iota',\kappa')} \\ &\text{iff} \quad W^{\flat}_{(\iota,\kappa)} \leq_{\mathrm{W}} W^{\flat}_{(\iota',\kappa')}\end{aligned}$$

This result has several applications involving automata. Let us first recall definition of an alternating parity automaton.

An *alternating parity tree automaton* can be presented as a tuple $\mathcal{A} = \langle \Sigma, Q_{\exists}, Q_{\forall}, q_0, \delta, \mathrm{rank}\rangle$, where the set of states Q is partitioned into existential states $Q_{\exists}$ and universal states $Q_{\forall}$, $\delta \subseteq Q \times \Sigma \times \{1,2,\varepsilon\} \times Q$ is a transition relation, and rank : $Q \to \omega$ a *rank* function. An input tree t is accepted by $\mathcal{A}$ iff Eve has a winning strategy in the parity game $\langle Q_{\exists} \times \{1,2\}^*, Q_{\forall} \times \{1,2\}^*, (q_0,\varepsilon), \mathrm{Move}, \mathrm{rank}\rangle$, where $\mathrm{Move} = \{((p,v),(q,vd))\colon v \in \mathrm{dom}(t), (p,t(v),d,q) \in \delta\}$ and $\mathrm{rank}(q,v) = \mathrm{rank}(q)$.

We can assume without loss of generality that $\min \mathrm{rank}(Q)$ is 0 or 1. The pair $(\min \mathrm{rank}(Q), \max \mathrm{rank}(Q))$ is the *Mostowski-Rabin index* of the automaton.

A *weak alternating parity tree automaton* is defined similarly, by restriction to weak parity games. Strictly speaking, a weak automaton is not a parity automaton, but it can be easily turned into one. It is enough to multiply the set of states by $\mathrm{rank}(Q)$ so that the second component keeps record of the highest rank seen so far (it can only increase). It is well known that the languages recognized by weak alternating automata are exactly those recognizable by both $(0,1)$ and $(1,2)$ automata (it follows essentially from [Rab70]).

It is straightforward to see that each $W_{(\iota,\kappa)}$ is recognized by a parity automaton of index (ι,κ), and each $W^{\flat}_{(\iota,\kappa)}$ is recognized by a weak parity automaton of index (ι,κ).

The next important observation is the following lemma:

Lemma 3.3. If a set of trees T is recognized by a (weak) alternating automaton of index (ι,κ) then $T \leq_{\mathrm{W}} W_{(\iota,\kappa)}$ (resp. $T \leq_{\mathrm{W}} W^{\flat}_{(\iota,\kappa)}$).

The exact construction is somewhat tedious, but the idea of the reduction is simple: for a tree t, we construct a full game tree and then forget anythings but ranks. The details are presented in [Arn99, AN01], where the reduction is even made contracting, but in view of Theorem 3.2, it is not necessary.

Combining Theorem 3.2 with Lemma 3.3, we obtain

Theorem 3.4. The tree languages $W_{(\iota,\kappa)}$ form a strict hierarchy for the Mostowski-Rabin indices of alternating parity automata.

The tree languages $W^{\flat}_{(\iota,\kappa)}$ form a strict hierarchy for Mostowski-Rabin indices of weak alternating parity automata.

The first claim was established by Bradfield [Bra98]; the proof *via* the Banach Theorem was given later by Arnold [Arn99] (see also [AN01]).

The strictness of the hierarchy of weak automata was first established by Mostowski [Mos91b], who shown that it is equivalent to a hierarchy based on weak monadic formulas, and then used the strictness of the latter hierarchy, previously proved by W. Thomas [Tho83].

As Skurczyński showed [Sku93] (by other examples) that there are $\mathbf{\Pi}^0_n$ and $\mathbf{\Sigma}^0_n$-complete tree languages recognized by weak alternating automata of index $(0,n)$ and $(1,n+1)$ accordingly, Lemma 3.3 also implies that the sets $W^{\flat}_{(\iota,\kappa)}$ are hard on the corresponding finite levels of the Borel hierarchy. Recently, Duparc and Murlak [DM07] showed that these sets are actually complete in these classes.

Theorem 3.5 (Duparc-Murlak, [DM07])**.** If a tree language T is recognized by a weak alternating automaton of index $(0,n)$ (resp. $(1,n+1)$) it holds that $T \in \mathbf{\Pi}^0_n$ (resp. $T \in \mathbf{\Sigma}^0_n$).

Let us complete this recent theorem by what we have known since long time about strong alternating automata.

Theorem 3.6. If a tree language T is recognized by an alternating automaton of index $(0,1)$ (resp. $(1,2)$) it holds that $T \in \mathbf{\Pi}^1_1$ (resp. $T \in \mathbf{\Sigma}^1_1$).

For any recognizable tree language T, $T \in \mathbf{\Delta}^1_2$.

The first claim was (essentially) established by Rabin [Rab70] in terms of the formulas of S2S and for *nondeterministic* automata of index $(1,2)$, now called *Büchi automata*. It was later shown [AN90] that for Büchi automata alternation does not matter. Note that this implies in particular that the set M of Example 2.3 cannot be recognized by a Büchi automaton [Rab70]. The second claim follows from definition and Rabin's Complementation Lemma [Rab69].

4 How fine is the Wadge hierarchy?

In the previous section we saw that with regular tree languages one can go much higher in the Borel hierarchy than with regular ω-languages. Now we should like to concentrate on the *fineness* of the hierarchy. Let us start with a simple example.

For $n \in \omega$, let L_n denote the set of trees over the alphabet $[0,n] = \{0,1,\ldots,n\}$, whose leftmost path satisfies the weak parity condition, i. e., the highest label on this path is even. For example: $L_0 = T_{[0,0]}$ consists of the only tree over the alphabet $\{0\}$, and L_1, a closed subset of $T_{[0,1]}$, consists of trees with 0's on the leftmost path and 0's or 1's elsewhere. It is an easy exercise to show that L_n are regular.

Even everyday intuition of complexity tells us that L_{k+1} is more complex then L_k. This can be formalized by means of continuous reductions introduced in the previous section. Consider an identity function $\mathrm{id} : T_{[0,\ell]} \to T_{[0,k]}$, with $\ell < k$. Clearly, this function reduces L_ℓ to L_k: $t \in L_\ell$ iff $\mathrm{id}(t) \in L_k$. Hence the languages L_n form a hierarchy: $L_0 \leq_{\mathrm{W}} L_1 \leq_{\mathrm{W}} L_2 \leq_{\mathrm{W}} \ldots$.

OK, but this already happened with the weak game languages from the previous section, so what is the difference? Well, observe that all these languages can be presented as a finite Boolean combination of closed sets, e. g.

$$L_3 = \{t\colon \forall_i \; t(0^i) \in [0,2]\} \setminus \{t\colon \forall_i \; t(0^i) \in [0,1]\} \cup \{t\colon \forall_i \; t(0^i) \in [0,0]\}\,.$$

Consequently, our entire hierarchy lies within Δ^0_2!

'All right,' the reader might say, 'but how do I know that, say, L_7 cannot be reduced to L_6? How do I know that this "hierarchy" is *strict*?' It is, but showing that directly would be rather tiresome. Instead, we shall use a handy characterization provided by Wadge games.

Originally, these games were defined for ω-words (see [PP04]). Here, we shall use a tree version. For any pair of tree languages $L \subseteq T_\Sigma, M \subseteq T_\Gamma$ the *Wadge game* $G_{\mathrm{W}}(L,M)$ is played by Spoiler and Duplicator. Each player builds a tree, $t_S \in T_\Sigma$ and $t_D \in T_\Gamma$ respectively. In every round, first Spoiler adds at least one level to t_S and then Duplicator can either add some levels to t_D or skip a round. Duplicator must not skip infinitely long, so that t_D is really an infinite tree. Duplicator wins the game if $t_S \in L \iff t_D \in M$.

Lemma 4.1 (Wadge)**.** Duplicator has a winning strategy in $G_{\mathrm{W}}(L,M)$ if and only if $L \leq_{\mathrm{W}} M$.

Proof. Essentially, a winning strategy for Duplicator can be transformed into a continuous reduction, and vice versa.

Suppose Duplicator has a winning strategy ρ. For any tree t constructed by Spoiler, there exist a unique tree t_ρ which will be constructed by Duplicator if he is using the strategy ρ. The map $t \mapsto t_\rho$ is continuous by the rules of the Wadge game, and $t \in L \iff t_\rho \in M$ since ρ is winning.

Conversely, suppose there exist a reduction $t \mapsto \varphi(t)$. It follows that there exist a sequence n_k (without loss of generality, increasing) such that the level k of $\varphi(t)$ depends only on the levels $1, \ldots, n_k$ of t. Then the strategy for Duplicator is the following: if the number of the round is n_k, fill in the k-th level of t_D according to $\varphi(t_S)$; otherwise skip. Q.E.D.

Let us now see that the languages $L_1, L_2, \ldots$ form a strict hierarchy, i.e., $L_\ell \not\leq_{\mathrm{W}} L_k$ for $\ell > k$. Consider the following strategy for Spoiler in $G_{\mathrm{W}}(L_\ell, L_k)$. Outside of the leftmost path play 1 all the time - it does not matter anyhow. On the leftmost path always play $m+1$, where m is the last number played by Duplicator on the leftmost path of his tree (or 0 if he has kept skipping so far). This strategy only uses numbers $[1, k+1] \subseteq [1, \ell]$, so it is legal. Obviously, the highest number we use on the leftmost path is of different parity then the highest number used by Duplicator, so $t_S \in L_\ell \iff t_D \notin L_k$. Hence, the strategy is winning for Spoiler, and by the lemma above $L_\ell \not\leq_{\mathrm{W}} L_k$.

Observe that in the above argument we have shown that Duplicator does not have a winning strategy by providing a winning strategy for Spoiler. In general it does not always hold that one of the players must have a winning strategy in $G_{\mathrm{W}}(L, M)$. Luckily, by Martin's famous determinacy theorem, it holds for Borel sets.

Theorem 4.2. If L, M are Borel languages, than one of the players has a winning strategy in $G_{\mathrm{W}}(L, M)$.

In fact the power of Wadge games relies on the above result: it lets us replace a non-existence proof with an existence proof. Without determinacy, Wadge games only give a rather trivial correspondence between reductions and strategies.

The Wadge ordering $\leq_{\mathrm{W}}$ induces a natural equivalence relation, $L \equiv_{\mathrm{W}} M$ iff $L \leq_{\mathrm{W}} M$ and $L \geq_{\mathrm{W}} M$. The order induced on the $\equiv_{\mathrm{W}}$ equivalence classes of Borel languages is called the *Wadge hierarchy*. The determinacy theorem actually gives a very precise information on the shape of the Wadge hierarchy.

Theorem 4.3 (Wadge Lemma)**.** For Borel languages L, M it holds that

$$L \leq_{\mathrm{W}} M \quad \text{or} \quad \overline{L} \geq_{\mathrm{W}} M\,.$$

The proof of this result simply transforms Spoiler's winning strategy in $G_{\mathrm{W}}(L, M)$, which must exist by determinacy, into Duplicator's winning strategy in $G_{\mathrm{W}}(M, \overline{L})$ (see [Kec95] or [PP04]). In other words the theorem says that the width of the Wadge hierarchy is at most two, and if L and M are incomparable, then $L \equiv_{\mathrm{W}} \overline{M}$. It means that the Wadge ordering is almost linear. The second fundamental result states that it is also a well-ordering.

Theorem 4.4 (Wadge-Martin-Monk)**.** The Wadge hierarchy is wellfounded.

Altogether, the position of a language in the Wadge hierarchy is determined, up to complementation, by its height.

If $L \equiv_W \overline{L}$ then L is called *selfdual.* Otherwise L is not comparable with $\overline{L}$ and is called *non-selfdual.* Steel and Van Weesp proved that the selfdual and non-selfdual levels alternate (see [Kec95]). If the alphabet is finite, which is our case, on limit steps we have non-selfduals. Furthermore, the selfduals on successor levels can be obtained as disjoint unions of their predecessors. All this makes it reasonable to ignore selfduals when counting the height. Hence, we choose the following definition of the Wadge degree:

- $\mathbf{d}_W(\varnothing) = \mathbf{d}_W(\overline{\varnothing}) = 1$,
- $\mathbf{d}_W(L) = \sup\{\mathbf{d}_W(M)+1 : M$ is non-selfdual, $M <_W L\}$ for $L >_W \varnothing$.

We have now all the tools necessary to formalize the question asked in the title of the present section. For a family of languages $\mathcal{F}$ define the *height* of the Wadge hierarchy restricted to $\mathcal{F}$ as the order type of the set $\{\mathbf{d}_W(L) \colon L \in \mathcal{F}\}$ with respect to the usual order on ordinals. What we are interested in is the height of the hierarchy of regular languages.

We have shown already that the height of the hierarchy of $\{L_0, L_1, \ldots\}$ is ω. This of course gives a lower bound for the height of the hierarchy of all regular languages. We shall now see how this result can be improved. We consider a subclass of regular languages, the languages recognized by weak alternating automata. Any lower bound for weak languages will obviously hold for regular languages as well.

It will be convenient to work with languages of binary trees which are not necessarily full, i.e., partial functions from $\{0,1\}^*$ to Σ with prefix closed domain. We call such trees *conciliatory.* Observe that the definition of weak automata works for conciliatory trees as well. We shall write $L_C(\mathcal{A})$ to denote the set of conciliatory trees accepted by $\mathcal{A}$. For conciliatory languages L, M one can define a suitable version of Wadge games $G_C(L, M)$. Since it is not a problem if the players construct a conciliatory tree during the play, they are now both allowed to skip, even infinitely long. Analogously one defines the conciliatory hierarchy induced by the order $\leq_C$, and the conciliatory degree $\mathbf{d}_C$.

The conciliatory hierarchy embeds naturally into the non-selfdual part of the Wadge hierarchy. The embedding is given by the mapping $L \mapsto L_S$, where L is a language of conciliatory trees over Σ, and L_s is a language of full trees over $\Sigma \cup \{s\}$ which belong to L when we ignore the nodes labeled with s (together with the subtrees rooted in their right children) in a top down manner. Proving that $L \leq_C M \iff L_s \leq_W M_s$ for all conciliatory languages L and M only requires translating strategies form one game to

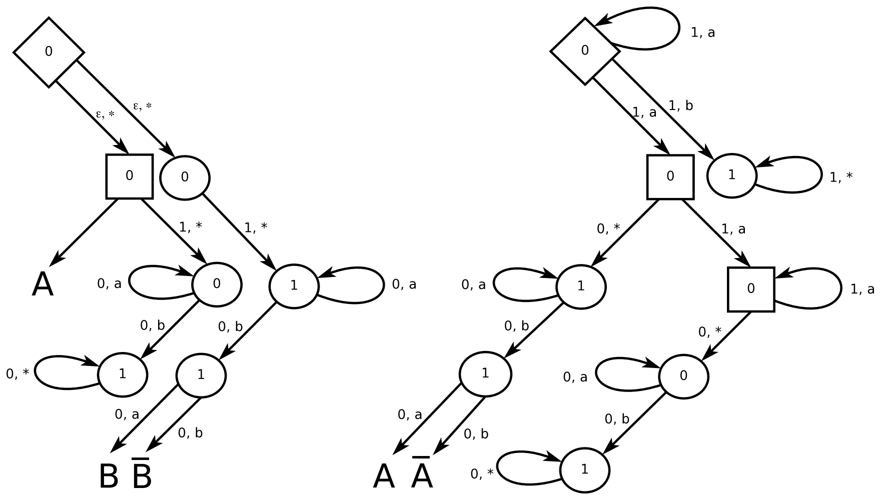

FIGURE 2. The automata $\mathcal{B} + \mathcal{A}$ and $\mathcal{A} \cdot \omega$.

the other. It can be done easily, since arbitrary skipping in $G_{\mathrm{C}}(L, M)$ gives the same power as the s labels in $G_{\mathrm{W}}(L_s, M_s)$. Within the family of languages of finite Borel rank, the embedding is actually an isomorphism, and $\mathbf{d}_{\mathrm{C}}(L) = \mathbf{d}_{\mathrm{W}}(L_s)$ [DM07].

Observe that if L is recognized by a weak alternating automaton, so is L_s. Indeed, by adding to δ a transition $p \xrightarrow{0,s} p$ for each state p one transforms an automaton $\mathcal{A}$ into $\mathcal{A}_s$ such that $L(\mathcal{A}_s) = (L_{\mathrm{C}}(\mathcal{A}))_s$. Hence, the conciliatory subhierarchy of weakly recognizable languages embeds into the Wadge hierarchy of weakly recognizable languages, and it is enough to show a lower bound for conciliatory languages.

So far, when constructing hierarchies, we have been defining the whole family of languages right off. This time we shall use a different method. We shall define operations transforming simple languages into more sophisticated ones. These operations will induce, almost accurately, classical ordinal operations on the degrees of languages: sum, multiplication by ω, and exponentiation with the base ω_1. We shall work with automata on trees over a fixed alphabet $\{a, b\}$.

The sum $\mathcal{B} + \mathcal{A}$ and multiplication $\mathcal{A} \cdot \omega$ are realized by combining automata recognizing simpler languages with a carefully designed gadget. The constructions are shown on Figure 2. 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 $\mathcal{A}$, $\overline{\mathcal{A}}$, $\mathcal{B}$ and $\overline{\mathcal{B}}$ should be understood as transitions to the initial

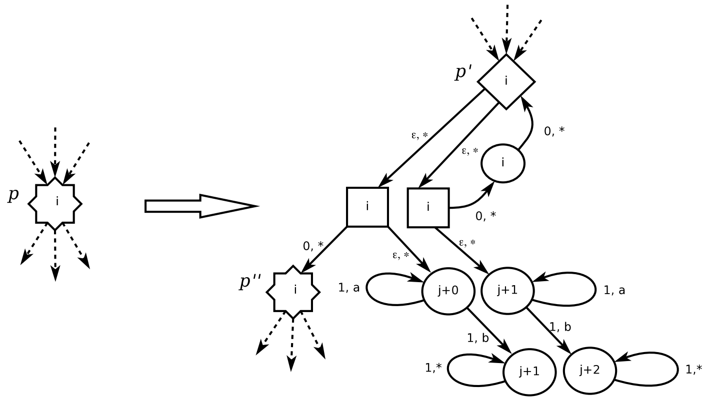

FIGURE 3. The gadget to replace p in the construction of $\exp \mathcal{A}$.

states of the according automata. The priority functions of these automata might need shifting up, so that they were not using the value 0.

The automaton $\exp \mathcal{A}$ 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 Figure 3. 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. The number j is the least even number greater or equal to $i = \operatorname{rank} p$.

Abusing slightly the notation we may formulate the properties of the three constructions as follows.

Theorem 4.5 (Duparc-Murlak, [DM07]). For all weak alternating automata $\mathcal{A}$, $\mathcal{B}$ it holds that $\mathbf{d}_{\mathrm{C}}(\mathcal{B} + \mathcal{A}) = \mathbf{d}_{\mathrm{C}}(\mathcal{B}) + \mathbf{d}_{\mathrm{C}}(\mathcal{A})$, $\mathbf{d}_{\mathrm{C}}(\mathcal{A} \cdot \omega) = \mathbf{d}_{\mathrm{C}}(\mathcal{A}) \cdot \omega$, and $\mathbf{d}_{\mathrm{C}}(\exp \mathcal{A}) = \omega_1^{\mathbf{d}_{\mathrm{C}}(\mathcal{A})+\varepsilon}$, where

$$\varepsilon = \begin{cases} -1 & \text{if } \mathbf{d}_{\mathrm{C}}(\mathcal{A}) < \omega \\ \ \ 0 & \text{if } \mathbf{d}_{\mathrm{C}}(\mathcal{A}) = \beta + n \text{ and } \operatorname{cof}\beta = \omega_1 \\ +1 & \text{if } \mathbf{d}_{\mathrm{C}}(\mathcal{A}) = \beta + n \text{ and } \operatorname{cof}\beta = \omega. \end{cases}$$

As a corollary we obtain the promised bound.

Theorem 4.6 (Duparc-Murlak, [DM07]). The Wadge hierarchy of weakly recognizable tree languages has the height of at least ε_0, the least fixed point of the exponentiation with the base ω.

Proof. It is enough to show the bound for conciliatory languages. By iterating finitely many times sum and multiplication by ω we obtain multiplication by ordinals of the form $\omega^n k_n + \ldots + \omega k_1 + k_0$, i.e., all ordinals less then 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. Q.E.D.

Recently, the second author of this survey has found a modification of the pseudo-exponentiation construction which results in ordinary exponentiation $\alpha \mapsto \omega_1^\alpha$. This result makes it very tempting to conjecture that these are in fact all Wadge degrees realised by weak automata, and if one replaces ω_1 by ω^ω, one gets the degree of the language in the Wadge hierarchy restricted to weakly recognizable languages.

Supposing that the conjecture is true, the next step is an effective description of each degree. Or, in other words, an algorithm to calculate the position of a given language in the hierarchy. Obtaining such a description for all regular languages is the ultimate goal of the field we are surveying. So far this goal is seems far away. The solution might actually rely on analytical determinacy. On the other hand, it may also be the case that determinacy for regular languages is implied by ZFC. The knowledge in this subject is scarce.

To end up with some good news, the problem has been solved for an important and natural subclass of regular languages, the languages recognized by deterministic automata (see below for definition).

Theorem 4.7 (Murlak, [Mur06]). The hierarchy of deterministically recognizable languages has the height of $\omega^{\omega \cdot 3} + 3$. Furthermore, there exist an algorithm calculating the exact position of a given language in this hierarchy.

5 Topology versus computation

In this concluding section we should like to confront the classical definability hierarchies with the automata-theoretic hierarchies based on the Mostowski–Rabin index. To this end, let us first recall the concepts of non-deterministic and deterministic tree automata. They are special cases of alternating automata, but it is convenient to use traditional definitions. A *non-deterministic* parity tree automaton over trees in T_Σ can be presented as $A = \langle \Sigma, Q, q_0, \delta, \mathrm{rank} \rangle$, where $\delta \subseteq Q \times \Sigma \times Q \times Q$. A transition $(q, \sigma, p_1, p_2) \in \delta$ is usually written $q \xrightarrow{\sigma} p_1, p_2$.

A *run* of A on a tree $t \in T_\Sigma$ is itself a tree in T_Q such that $\rho(\varepsilon) = q_0$, and, for each $w \in dom\,(\rho)$, $\rho(w) \overset{t(w)}{\to} \rho(w1), \rho(w2)$ is a transition in δ. A *path* in ρ is *accepting* if the highest rank occurring infinitely often along it is even. A *run is accepting* if so are all its paths. Again, the Mostowski-Rabin index of an automaton is the pair $(\min \mathrm{rank}(Q), \max \mathrm{rank}(Q))$, where we assume that the first component is 0 or 1.

An automaton is *deterministic* if δ is a partial function from $Q \times \Sigma$ to $Q \times Q$. It can be observed that languages $W_{(\iota,\kappa)}$ defined in Section 3 can be recognized by non-deterministic automata of index (ι, κ), respectively, and that languages $T_{(\iota,\kappa)}$ defined there can be recognized by deterministic automata of corresponding indices.

In general, the index may decrease if we replace an automaton by an equivalent one of higher type. For example, it is not hard to see that the complements of languages $T_{(\iota,\kappa)}$ can all be recognized by non-deterministic automata of index $(1, 2)$ (Büchi automata), hence these languages themselves are of alternating index $(0, 1)$. But it was showed in [Niw86] that these languages form a hierarchy for the Mostowski-Rabin index of non-deterministic automata. It can be further observed that all $T_{(\iota,\kappa)}$ with $(0, 1) \sqsubseteq (\iota, \kappa)$ are $\mathbf{\Pi}^1_1$-complete, hence by the general theory [Kec95], they are all equivalent w.r.t. the Wadge reducibility. (In fact, it is not difficult to find the reductions to $T_{(0,1)}$ directly.) So in this case the automata-theoretic hierarchy is more fine than the Wadge hierarchy, which is a bit surprising in view of the fineness of the latter hierarchy, as seen in the previous section.

Let us now compare the index hierarchy and the Wadge hierarchy. For infinite words, this comparison reveals a beautiful correspondence, discovered by Klaus Wagner.

Theorem 5.1 (Wagner, [Wag79])**.**

1. Regular ω-languages have exactly the Wadge degrees of the form $\omega_1^k n_k + \ldots + \omega_1^1 n_1 + n_0$ for $k < \omega$ and $n_0, \ldots, n_k < \omega$.

2. The languages recognized by deterministic automata using $k+1$ ranks (index $[0, k]$ or $[1, k+1]$) correspond to degrees $\leq \omega_1^k$.

Hence, for regular ω-languages, the Wadge hierarchy is a refinement of the index hierarchy. For trees the situation is more complex because we have four nontrivial hierarchies (alternating, weak-alternating, nondeterministic, and deterministic).

The correspondence for weak alternating automata is not yet fully understood. By Theorem 3.5, the raise of topological complexity (in terms of Borel hierarchy) forces the raise of the index complexity. However, the converse is an open problem. *A priori* it is possible that an infinite sequence of tree languages witnessing the weak index hierarchy can be found inside a single Borel class, although it would be rather surprising.

What we do know is that a similar pathology cannot happen for deterministically recognizable tree languages. Indeed, for this class the two hierarchies are largely compatible, however their scope is not large: a deterministic language can either be recognized by a weak automaton of index (at most) $(0,3)$, and hence, by Theorem 3.5 is in the Borel class $\mathbf{\Pi}^0_3$, or it is $\mathbf{\Pi}^1_1$-complete [NW03]. Moreover, the membership in Borel and in weak-index classes is decidable for deterministic languages [NW03, Mur05].

On the other hand, the kind of pathology described above actually does happen if we regard the *deterministic* index hierarchy, i.e., for a deterministically recognizable language we look for the lowest index of a deterministic automaton recognizing it (the case rarely considered in literature). Observe that the hierarchy of regular ω-languages embeds into the hierarchy of deterministic tree languages by a mapping $L \mapsto \{t\colon$ the leftmost branch of t is in $L\}$. Recall that all the regular ω-languages are Boolean combinations of Σ^0_2 languages, denoted Boole(Σ^0_2). It follows that there are deterministic tree languages from each level of the deterministic index hierarchy which are inside Boole(Σ^0_2). At the same time one only needs index $(0,1)$ to get a $\mathbf{\Pi}^1_1$-complete set. In other words, for some $\mathbf{\Pi}^1_1$-complete languages $(0,1)$ is enough, but there are Σ^0_2 languages which need an arbitrarily high index! This means that the deterministic index hierarchy does not embed into the Wadge hierarchy. Apparently, it measures an entirely different kind of complexity.

One might suspect that alternating index would be a more suitable measure in this context. Alternation saves us from increasing the index with complementation. Indeed, the complementation of an alternating automaton is done simply by swapping $Q_\exists$ and $Q_\forall$, and shifting the ranks by one. (To make complementation easy was an original motivation behind alternating automata [MS87].) If a language has index (ι,κ), its complement will only need $\overline{(\iota,\kappa)}$, and vice versa. As it was stated in Section 3, the strong game languages showing the strictness of the alternating hierarchy form also a strict hierarchy within the Wadge hierarchy. In fact, since each recognizable tree language can be continuously reduced to one of them, they give a scaffold for further investigation of the hierarchy. Such a scaffold will be much needed since the non-Borel part of the Wadge hierarchy is a much dreaded and rarely visited place.

References

[Add04] John W. Addison. Tarski's theory of definability: common themes in descriptive set theory, recursive function theory, classical pure logic, and finite-universe logic. *Ann. Pure Appl. Logic*, 126(1-3):77–92, 2004.

[AN] A. Arnold and D. Niwiński. Continuous separation of game languages. To appear.

[AN90] André Arnold and Damian Niwinski. Fixed point characterization of büchi automata on infinite trees. *Elektronische Informationsverarbeitung und Kybernetik*, 26(8/9):451–459, 1990.

[AN01] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2001.

[Arn99] André Arnold. p329 the μ-calculus alternation-depth hierarchy is strict on binary trees. *ITA*, 33(4/5):329–340, 1999.

[Bra98] Julian C. Bradfield. Simplifying the modal mu-calculus alternation hierarchy. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *STACS*, volume 1373 of *Lecture Notes in Computer Science*, pages 39–49. Springer, 1998.

[DM07] Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. In Erzsébet Csuhaj-Varjú and Zoltán Ésik, editors, *FCT*, volume 4639 of *Lecture Notes in Computer Science*, pages 261–273. Springer, 2007.

[EJ91] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS*, pages 368–377. IEEE, 1991.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[Hin78] Peter G. Hinman. *Recursion-theoretic hierarchies*. Springer-Verlag, Berlin, 1978. Perspectives in Mathematical Logic.

[Kec95] Alexander S. Kechris. *Classical descriptive set theory*, volume 156 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.

[Mos80] Yiannis N. Moschovakis. *Descriptive set theory*, volume 100 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1980.

[Mos91a] Andrzej W. Mostowski. Games with forbidden positions. Preprint 78, Uniwersytet Gdańsk, Instytyt Matematyki, 1991.

[Mos91b] Andrzej Wlodzimierz Mostowski. Hierarchies of weak automata and weak monadic formulas. *Theor. Comput. Sci.*, 83(2):323–335, 1991.

[MS87] David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theor. Comput. Sci.*, 54:267–276, 1987.

[Mur05] Filip Murlak. On deciding topological classes of deterministic tree languages. In C.-H. Luke Ong, editor, *CSL*, volume 3634 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2005.

[Mur06] Filip Murlak. The wadge hierarchy of deterministic tree languages. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 408–419. Springer, 2006.

[Niw86] Damian Niwiński. On fixed-point clones (extended abstract). In Laurent Kott, editor, *ICALP*, volume 226 of *Lecture Notes in Computer Science*, pages 464–473. Springer, 1986.

[NW03] Damian Niwinski and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

[PP04] Dominique Perrin and Jean-Éric Pin. *Infinite Words. Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics*. Elsevier, Amsterdam, 2004.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[Rab70] Michael O. Rabin. Weakly definable relations and special automata. In *Mathematical Logic and Foundations of Set Theory (Proc. Internat. Colloq., Jerusalem, 1968)*, pages 1–23. North-Holland, Amsterdam, 1970.

[Rog67] Hartley Rogers, Jr. *Theory of recursive functions and effective computability*. McGraw-Hill Book Co., New York, 1967.

[Sku93] Jerzy Skurczynski. The borel hierarchy is infinite in the class of regular sets of trees. *Theor. Comput. Sci.*, 112(2):413–418, 1993.

[Tho83] Wolfgang Thomas. A hierarchy of sets of infinite trees. In Armin B. Cremers and Hans-Peter Kriegel, editors, *Theoretical Computer Science*, volume 145 of *Lecture Notes in Computer Science*, pages 335–342. Springer, 1983.

[Tho97] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*, pages 389–455. Springer, Berlin, 1997.

[Wag79] Klaus Wagner. On omega-regular sets. *Information and Control*, 43(2):123–177, 1979.