On the separation question for tree languages

André Arnold[1], Henryk Michalewski[*2], and Damian Niwiński[†2,3]

1 Talence, `andre.arnold@club-internet.fr`
2 University of Warsaw
Faculty of Mathematics, Informatics, and Mechanics
`{H.Michalewski,D.Niwinski}@mimuw.edu.pl`
3 Institute of Mathematics
Polish Academy of Sciences

Abstract

We show that the separation property fails for the classes Σ_n of the Rabin-Mostowski index hierarchy of alternating automata on infinite trees. This extends our previous result (obtained with Szczepan Hummel) on the failure of the separation property for the class Σ_2 (i.e., for co-Büchi sets). It remains open whether the separation property does hold for the classes Π_n of the index hierarchy. To prove our result, we first consider the Rabin-Mostowski index hierarchy of deterministic automata on infinite words, for which we give a complete answer (generalizing previous results of Selivanov): the separation property holds for Π_n and fails for Σ_n-classes. The construction invented for words turns out to be useful for trees *via* a suitable game.

1998 ACM Subject Classification F.1.1 Models of Computation, F.4.1 Mathematical Logic, F.4.3 Formal Languages

Keywords and phrases Alternating automata on infinite trees, Index hierarchy, Separation property

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.396

1 Introduction

The separation question is whether two disjoint sets A and B can be separated by a set C (i.e., $A \subseteq C$ and $B \subseteq \overline{C}$) which is in some sense simpler. Separation is one of the main issues in descriptive set theory. A fundamental result due to Lusin is that two analytic sets can be always separated by a Borel set, but two co-analytic sets in general cannot. The former implies that if a set is simultaneously analytic and co-analytic then it is necessarily Borel, which is the celebrated Suslin Theorem (see, e.g., [8] or [7]).

A well-known fact in automata theory exhibits a similar pattern: if a set of infinite trees as well as its complement are both recognizable by Büchi automata then they are also recognizable by weak alternating automata (weakly recognizable, for short). This result was first proved by Rabin [10] in terms of monadic second-order logic; the automata-theoretic statement was given by Muller, Saoudi, and Schupp [9]. It is not difficult to adapt Rabin's proof to obtain the separation property: any two disjoint Büchi recognizable sets of trees can be separated by a weakly recognizable set (see, e.g., [5]). Quite analogical to the co-analytic case, the separation property fails in general for the dual class of co-Büchi tree languages

* Supported by the Polish Ministry of Science grant nr N N206 567840.
† Supported by the Polish Ministry of Science grant nr N N206 567840.

29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 396–407
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

(i.e., the complements of Büchi recognizable sets). In [5], a pair of such sets is presented that cannot be separated by any Borel set, hence *a fortiori* by any weakly recognizable set.

A systematic study of the separation property for tree automata has been undertaken by Santocanale and Arnold [11]. They asked if the above-mentioned result of Rabin can be shifted to the higher levels of the index hierarchy of alternating automata with an appropriate generalization of weak recognizability. The question stems naturally from the μ-calculus version of Rabin's result which states that if a tree language is definable both by a Π_2-term (i.e., with a pattern $\nu\mu$) and a Σ_2-term ($\mu\nu$), then it is also definable by an alternation free term, i.e., one in $Comp(\Pi_1 \cup \Sigma_1)$ [2]. Somewhat surprisingly, Santocanale and Arnold [11] discovered that the equation

$$\Pi_n \cap \Sigma_n = Comp(\Pi_{n-1} \cup \Sigma_{n-1}),$$

which amounts to Rabin's result for $n = 2$, fails for all $n \geq 3$. Consequently, it is in general not possible to separate two disjoint sets in the class Σ_n by a set in $Comp(\Pi_{n-1} \cup \Sigma_{n-1})$; similarly for Π_n.

There is however another plausible generalisation of Rabin's result suggested by the analogy with descriptive set theory. Letting

$$\Delta_n = \Pi_n \cap \Sigma_n,$$

we may ask if two disjoint sets in a class Σ_n can be separated by a set in Δ_n; a similar question can be stated for Π_n. By remarks above, we know that the separation property in this sense holds for Π_2 (Büchi) and fails for Σ_2 (co-Büchi) class, in a perfect analogy with the properties of analytic vs. co-analytic classes in the descriptive set theory[1].

In the present paper we answer the question negatively for all classes Σ_n of the Rabin–Mostowski index hierarchy for alternating automata on infinite trees. (The Σ_n-classes correspond to the indices (i, k) with k odd; see the definition below.) By an analogy with the Borel hierarchy [7, 8], one is tempted to conjecture that the separation property actually does hold for all classes Π_n, but this question seems to be difficult already for $n = 3$.

To prove our main result, we first study a conceptually simpler case of infinite words and the Rabin–Mostowski index hierarchy of deterministic automata. In this case we give a complete answer: the separation property holds for classes Π_n and fails for classes Σ_n. The argument is based on a uniform construction of an inseparable pair in each class Σ_n. This construction is further used in the case of trees. More specifically, we consider labeled trees whose vertices are divided between two players: Eve and Adam, who wish to form a path in the tree. For a set on infinite words L, we consider the set $Win^{\exists}(L)$ of those trees where Eve has a strategy to force a path into L. The operation $Win^{\exists}$ allows us to shift the witness family from words to trees.

It should be noted that in the case of deterministic automata on infinite words, the separation property of the class $(1, 2)$ was proved earlier by Selivanov [12], who also gave a hint [13] how this result can be generalized for all classes Π_n.

2 Index hierarchy

Throughout the paper, ω stands for the set of natural numbers, which we identify with its ordinal type. We also identify a natural number $n < \omega$ with the set $\{0, 1, \ldots, n-1\}$.

[1] However, the classical notation plays a trick here, as the analogy matches the classes $\mathbf{\Sigma}^1_1 \sim \Pi_2$ and $\mathbf{\Pi}^1_1 \sim \Sigma_2$.

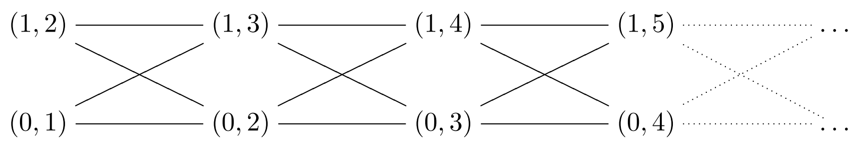

Figure 1 The Mostowski–Rabin index hierarchy.

We will consider deterministic automata on infinite words and alternating automata on infinite trees. For more background, we refer the reader to a survey by Thomas [14].

A *deterministic parity automaton* over an input alphabet A can be presented by $\mathcal{A} = \langle A, Q, q_I, Tr, rank\rangle$, where Q is a finite set of *states* ranked by the function $rank : Q \to \omega$, and $Tr : Q \times A \to Q$ is a *transition* function. A *run* of $\mathcal{A}$ on a word $u \in A^\omega$ is a word $r \in Q^\omega$ whose first element r_0 is the *initial state* q_I, and $r_{n+1} = Tr(r_n, u_n)$, for $n < \omega$. It is *accepting* if the highest rank occurring infinitely often (i.e., $\limsup_{n\to\infty} rank(r_n)$) is *even*. The language $L(\mathcal{A})$ recognized by $\mathcal{A}$ consists of those words $u \in A^\omega$ which admit an accepting run. The *Rabin-Mostowski index* of $\mathcal{A}$ is the pair $(\min rank(Q), \max rank(Q))$; we may assume without loss of generality that $\min rank(Q)$ is 0 or 1. It is useful to partially order the indices as represented on Figure 1. That is, we let $(\iota, \kappa) \sqsubseteq (\iota', \kappa')$ if either $\{\iota, \ldots, \kappa\} \subseteq \{\iota', \ldots, \kappa'\}$, or $\iota = 0$, $\iota' = 1$, and $\{\iota+2, \ldots, \kappa+2\} \subseteq \{\iota', \ldots, \kappa'\}$. We consider the indices $(1, \kappa)$ and $(0, \kappa-1)$ as *dual*, and let $\overline{(\iota, \kappa)}$ denote the index dual to (ι, κ). The above ordering induces a hierarchy, that is, if a language L is recognized by an automaton of index (ι, κ) and $(\iota, \kappa) \sqsubseteq (\iota', \kappa')$ then L is also recognized by an automaton of index (ι', κ'). Moreover, the hierarchy is *strict* in the sense that, for any index (ι, κ), there is a language recognized by an automaton of index (ι, κ), but not by any (deterministic) automaton of the dual index $\overline{(\iota, \kappa)}$ [6, 15]. Indeed, the witness can be the parity condition itself:

$$L_{\iota,\kappa} \quad = \quad \{u \in \{\iota, \ldots, \kappa\}^\omega : \limsup_{n\to\infty} u_n \text{ is even } \}. \tag{1}$$

The concept of alternating automaton is best understood via parity games. For the sake of further application, we present them in a more general setting of *graph games*. A graph game is a perfect information game of two players, say Eve and Adam, where plays may have infinite duration. It can be presented by a tuple

$$\langle V_\exists, V_\forall, Move, p_I, \ell, A, L_\exists, L_\forall\rangle.$$

Here $V_\exists$ and $V_\forall$ are (disjoint) sets of positions of Eve and Adam, respectively, $Move \subseteq V \times V$ is the relation of possible moves, with $V = V_\exists \cup V_\forall$, $p_I \in V$ is a designated initial position, and $\ell : V \to A$ is a labelling function, with some alphabet A. These items constitute an arena of the game. Additionally, $L_\exists, L_\forall \subseteq A^\omega$ are two disjoint sets representing the *winning criteria* for Eve and Adam, respectively.

The players start a play in the position p_I and then move the token according to the relation *Move* (always to a successor of the current position), thus forming a path in the arena. The move is selected by Eve or Adam, depending on who the owner of the current position is. If a player cannot move, she/he looses. Otherwise, the result of the play is an infinite path $v_0, v_1, v_2, \ldots$, inducing the sequence of labels $\ell(v_0), \ell(v_1), \ell(v_2), \ldots$ If this sequence belongs to $L_\exists$ then Eve wins, if it belongs to $L_\forall$ then Adam wins; otherwise there is a draw. We say that Eve *wins* the game if she has a winning strategy, the similar for Adam.

In the games considered in this paper, we always have $L_\forall = A^\omega - L_\exists$, hence a draw will not occur. But it is convenient to consider the winning criteria for both players.

A *parity game of index* (ι, κ) is defined as above with $A = \{\iota, \ldots, \kappa\}$, $L_\exists = L_{\iota,\kappa}$ (see equation (1)), and $L_\forall = \overline{L_\exists} = \{u \in \{\iota, \ldots, \kappa\}^\omega : \limsup_{n\to\infty} u_n$ is odd $\}$.

A (full) *k-ary tree* over a finite alphabet A is a mapping $t : k^* \to A$. An *alternating parity tree automaton of index* (ι, κ) running on such trees can be presented by

$$\mathcal{A} = \langle A, Q_\exists, Q_\forall, q_I, \delta, \text{rank}\rangle$$

where Q is a finite set of states with an initial state q_I, partitioned into existential states $Q_\exists$ and universal states $Q_\forall$, $\delta \subseteq Q \times A \times \{0, 1, \ldots, k-1, \varepsilon\} \times Q$ is a transition relation, and *rank* : $Q \to \omega$. An input tree t is accepted by $\mathcal{A}$ iff Eve has a winning strategy in the parity game

$$G(\mathcal{A}, t) \quad = \quad \langle Q_\exists \times k^*, Q_\forall \times k^*, (q_0, \varepsilon), \text{Mov}, \ell, L_{\iota,\kappa}, \overline{L_{\iota,\kappa}}\rangle, \tag{2}$$

where $\text{Mov} = \{((p, v), (q, vd)) \colon v \in \text{dom}(t),\ (p, t(v), d, q) \in \delta\}$ and $\ell(q, v) = \text{rank}(q)$. Intuitively, the players follow a path in the tree t, additionally annotated by the states. A transition is always selected by the owner of the state. The automaton accepts the tree if Eve can force each such path to be accepting.

The hierarchy of tree languages induced by the indices of alternating parity automata is strict, as showed by Bradfield [4]. An alternative proof of this difficult result was later given [1] based on the Banach Fixed-Point Theorem; Both proofs [1, 4] use the same witness family of sets of binary trees defined by parity games. For the sake of further application, we will present this concept in a more general setting.

We consider k-ary trees over an alphabet $\{\exists, \forall\} \times A$. The labels $\{\exists, \forall\}$ are used to partition the nodes of a tree t into positions of Eve and Adam. In the game, the players form a path in t, starting from the root. The next move is selected by Eve or Adam, depending on whether the actual label contains $\exists$ or $\forall$. The winning criteria concern the sequence formed by the second components of the labels occurring in the play, which is a word in A^ω .

Each language $L \subseteq A^\omega$ induces two winning criteria: $L_\exists = L$, and $L_\forall = L$, which give rise to two games: an *L-$\exists$ game*, and an *L-$\forall$ game*. Let us describe formally an L-$\exists$ game over a tree $t : k^* \to \{\exists, \forall\} \times A$. It is a graph game with the following items:

$$\begin{array}{lcllcl} V_\exists & = & \{v \in k^* : t(v)\downarrow_1 = \exists\} & \ell(v) & = & t(v)\downarrow_2,\ \text{for } v \in k^* \\ V_\forall & = & \{v \in k^* : t(v)\downarrow_1 = \forall\} & L_\exists & = & L \\ \mathit{Move} & = & \{(w, wi) : w \in k^*,\ i \in k\} & L_\forall & = & \overline{L}. \\ p_0 & = & \varepsilon \text{ (the root of the tree)} & & & \end{array}$$

An L-$\forall$ game is defined similarly with the winning criteria $L_\forall = L$, and $L_\exists = \overline{L}$.

The set $\mathit{Win}_k^\exists(L)$ consists of those trees t, for which Eve has a winning strategy in L-$\exists$-game. The set $\mathit{Win}_k^\forall(L)$ consists of those trees t, for which Adam has a winning strategy in L-$\forall$-game. The following can be easily verified.

▶ Fact 1. If a language L of infinite words is recognized by a deterministic automaton of index (ι, κ) then both languages $\mathit{Win}_k^\exists(L)$ and $\mathit{Win}_k^\forall(L)$ can be recognized by an alternating[2] tree automaton of index (ι, κ).

[2] In fact, even *non-deterministic*, but we don't explore it in this paper.

A family witnessing the strictness of the index hierarchy of alternating tree automata [1, 4] consists of the sets of binary trees $W_{i,k}$, which can be presented by $W_{i,k} = Win_2^{\exists}(L_{i,k})$.

The following Σ/Π terminology for the index hierarchy,motivated by the connection with the μ-calculus (see, e.g., [3]); will be convenient to handle dualities. For each $m \geq 1$, we consider two indices: $(1, m)$ and $(0, m-1)$, and associate the symbol Σ_m with this index whose maximum is odd, and Π_m with the one whose maximum is even. For example, $(0,1) \approx \Sigma_2$, $(1,2) \approx \Pi_2$, $(1,3) \approx \Sigma_3$, $(0,2) \approx \Pi_3$, $(1,4) \approx \Pi_4$, $(0,3) \approx \Sigma_4$, etc. We will then refer to an automaton of index (ι, κ) as to Σ_m or Π_m-automaton with an appropriate m.

3 Deterministic hierarchy over words

In this section we investigate the index hierarchy for deterministic automata on infinite words. A language of infinite words is in the class Σ_m if it is recognized by a deterministic Σ_m-automaton; similarly for Π_m. A language is in the class Δ_m if it is simultaneously Σ_m and Π_m. We show that the separation property holds for classes Π_m and fails for Σ_m, for $m \geq 2$. In fact, both properties will follow from a single construction (parametrized by m).

Note that we do not consider the classes Σ_1 and Π_1, which are uninteresting from the point of view of the separation property[3].

For $m \geq 2$, we fix an alphabet

$$m_\pi = \begin{cases} \{1, \dots, m\} & \text{if } m \text{ is even} \\ \{0, \dots, m-1\} & \text{if } m \text{ is odd.} \end{cases}$$

Note that $\max m_\pi$ is always even. Let $I_m \subseteq m_\pi^\omega$ be the set of infinite words where $\max m_\pi$ occurs infinitely often. Let K_m be a superset of I_m consisting of the words satisfying parity condition,

$$K_m \quad = \quad \{u \in m_\pi^\omega : \limsup_{n \to \infty} u_n \text{ is even } \}.$$

That is, K_m coincides with $L_{1,m}$ or $L_{0,m-1}$ of (1), depending on whether m is even or odd. It is straightforward to see that K_m is in the class Π_m.

In the sequel we consider words over a product alphabet m_π^2. We identify a pair of words $\langle u, v \rangle \in (m_\pi^\omega)^2$ with a single word over (m_π^2) in an obvious manner. The subsequent lemma is the heart of our paper.

▶ **Lemma 1.** *For each $m \geq 2$, there exist disjoint sets $U_1, U_2 \subseteq m_\pi^\omega$ of class Σ_m, satisfying the following:*

$$\begin{array}{rcl} K_m \times \overline{K_m} & \subseteq & U_1 \\ \overline{K_m} \times K_m & \subseteq & U_2 \\ \overline{K_m \times K_m} & \subseteq & U_1 \cup U_2 \quad = \quad \overline{I_m \times I_m}. \end{array}$$

Proof. We first present the construction in the case when m is odd; thus the Σ_m-automata have index $(1, m)$, and Π_m-automata have index $(0, m-1)$.

Let P_m be an automaton over the alphabet m_π with the set of states also equal to m_π, and the transition function $Tr(q, s) = s$, for any q and s. (We leave the remaining items temporarily unspecified.) Let $P_m \times \top$ be an automaton over m_π^2 which behaves like P_m reading only the first component. Similarly, $\top \times P_m$ reads only the second component.

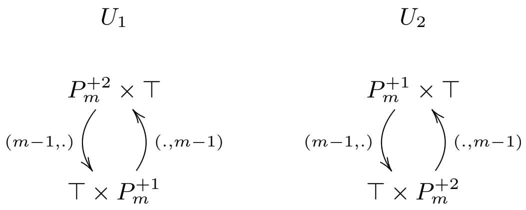

Figure 2 Automata for U_1 and U_2 for m odd.

We represent the Σ_m-automata recognizing U_1 and U_2 on Figure 2. The states of the automaton for U_1 are $\{0, 1, \ldots, m-1\}$ (upper component) and $\{0', 1', \ldots, (m-1)'\}$ (lower component). In its upper component, the automaton reads the left component of the input symbols until it eventually encounters a symbol $(m-1, s)$, for some s. Then the edge is directed to the state $(m-1)'$ in the lower component. Here the automaton reads the right component of the input symbols until it eventually encounters a symbol $(s, m-1)$, for some s, in which case it moves to the state $m-1$ in the upper component. The ranks in the upper component are $\mathit{rank}(i) = i+2$, for $i = 0, 1, \ldots, m-2$, and $\mathit{rank}(m-1) = m$. The ranks in the lower component are $\mathit{rank}(i') = i+1$, for all i. For the initial state we set 0.

The automaton for U_2 is defined analogously; the difference concerns only rankings[4], namely $\mathit{rank}(i) = i+1$, for all i, and $\mathit{rank}(i') = i+2$, for $i = 0, 1, \ldots, m-2$ and $\mathit{rank}((m-1)') = m$. Note that the states $m-1$ and $(m-1)'$ can be reached only while changing the levels and, whenever it happens, both automata assume the highest odd rank m.

Each word $u \in \left(m_\pi^2\right)^\omega$ induces the same run in both automata up to the rankings. Clearly, a word u causes infinitely many changes of the level if and only if it contains infinitely many occurrences of $m-1$ on both left and right track. By remark above, such a word is accepted by neither of the automata. On the other hand, if the run on u stabilizes on some level then one of the automata necessarily accepts, as the ranks they assume in their runs (after stabilization) differ precisely by 1.

This shows that U_1 and U_2 are disjoint and $U_1 \cup U_2 = \overline{I_m \times I_m}$. The inclusion $\overline{K_m \times K_m} \subseteq \overline{I_m \times I_m}$ is obvious.

If $u \in K_m \times \overline{K_m}$ then the run on u stabilizes in the upper or lower component (as clearly $u \notin I_m \times I_m$). Then the automaton for U_1, from some moment on, either reads a word in K_m in the upper component or a word in $\overline{K_m}$ in the lower component; in either case it accepts. A similar argument shows the second inclusion, which completes the proof of the lemma in case m is odd. The construction for the case of m even is analogous. We leave it to the reader with Figure 3 as a hint.

◀

The properties of U_1 and U_2 mentioned in Lemma 1 imply a kind of hardness of these sets. Generally, for an automaton $\mathcal{A}$ over an alphabet A of some index (ι, κ), let $\mathit{rank}^{\mathcal{A}}$ denote the function sending a word $u \in A^\omega$ onto the sequence of ranks assumed by $\mathcal{A}$. More precisely,

---

[3] By definition, a deterministic automaton of index $(1,1)$ accepts no words, and an automaton of index $(0,0)$ accepts all words.

[4] The somewhat awkward exceptions in ranking of $(m-1)$ in the first automaton and $(m-1)'$ in the second follow from our desire of having the graphs of both automata the same. Otherwise we could merge the "nasty" states with their companions.

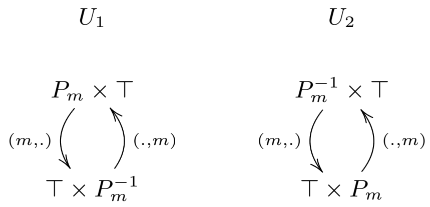

Figure 3 Automata for U_1 and U_2 for m even.

$rank^{\mathcal{A}} : A^\omega \to \{\iota, \dots, \kappa\}^\omega$, and

$$rank^{\mathcal{A}}(u) \quad = \quad rank(r_0)rank(r_1)\dots, \tag{3}$$

where $r_0 r_1 \dots$ is the unique run of $\mathcal{A}$ on u. If $\mathcal{B}$ is another automaton over A with index (ι, κ), we define $rank^{\mathcal{A}\times\mathcal{B}} : A^\omega \to \left(\{\iota, \dots, \kappa\}^2\right)^\omega$, by

$$rank^{\mathcal{A}\times\mathcal{B}}(u) \quad = \quad \langle rank^{\mathcal{A}}(u), rank^{\mathcal{B}}(u)\rangle.$$

▶ **Lemma 2.** *Let $\mathcal{A}$ and $\mathcal{B}$ be automata of class Π_m over some alphabet A, such that $L(\mathcal{A}) \cap L(\mathcal{B}) = \emptyset$. Let U_1 and U_2 satisfy the properties of Lemma 1. Then, for each word $u \in A^\omega$,*
1. *if $u \in L(\mathcal{A})$ then $rank^{\mathcal{A}\times\mathcal{B}}(u) \in U_1$,*
2. *if $u \in L(\mathcal{B})$ then $rank^{\mathcal{A}\times\mathcal{B}}(u) \in U_2$,*
3. *$rank^{\mathcal{A}\times\mathcal{B}}(u) \in U_1 \cup U_2$.*

Proof. Generally, if $\mathcal{D}$ is a deterministic parity automaton of index (ι, κ) then, by definition of acceptance,

$$u \in L(\mathcal{D}) \quad \Leftrightarrow \quad rank^{\mathcal{D}}(u) \in L_{\iota,\kappa}.$$

Hence, in our case, $u \in L(\mathcal{A}) \Rightarrow rank^{\mathcal{A}}(u) \in K_m$, and $u \notin L(\mathcal{B}) \Rightarrow rank^{\mathcal{B}}(u) \in \overline{K_m}$. As $L(\mathcal{A})$ and $L(\mathcal{B})$ are disjoint, $u \in L(\mathcal{A})$ implies $rank^{\mathcal{A}\times\mathcal{B}}(u) \in K_m \times \overline{K_m}$, but we know from Lemma 1 that $K_m \times \overline{K_m} \subseteq U_1$. The argument for 2 is similar. Finally, again by disjointness of $L(\mathcal{A})$ and $L(\mathcal{B})$, we have $rank^{\mathcal{A}\times\mathcal{B}}(u) \in \overline{K_m \times K_m}$, for any u, but we know from Lemma 1 that $\overline{K_m \times K_m} \subseteq U_1 \cup U_2$, which completes the proof. ◀

We are now ready to state the main result of this section. Recall that the separation property for the class Π_2 was proved earlier by Selivanov [12], who also gave[5] a hint [13] how this result can be generalized for all classes Π_n.

▶ **Theorem 3.** *The separation property holds for classes Π_m and fails for classes Σ_m of the index hierarchy of deterministic word automata.*

Proof. We will show that any pair of disjoint languages of class Π_m over some finite alphabet A is separable by a language of class Δ_m, whereas this property fails for the pair of sets U_1, U_2 of Lemma 1 (which are of class Σ_m).

Let $\mathcal{A}$ and $\mathcal{B}$ be as in Lemma 2. It follows from 1 and 2 that the inverse image of U_1 under the mapping $rank^{\mathcal{A}\times\mathcal{B}}$, i.e.,

$$\left(rank^{\mathcal{A}\times\mathcal{B}}\right)^{-1}(U_1) \quad = \quad \{u \in A^\omega : rank^{\mathcal{A}\times\mathcal{B}}(u) \in U_1\}$$

[5] More precisely, that author considered the *reduction* property for the dual classes Σ_m. See a comment after Proposition 3.5 in [13].

separates $L(\mathcal{A})$ and $L(\mathcal{B})$. Let us see that this set is recognizable by an Σ_m-automaton. For an input u, we just use the automaton for U_1 reading the subsequent values of the function $rank^{\mathcal{A}\times\mathcal{B}}$; the construction is straightforward. In a similar vein we can show that $\left(rank^{\mathcal{A}\times\mathcal{B}}\right)^{-1}(U_2)$ is in the class Σ_m as well. Clearly these sets are disjoint as U_1 and U_2 are disjoint. But it follows from condition 3 of Lemma 2 that they sum up to A^ω, hence they both are of class Δ_m.

To show that U_1 and U_2 are inseparable, we start with the following observation. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are Π_m-automata over the alphabet m_π^2. Then the function $rank^{\mathcal{A}\times\mathcal{B}}$, which in this case has type $rank^{\mathcal{A}\times\mathcal{B}} : (m_\pi^2)^\omega \to (m_\pi^2)^\omega$, has a fixed point. Indeed, a fixed point f can be defined[6] by an inductive formula

$$
\begin{aligned}
f_0 &= \left(rank(q_I^{\mathcal{A}}), rank(q_I^{\mathcal{B}})\right) \\
f_{n+1} &= \left(rank\left(\hat{Tr}^{\mathcal{A}}(q_I^{\mathcal{A}}, f_0 \dots f_n)\right), rank\left(\hat{Tr}^{\mathcal{B}}(q_I^{\mathcal{B}}, f_0 \dots f_n)\right)\right)
\end{aligned}
$$

(where $\hat{Tr}$ is the standard extension of Tr from letters to finite words).

Now suppose, for the sake of contradiction, that there is a set $C \subseteq m_\pi^\omega$ of class Δ_m, such that $U_1 \subseteq C$ and $U_2 \subseteq \overline{C}$. Let $\mathcal{A}$ and $\mathcal{B}$ be two automata of class Π_m, such that

$$
\begin{aligned}
L(\mathcal{A}) &= \overline{C} \\
L(\mathcal{B}) &= C.
\end{aligned}
$$

By remark above, the function $rank^{\mathcal{A}\times\mathcal{B}}$ has the fixed point $(f_0, f_1, \dots)$. On the other hand, by conditions 1 and 2 of Lemma 2, it reduces $\overline{C}$ to $U_1 \subseteq C$, and C to $U_2 \subseteq \overline{C}$. This contradiction completes the proof. ◀

4 Alternating hierarchy over trees

In this section we investigate the Rabin-Mostowski index hierarchy for alternating automata on k-ary trees. A tree language is in the class Σ_m if it is recognized by an alternating Σ_m-automaton; similarly for Π_m. A tree language is in the class Δ_m if it is simultaneously Σ_m and Π_m. We show that the separation property fails in general for the classes Σ_m, for $m \geq 1$.

It will be convenient to have some normal form of alternating automata. We call an alternating automaton on k-ary trees, $\mathcal{A} = \langle A, Q_\exists, Q_\forall, q_I, \delta, \mathrm{rank}\rangle$, an $\exists\forall$-automaton if it satisfies the following conditions:

1. $q_I \in Q_\exists$,
2. if $(p, s, d, q) \in \delta$ is a transition then $p \in Q_\exists$ iff $q \in Q_\forall$,
3. for any pair $(p, s) \in Q \times A$, there are *exactly* two transitions $(p, s, d, q), (p, s, d', q') \in \delta$.

These conditions imply that the graph of the game $G(\mathcal{A}, t)$ (see equation (2)) unravels to a full binary tree, where $\exists$ and $\forall$ alternate starting with $\exists$.

We will focus on the ranks of the states occurring in this tree. More precisely, let the index of $\mathcal{A}$ be (ι, κ). With any tree $t : k^* \to A$, we associate a binary tree

$$\mathcal{T}(\mathcal{A}, t) : 2^* \to \{\exists, \forall\} \times \{\iota, \dots, \kappa\}$$

[6] The existence and uniqueness of this fixed point can be also inferred from the Banach Fixed-Point Theorem.

as follows. We define an auxiliary function $\gamma : 2^* \to Q \times k^*$, using the notation $own(q) = \xi \in \{\exists, \forall\}$, whenever $q \in Q_\xi$. The value of γ represents the state and the current position in the game-play on the tree t. Let $\gamma(\varepsilon) = (q_I, \varepsilon)$, and $\mathcal{T}(\mathcal{A}, t)((\varepsilon) = (own(q_I), rank(q_I))$. Suppose $\mathcal{T}(\mathcal{A}, t)(v)$ and $\gamma(v)$ are defined, say $\gamma(v) = (p, w)$. By condition 3 above, there are exactly two pairs that extend $(p, t(w))$ to a transition in δ. Suppose they are (d, q) and (d', q') (in this pre-defined order, for definiteness). We let $\gamma(v0) = (q, wd)$ and $\gamma(v1) = (q', wd')$. We further define $\mathcal{T}(\mathcal{A}, t)(v0) = (own(q), rank(q))$ and $\mathcal{T}(\mathcal{A}, t)(v1) = (own(q'), rank(q'))$. It is straightforward to see that

$$t \in L(\mathcal{A}) \quad \Leftrightarrow \quad \mathcal{T}(\mathcal{A}, t) \in W_{\iota,\kappa} \tag{4}$$

(see page 400 for the definition of $W_{\iota,\kappa}$). We leave to the reader the proof of the following simple observation.

▶ **Lemma 4.** *Any alternating tree automaton can be transformed to an $\exists\forall$-automaton of the same index, recognizing the same language.*

A $\forall\exists$-automaton is defined similarly, with the only difference that $q_I \in Q_\forall$. Clearly, an analogue of Lemma 4 for $\forall\exists$-automata holds as well.

We are going to define a tree version of the "hard pairs" from Section 3. Let $m \geq 1$, and let U_1 and U_2 be the languages defined in the proof of Lemma 1. We let

$$\begin{aligned} \nabla_1 &= Win_4^\exists(U_1) \\ \nabla_2 &= Win_4^\forall(U_2). \end{aligned}$$

By Fact 1, the sets ∇_1 and ∇_2 are of class Σ_m. To have some analogue of Lemma 2, we need some analogue of the function $rank^{\mathcal{A}\times\mathcal{B}}$; we will define it only for automata in a special form.

We call a tree $t : k^* \to \{\exists, \forall\} \times A$ a $\exists\forall$-tree if

$$t(v) \downarrow_1 = \begin{cases} \exists & \text{if } |v| \text{ is even} \\ \forall & \text{if } |v| \text{ is odd.} \end{cases}$$

The concept of a $\forall\exists$-tree is defined analogously. Note that the trees $\mathcal{T}(\mathcal{A}, t)$ defined above are $\exists\forall$-trees or $\forall\exists$-trees, whenever $\mathcal{A}$ is an $\exists\forall$-automaton or $\forall\exists$-automaton, respectively.

At first, we define a *product* of a binary $\exists\forall$-tree t_1 and a binary $\forall\exists$ tree t_2 as a 4-ary $\exists\forall$-tree $t_1 \star t_2 : 4^* \to \{\exists, \forall\} \times A \times A$. It is convenient to fix some bijection $4 \sim 2 \times 2$, so that a (finite) word w in 4^* can be identified with a pair of words v, u in 2^* of the same length (such that $w_i = (v_i, u_i)$); we then use the notation $w = (u \circ v)$. We then let[7]

$$t_1 \star t_2(u \circ v) \quad = \quad (t_1(u) \downarrow_1, t_1(u) \downarrow_2, t_2(v) \downarrow_2)\,.$$

Now fix k and an alphabet A. Let $\mathcal{A}$ and $\mathcal{B}$ be two automata on k-ary trees over the alphabet A, both of the class Π_m. Assume moreover that $\mathcal{A}$ is an $\exists\forall$-automaton and $\mathcal{B}$ a $\forall\exists$-automaton. For a tree $t : k^* \to A$, consider an $\exists\forall$-tree $\mathcal{T}(\mathcal{A}, t)$ and a $\forall\exists$-tree $\mathcal{T}(\mathcal{B}, t)$. We let

$$g^{\mathcal{A}\times\mathcal{B}}(t) \quad = \quad (\mathcal{T}(\mathcal{A}, t) \star \mathcal{T}(\mathcal{B}, t)). \tag{5}$$

The following is a (partial) analogue of Lemma 2.

[7] Figure 4 at the end of the paper shows how from a green $\exists\forall$ tree t_1 and a red $\forall\exists$ tree t_2 we obtain a blue 4-ary tree $t_1 \star t_2$.

▶ **Lemma 5.** *With the notations above,*
1. *if* $t \in L(\mathcal{A})$ *then* $g^{\mathcal{A}\times\mathcal{B}}(t) \in \nabla_1$,
2. *if* $t \in L(\mathcal{B})$ *then* $g^{\mathcal{A}\times\mathcal{B}}(t) \in \nabla_2$.

Proof. Assume that $t \in L(\mathcal{A})$. It implies, that Eve has a winning strategy $\sigma_{\mathcal{A}}$ showing that $\mathcal{T}(\mathcal{A}, t) \in W_{\iota,\kappa}$. Since $L(\mathcal{A})$ and $L(\mathcal{B})$ are disjoint, Adam has a winning strategy $\sigma_{\mathcal{B}}$ showing that $\mathcal{T}(\mathcal{B}, t) \notin W_{\iota,\kappa}$.

Let σ be the strategy for Eve on the tree $\mathcal{T}(\mathcal{A}, t) \star \mathcal{T}(\mathcal{B}, t)$ which combines $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$. Namely, $\sigma_{\mathcal{A}}$, $\sigma_{\mathcal{B}}$ choose one of the two options available for respectively Eve and Adam and σ chooses the uniquely defined combination of these two options. Since $\sigma_{\mathcal{A}}$ leads to a sequence in K_m, $\sigma_{\mathcal{B}}$ leads to a sequence in the complement of K_m, the resulting sequence defined by σ belongs to $K_m \times \overline{K_m}$, in particular it belongs to U_1.

Assume now that $t \in L(\mathcal{B})$. It implies, that Eve has a winning strategy $\sigma_{\mathcal{B}}$ showing that $\mathcal{T}(\mathcal{B}, t) \in W_{\iota,\kappa}$. Since $L(\mathcal{A})$ and $L(\mathcal{B})$ are disjoint, Adam has a winning strategy $\sigma_{\mathcal{A}}$ showing that $\mathcal{T}(\mathcal{A}, t) \notin W_{\iota,\kappa}$.

Let σ be the strategy for Adam on the tree $\mathcal{T}(\mathcal{A}, t) \star \mathcal{T}(\mathcal{B}, t)$ which combines $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$. Namely, $\sigma_{\mathcal{A}}$, $\sigma_{\mathcal{B}}$ choose one of the two options available for respectively Adam and Eve and σ chooses the uniquely defined combination of these two options. Since $\sigma_{\mathcal{A}}$ leads to a sequence in the complement of K_m, $\sigma_{\mathcal{B}}$ leads to a sequence in K_m, the resulting sequence defined by σ belongs to $\overline{K_m} \times K_m$, in particular it belongs to U_2. ◀

The reader may have noticed that the point 3 of Lemma 2 is missing in Lemma 5. This is precisely why we fail to extend the positive results on the classes Π_m from words to trees.

We can state the main result of the paper.

▶ **Theorem 6.** *The separation property fails for classes* Σ_m *of the index hierarchy of alternating tree automata. More specifically, for any* $m \geq 2$*, there exists a pair of sets of 4-ary trees of class* Σ_m *inseparable by any set of class* Δ_m.

Proof. The proof is similar to the proof of Theorem 3. To show that ∇_1 and ∇_2 are inseparable, we start with the following observation. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are Π_m-automata over the alphabet $\{\exists, \forall\} \times (\iota, \kappa)^2$. Suppose moreover that $\mathcal{A}$ is an $\exists\forall$-automaton and $\mathcal{B}$ is a $\forall\exists$-automaton. We will show that the mapping $g^{\mathcal{A}\times\mathcal{B}}$ has a fixed point t. Since the range of the mapping $g^{\mathcal{A}\times\mathcal{B}}$ consists of $\exists\forall$ trees, it implies the first coordinate of $t(u \circ v)$ has to be $\exists$ for $u \circ v$ of even length and $\forall$ otherwise. The second and third coordinates of $t(u \circ v)$ for $u = u_0 \dots u_{n+1}$ and $v = v_0 \dots v_{n+1}$ will be defined along the same lines as in the proof of Theorem 3, however we have to take care of ϵ-transitions. For this sake we will define the token mappings $\gamma_{\mathcal{A}}$ and $\gamma_{\mathcal{B}}$ like the token mapping γ used in the definition of $\mathcal{T}(\mathcal{A}, t)$ (see (4)). The mappings will be defined successively together with the tree t.

Let us define

$$t(\varepsilon)\downarrow_2 = rank(q_I^{\mathcal{A}}), t(\varepsilon)\downarrow_3 = rank(q_I^{\mathcal{B}}).$$

The $\mathcal{A}$- and $\mathcal{B}$- tokens are placed in the root. In automaton $\mathcal{A}$ there are two transitions from q_I on the letter $t(\varepsilon)$. Similarly, there are two transitions in $\mathcal{B}$. The root of t has four successors uniquely defined by these two pairs of transitions. Assume now that Eve in $\mathcal{A}$ and Adam in $\mathcal{B}$ made their first moves. These two moves uniquely define a vertex $u_0 \circ v_0$ in the tree t, that is one of the four successors of the root of t. If Eve decided for an ϵ-transition then the second coordinate of $\gamma_{\mathcal{A}}$ remains unchanged. Otherwise we move the token to $u_0 \circ v_0$. Similarly if Adam decided for an ϵ-transition then the second coordinate of $\gamma_{\mathcal{B}}$ remains unchanged, otherwise we move the token to $u_0 \circ v_0$. In automaton $\mathcal{A}$ there are two transitions from the state $\gamma_{\mathcal{A}}(u_0)\downarrow_1$ on the letter $t(\gamma_{\mathcal{A}}(u_0)\downarrow_2)$. Similarly, there are two transitions in $\mathcal{B}$ from the state $\gamma_{\mathcal{A}}(v_0)\downarrow_1$ on the letter $t(\gamma_{\mathcal{B}}(v_0)\downarrow_2)$. The vertex $t(u_0 \circ v_0)$ has four successors uniquely defined by these two pairs of transitions. We extend $\gamma_{\mathcal{A}}$ and γ_{B} accordingly and continue building a full 4-ary tree t.

It is easy to verify that the tree t is a fixed point[8] of $g^{\mathcal{A}\times\mathcal{B}}$. Now suppose, for the sake of contradiction, that there exists a set C of trees over the alphabet $\{\exists,\forall\}\times(\iota,\kappa)^2$ such that

- C belongs to the class Δ_m,
- $\nabla_1 \subseteq C$ and $\nabla_2 \subseteq \overline{C}$.

Let $\mathcal{A}$ and $\mathcal{B}$ be two automata of class Π_m, such that

$$\begin{aligned} L(\mathcal{A}) &= \overline{C} \\ L(\mathcal{B}) &= C. \end{aligned}$$

By the remark above, the function $g^{\mathcal{A}\times\mathcal{B}}$ has the fixed point t. On the other hand, by conditions 1 and 2 of Lemma 5, it reduces $\overline{C}$ to $\nabla_1 \subseteq C$, and C to $\nabla_2 \subseteq \overline{C}$. This contradiction completes the proof. ◀

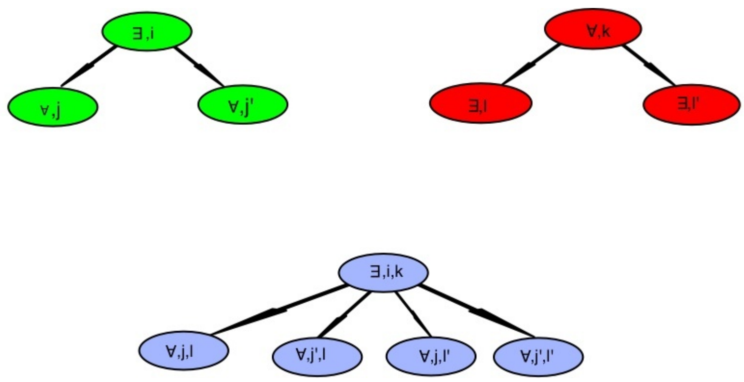

Figure 4 Operation $\star$ on a green $\exists\forall$ tree t_1 and a red $\forall\exists$ tree t_2 gives the blue 4-ary tree.

The consideration of 4-ary trees in Theorem 6 made the proof more transparent, but the result can be adapted to binary trees as well.

▶ **Corollary 7.** *There exists a pair of sets of binary trees of class Σ_m inseparable by any set of class Δ_m.*

Sketch of proof. Let ∇_1, ∇_2 be as in Theorem 6. We define languages V_1, V_2 of binary trees and a mapping η such that $\nabla_i = \eta^{-1}[V_i]$. Let V_1 consist of binary trees t over the alphabet $\{\exists,\forall\}\times m_\pi$ such that

1. the first two levels of t, that is the root and its children, belong to Eve, as well as all the levels $4k$, $4k+1$, for $k = 0, 1, 2, \ldots$,
2. the levels $4k+2$, $4k+3$, for $k = 0, 1, 2, \ldots$, belong to Adam,
3. Eve has a strategy, such that if the sequence $(\exists, w_0), (\exists, v_0), (\forall, w_1), (\forall, v_1), (\exists, w_2), (\exists, v_2), \ldots$ represents a game-play then $(w_0, v_0), (w_1, v_1), (w_2, v_2), \ldots$ belongs to U_1.

[8] As in Theorem 3, the existence and uniqueness of this fixed point can be also inferred from the Banach Fixed-Point Theorem.

The set V_2 satisfies the same conditions 1, 2, and 3 is replaced by the requirement that Adam has a strategy, which forces represents a game-play $(w_0, v_0), (w_1, v_1), (w_2, v_2), \ldots$ into U_2.

To a 4-ary tree t over the alphabet $\{\exists, \forall\} \times (m_\pi)^2$, we now assign a binary tree $t' = \eta(t)$ over the alphabet $\{\exists, \forall\} \times m_\pi$. As before, it is convenient to use a bijection $4 \sim 2 \times 2$, so that a node of t of level ℓ can be presented by $(x_0, y_0)(x_1, y_1), \ldots, (x_{\ell-1}, y_{\ell-1})$, with $x_i, y_i \in 2$. The root of t' is labeled $(t(\epsilon) \downarrow 1, t(\epsilon) \downarrow 2)$, and the second level is labeled by $(t(\epsilon) \downarrow 1, t(\epsilon) \downarrow 3)$ in both directions. More specifically, whenever

$$t\left((x_0, y_0)(x_1, y_1), \ldots (x_{\ell-1}, y_{\ell-1})\right) \quad = \quad (\xi, a, b),$$

we let

$$\begin{array}{lll} t' \quad (x_0, y_0, x_1, y_1, \ldots, x_{\ell-1}, y_{\ell-1}) & = & (\xi, a) \\ t' \quad (x_0, y_0, x_1, y_1, \ldots, x_{\ell-1}, y_{\ell-1}, z) & = & (\xi, b) \quad \text{for } z = 0, 1. \end{array}$$

It is straightforward to verify that $\nabla_i = \eta^{-1}[V_i]$, for $i = 1, 2$. Suppose that V_1, V_2 are separated by a set C of class Δ_m. It is easy to check that the preimage under the mapping η of a tree language recognized by an alternating automaton of an index (i, n) can be itself recognized by an automaton of the same index. Hence $\eta^{-1}[C]$ is in the class Δ_m and separates ∇_1 and ∇_2, which contradicts Theorem 6. ◀

References

1. Arnold, A., The μ-calculus alternation-depth hierarchy is strict on binary trees. *RAIRO-Theoretical Informatics and Applications* 33 (1999), 329–339.
2. Arnold, A., and Niwiński, D., Fixed point characterization of weak monadic logic definable sets of trees. In M.Nivat, A.Podelski, editors, *Tree Automata and Languages*, Elsevier, 1992, 159-188.
3. Arnold, A., and Niwiński, D., *Rudiments of μ-Calculus.* Elsevier Science, Studies in Logic and the Foundations of Mathematics, 146, North–Holland, Amsterdam, 2001.
4. Bradfield, J.C., Simplifying the modal mu-calculus alternation hierarchy. In: *Proc. STACS'98*, Lect. Notes Comput. Sci. 1373 (1998), 39–49.
5. Hummel, S., Michalewski, H., and Niwiński, D., On the Borel Inseparability of Game Tree Languages. STACS 2009:565-575.
6. Kaminski, M., A Classification of omega-Regular Languages. *Theor. Comput. Sci.* 36: 217-229 (1985).
7. Kechris, A.S., *Classical descriptive set theory.* Springer-Verlag, New York, 1995.
8. Moschovakis, Y. N., *Descriptive Set Theory.* North Holland, 1980.
9. Muller, D.E., Saoudi, A., and Schupp, P.E., Alternating Automata, the Weak Monadic Theory of Trees and its Complexity. *Theoret. Comput. Sci.* 97(2): 233-244 (1992).
10. Rabin, M.O., Weakly definable relations and special automata. In: *Mathematical Logic and Foundations of Set Theory*, Y. Bar-Hillel ed., 1970, 1-23.
11. Santocanale, L., and Arnold, A., Ambiguous classes in μ-calculi hierarchies. *Theoret. Comput. Sci.* 333 (2005), 265-296.
12. Selivanov, V., Fine hierarchy of regular ω-languages. *Theoret. Comput. Sci.* 191 (1998), 37–59.
13. Selivanov, V., Fine hierarchy and m-reducibilities theoretical computer science. *Theoret. Comput. Sci.* 405 (2008), 116–163.
14. Thomas, W., Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Springer-Verlag, 1997, pp. 389–455.
15. Wagner, K., On ω-regular sets. *Information and Control*, 43:123–177 (1979).