Theory Comput Syst (2014) 55:833–855
DOI 10.1007/s00224-013-9461-4

On the Separation Question for Tree Languages

André Arnold · Henryk Michalewski · Damian Niwiński

Published online: 22 March 2013

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). The non-separation result is also adapted to the analogous classes induced by weak alternating automata.

To prove our main 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.

It remains open if the separation property holds for all classes Π_n of the index hierarchy for tree automata. To give a positive answer it would be enough to show the reduction property of the dual classes—a method well-known in descriptive set theory. We show that it cannot work here, because the reduction property fails for all classes in the index hierarchy.

Keywords Alternating tree automata · Separation property · Rabin-Mostowski index

H. Michalewski and D. Niwiński supported by the Polish Ministry of Science grant nr. N N206 567840.

H. Michalewski · D. Niwiński (✉)
Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, Warsaw, Poland
e-mail: D.Niwinski@mimuw.edu.pl

H. Michalewski
e-mail: H.Michalewski@mimuw.edu.pl

H. Michalewski
Institute of Mathematics, Polish Academy of Sciences, Warsaw, Poland

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 $B \cap C = \emptyset$) 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 always be 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., [11] or [9]).

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 [13] in terms of monadic second-order logic; the automata-theoretic statement was given by Muller, Saoudi, and Schupp [12] in terms of *weak* alternating automata. It is not difficult to adapt Rabin's proof to obtain a—slightly stronger—separation property: any two disjoint Büchi recognizable sets of trees can be separated by a weakly recognizable set (see, e.g., [7]). Quite analogical to the co-analytic case, the separation property fails in general for the dual class of co-Büchi tree languages (i.e., the complements of Büchi recognizable sets). In [7], 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 [14]. 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 $\mathrm{Comp}(\Pi_1 \cup \Sigma_1)$ [3]. Somewhat surprisingly, Santocanale and Arnold [14] discovered that the equation

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

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 $\mathrm{Comp}(\Pi_{n-1} \cup \Sigma_{n-1})$; 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]

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

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 classes Σ_n correspond to the indices (i,k) with k odd; see the definition in Sect. 2 below.) By an analogy with the Borel hierarchy [9, 11], 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. It should be noted that the separation property of the class $(1,2)$ was proved earlier by Selivanov [15], who also gave a hint [16] how this result can be generalized for all classes Π_n in the index hierarchy of deterministic automata on infinite words.

The construction made for words 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 a tree. Given a set on infinite words L, we consider the set $\mathrm{Win}^{\exists}(L)$ of those trees where Eve has a strategy to force a path into L. The operation $\mathrm{Win}^{\exists}$ allows us to shift the witness family from words to trees.

The main result is completed by two further observations. We show that the non-separation result holds for the analogous classes in the index hierarchy of the weak alternating automata. Next, in quest to solve the aforementioned conjecture about the classes Π_n, we investigate *reduction property* for tree automata. Roughly, this property means that any union $A \cup B$ can be refined to a disjoint union $A' \cup B'$ of sets of the same complexity. In descriptive set theory, the reduction property of a class of sets is often used to show the separation property of the dual class (see [9]). We show that this method cannot be used here, because the reduction property fails for all non-trivial classes in the index hierarchy of alternating automata on trees.

The preliminary version of this paper was presented to the conference STACS 2012 [2]. The results about weak automata and about reduction property are new and presented for the first time in this paper.

2 Index Hierarchy

Throughout the paper, ω stands for the set of natural numbers, and notations $n \in \omega$ and $n < \omega$ are interchangeable. We identify a natural number $n < \omega$ with the set $\{0, 1, \ldots, n-1\}$; in particular $2 = \{0, 1\}$. An *alphabet* is any set, which is *finite* but *non-empty*.

We will consider deterministic automata on infinite words and alternating automata on infinite trees. The latter will be introduced *via* games. For more background on automata, we refer the reader to a survey by W. Thomas [17].

2.1 Automata on Infinite Words

A *deterministic parity automaton* on infinite words over an input alphabet A can be presented by $\mathcal{A} = \langle A, Q, q_I, \delta, \mathrm{rank}\rangle$, where Q is a finite set of *states* ranked by the

$$\begin{tikzcd}
(1,1) \arrow[r, no head] \arrow[dr, no head] & (1,2) \arrow[r, no head] \arrow[dr, no head] & (1,3) \arrow[r, no head] \arrow[dr, no head] & (1,4) \arrow[r, no head] \arrow[dr, no head] & (1,5) \arrow[r, no head, dotted] \arrow[dr, no head, dotted] & \cdots \\
(0,0) \arrow[r, no head] \arrow[ur, no head] & (0,1) \arrow[r, no head] \arrow[ur, no head] & (0,2) \arrow[r, no head] \arrow[ur, no head] & (0,3) \arrow[r, no head] \arrow[ur, no head] & (0,4) \arrow[r, no head, dotted] \arrow[ur, no head, dotted] & \cdots
\end{tikzcd}$$

Fig. 1 The Mostowski–Rabin index hierarchy

function rank : $Q \to \omega$, and $\delta : Q \times A \to Q$ 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 $r_{n+1} = \delta(r_n, u_n)$, for $n < \omega$. It is *accepting* if the highest rank occurring infinitely often (i.e., $\limsup_{n\to\infty} \operatorname{rank}(r_n)$) 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 $(\min \operatorname{rank}(Q), \max \operatorname{rank}(Q))$. It should be clear that moving all ranks up by an even number would not change the recognized language; therefore we may assume without loss of generality that $\min \operatorname{rank}(Q)$ is 0 or 1. It is useful to partially order the indices as represented on Fig. 1. That is, we let $(\iota, \kappa) \sqsubseteq (\iota', \kappa')$ if either $\{\iota, \dots, \kappa\} \subseteq \{\iota', \dots, \kappa'\}$, or $\iota = 0, \iota' = 1$, and $\{\iota + 2, \dots, \kappa + 2\} \subseteq \{\iota', \dots, \kappa'\}$. We consider the indices $(1, \kappa)$ and $(0, \kappa - 1)$ as *dual*, and let $\overline{(\iota, \kappa)}$ denote the index dual to (ι, κ).

To better understand the structure of indices, we can think of a structure $\langle \{\iota, \dots, \kappa\}, \leq, \mathit{Even} \rangle$ associated with an index (ι, κ), where *Even* stands for the evenness predicate. We identify indices whose structures are isomorphic. The ordering of indices corresponds to inclusion of isomorphism types, and the types of dual indices differ only by the monadic predicates, which are complement to each other.

The ordering of indices induces a hierarchy of languages, 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'). The hierarchy is known to be *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 $\overline{(\iota, \kappa)}$ [8, 18]. Indeed, the witness can be the parity condition itself:

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

2.2 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

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

Here $V_\exists$ and $V_\forall$ are (disjoint) sets of positions of Eve and Adam, respectively, $\mathrm{Move} \subseteq V \times V$ 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.

In a special case of $A = \{\iota, \dots, \kappa\}$, $L_\exists = L_{\iota,\kappa}$ (see Eq. (1)), and $L_\forall = \overline{L_\exists} = \{u \in \{\iota, \dots, \kappa\}^\omega : \limsup_{n\to\infty} u_n \text{ is odd}\}$, we refer to the game as to a *parity game of index* (ι, κ).

The players in a graph game 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 $v_0, v_1, v_2, \dots$, inducing the sequence of labels $\ell(v_0), \ell(v_1), \ell(v_2), \dots$. If this sequence belongs to $L_\exists$ then Eve wins the play, if it belongs to $L_\forall$ then Adam is the winner; otherwise there is a draw.

In the games considered in this paper, we always have $L_\forall = A^\omega - L_\exists$, hence a draw cannot occur.

Intuitively, a strategy is an oracle that tells the player her/his next move in the play, depending on the current history of the game. A *history* is any finite path in the arena starting from the initial position, i.e., a sequence $\pi = (v_0, v_1, v_2, \dots, v_k)$, with $v_0 = p_I$, and $(v_i, v_{i+1}) \in \text{Move}$, for $i < k$. We let $\text{last}(\pi) = v_k$.

We represent a *strategy for Eve* as a set S of histories, such that

1. p_I is the unique history of length 1 belonging to S,
2. each history $\pi \in S$ with $\text{last}(\pi) \in V_\exists$ has a unique prolongation πw_π in S (where $(\text{last}(\pi), w_\pi) \in \text{Move}$),
3. each history $\pi \in S$ with $\text{last}(\pi) \in V_\forall$ has a prolongation πw in S, for each w, such that $(\text{last}(\pi), w) \in \text{Move}$,
4. S is closed under initial segments.[2]

Note that the condition 2 induces a (partial) mapping $\pi \mapsto w_\pi$, which can be used as an alternative representation of the strategy.

A play (finite or infinite) is *consistent* with a strategy S if all its finite prefixes are in S. A strategy S is *winning* for Eve, if every play consistent with S is won by Eve. That is, either the play is finite and ends in a position from which Adam has no move, or it is infinite, and the sequence of labels belongs to $L_\exists$.

Analogically we define a strategy and a winning strategy for Adam. We say that Eve *wins* the game $\mathcal{G}$ if she has a winning strategy in this game, the similar for Adam.

In this paper we will often consider games whose arenas are infinite binary trees. A (full) binary tree over a finite alphabet *Alph* is a mapping $t : 2^* \to \mathit{Alph}$. For a finite alphabet A, we denote by $\text{GTr}(A)$ the set of binary trees over the alphabet $\mathit{Alph} = \{\exists, \forall\} \times A$. Any tree in $\text{GTr}(A)$ constitutes an arena, where the labels $\{\exists, \forall\}$ induce a partition of the nodes of the tree into positions of Eve and Adam, the moves go to the successors of a current node, and the initial position is the root.

A language $L \subseteq A^\omega$ can serve as a winning criterion for either of the players. We will therefore consider two classes of games: L-$\exists$ *games*, and L-$\forall$ *games*.

[2]Thus S contains no other histories than those mentioned in conditions 1–3.

An L-$\exists$ game over a tree $t \in \mathrm{GTr}(A)$ is given by the following items:

$$\begin{aligned}
&V_\exists = \{v \in 2^* : t(v)\downarrow_1 = \exists\} && \ell(v) = t(v)\downarrow_2, \quad \text{for } v \in 2^* \\
&V_\forall = \{v \in 2^* : t(v)\downarrow_1 = \forall\} && L_\exists = L \\
&\mathrm{Move} = \{(w, wi) : w \in 2^*, i \in 2\} && L_\forall = \overline{L} \\
&p_0 = \varepsilon \text{ (the root of the tree).}
\end{aligned} \tag{2}$$

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

We let

$$\mathrm{Win}^\exists(L) = \{t \in \mathrm{GTr}(A) : \text{Eve wins the } L\text{-}\exists \text{ game on } t\} \tag{3}$$

$$\mathrm{Win}^\forall(L) = \{t \in \mathrm{GTr}(A) : \text{Adam wins the } L\text{-}\forall \text{ game on } t\}. \tag{4}$$

2.3 Automata on Trees

For a finite alphabet A, let $\mathrm{Tr}(A)$ denote the set of all binary trees over A (please note the difference with the notation $\mathrm{GTr}(A)$ introduced in the previous section).

An *alternating parity automaton of index* (ι, κ) on infinite binary trees over an input alphabet A can be presented by

$$\mathcal{A} = \langle A, Q_\exists, Q_\forall, q_I, \delta, \mathrm{rank}\rangle \tag{5}$$

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

$$G(\mathcal{A}, t) = \langle Q_\exists \times 2^*, Q_\forall \times 2^*, \mathrm{Move}, (q_0, \varepsilon), \ell, A, L_{\iota,\kappa}, \overline{L_{\iota,\kappa}}\rangle, \tag{6}$$

where $\mathrm{Move} = \{((p, v), (q, vd)) : v \in \mathrm{dom}(t), \ (p, t(v), d, q) \in \delta\}$ and $\ell(q, v) = \mathrm{rank}(q)$. The language $L(\mathcal{A})$ recognized by $\mathcal{A}$ consists of those trees $t \in \mathrm{Tr}(A)$, for which Eve has a winning strategy in the game $G(\mathcal{A}, t)$.

Intuitively, players in this game 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 make sure that the sequence of ranks encountered during an infinite play satisfies the parity condition (and she wins all finite plays if any).

Similarly as for deterministic automata on infinite words discussed in Sect. 2.1, the hierarchy of tree languages induced by the Rabin-Mostowski indices of alternating parity automata is strict, as showed by Bradfield [5]. An alternative proof of this difficult result was later given [1] based on the Banach Fixed-Point Theorem. Both proofs [1, 5] use the same witness family of sets of binary trees, which can be expressed in terms of the winning sets defined in Eq. (3)

$$W_{\iota,\kappa} = \mathrm{Win}^\exists(L_{\iota,\kappa}). \tag{7}$$

In the present paper, we will consider sets $\mathrm{Win}^\exists(L)$ (and occasionally $\mathrm{Win}^\forall(L)$), for other sets L recognized by word automata. It is useful to note the following.

Lemma 1 *If a language L of infinite words is recognized by a deterministic automaton of index (ι,κ) then both languages* $\mathrm{Win}^{\exists}(L)$ *and* $\mathrm{Win}^{\forall}(L)$ *can be recognized by an alternating*[3] *tree automaton of index (ι,κ).*

Proof Let $\mathcal{B}$ be a deterministic automaton on infinite words of index (ι,κ), such that $L(\mathcal{B})=L$. The alternating tree automaton $\mathcal{A}$ accepting $\mathrm{Win}^{\exists}(L)$ will have three copies of the states of the automaton $\mathcal{B}$, say $Q^{\exists}$, $Q^{\forall}$, and Q^{ε}. The states in $Q^{\exists}$ are existential and the states in $Q^{\forall}$ are universal. We let the states in Q^{ε} be also universal (for concreteness), but they will not leave Adam any choice; these states will be used only in ε-moves. Let q^{∇} denote a respective copy of a state q. We let $\mathrm{rank}(q^{\nabla})=\mathrm{rank}(q)$.

The initial state is q_I^{ε}. For any state q and letter a, there are transitions $(q^{\varepsilon},(\exists,a),\varepsilon,q^{\exists})$ and $(q^{\varepsilon},(\forall,a),\varepsilon,q^{\forall})$. Whenever (p,a,q) is a transition in $\mathcal{B}$, there are moreover transitions $(p^{\exists},(\exists,a),0,q^{\varepsilon})$, $(p^{\exists},(\exists,a),1,q^{\varepsilon})$, and $(p^{\forall},(\forall,a),0,q^{\varepsilon})$, $(p^{\forall},(\forall,a),1,q^{\varepsilon})$. Now it can easily be verified that a winning strategy for Eve in the L-$\exists$-game on t can be transferred to a winning strategy of Eve in the automaton game $G(\mathcal{A},t)$, and *vice-versa*.

An analogous construction works for $\mathrm{Win}^{\forall}(L)$ (by interchanging the players). □

We now explain the Σ/Π terminology for the index hierarchy, which we have used informally in Introduction. It originates from the μ-calculus (see, e.g., [4]), and 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 any kind) of index (ι,κ) as to Σ_m-automaton or Π_m-automaton with an appropriate m.

This terminology extends to classes of languages of infinite words and trees. 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 in the classes Σ_m and Π_m.

For trees, we use the analogical terminology for alternating automata.

3 Tinkering with Infinite Words

The following property of languages will be useful. Informally, a language is stuttering if it does not distinguish between a single letter a and an nonempty block a^{ℓ}.

Definition 2 A language $L\subseteq A^{\omega}$ is *stuttering* if, for any $a\in A$, the following holds.

- For any infinite sequence of finite words $(w_n)_{n<\omega}$,

$$w_0aw_1aw_2a\cdots\in L \quad\Longleftrightarrow\quad w_0aaw_1aaw_2aa\cdots\in L.$$

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

- For any finite sequence of finite words w_n ($n = 0, \ldots, k$) and any infinite word v,

$$w_0 a w_1 a w_2 a \cdots a w_k v \in L \iff w_0 aa w_1 aa w_2 aa \cdots aa w_k v \in L.$$

Directly from the definition we get the following.

Fact 3 *If L is stuttering, then $\overline{L}$ is stuttering too.*

Fact 4 *Each language $L_{\iota,\kappa}$ is stuttering.*

In the sequel we consider words over a product alphabet E^2. We identify a pair of words $\langle (u_n), (v_n) \rangle \in (E^\omega)^2$ with a single word over $(E^2)^\omega$ via a mapping $\langle (u_n), (v_n) \rangle \mapsto \langle (u_n, v_n) \rangle$.

A deterministic automaton $\mathcal{A}$ of index (ι, κ) over an alphabet A induces a mapping $g_{\mathcal{A}} : A^\omega \to \{\iota, \ldots, \kappa\}^\omega$ defined as follows. For a word $u \in A^\omega$, let $r \in Q^\omega$ be the unique run of $\mathcal{A}$ on u (c.f. Sect. 2.1), and let

$$g_{\mathcal{A}}(u) = \mathrm{rank} \circ r.$$

That is, $g_{\mathcal{A}}(u)$ is the sequence of ranks encountered in the run of $\mathcal{A}$ on u. Note that $u \in L(\mathcal{A})$ iff $g_{\mathcal{A}}(u) \in L_{i,k}$, i.e., $L(\mathcal{A}) = g_{\mathcal{A}}^{-1}(L_{i,k})$.

If $\mathcal{B}$ is another deterministic automaton of index (ι, κ) over A, we define $g_{\mathcal{A}\times\mathcal{B}} : A^\omega \to (\{\iota, \ldots, \kappa\}^2)^\omega$ by $g_{\mathcal{A}\times\mathcal{B}}(u) = (g_{\mathcal{A}}(u), g_{\mathcal{B}}(u))$.

Lemma 5 *Let $\mathcal{A}$ and $\mathcal{B}$ be automata of index (ι, κ) over some alphabet A, such that $L(\mathcal{A}) \cap L(\mathcal{B}) = \emptyset$, and let $L = L_{\iota,\kappa}$.*

Then, for each word $u \in A^\omega$,

1. *if $u \in L(\mathcal{A})$ then $g_{\mathcal{A}\times\mathcal{B}}(u) \in L \times \overline{L}$,*
2. *if $u \in L(\mathcal{B})$ then $g_{\mathcal{A}\times\mathcal{B}}(u) \in \overline{L} \times L$,*
3. *$g_{\mathcal{A}\times\mathcal{B}}(u) \in \overline{L \times L}$.*

Proof As mentioned above $u \in L(\mathcal{A}) \Rightarrow g_{\mathcal{A}}(u) \in L$, and $u \notin L(\mathcal{B}) \Rightarrow g_{\mathcal{B}}(u) \in \overline{L}$. As $L(\mathcal{A})$ and $L(\mathcal{B})$ are disjoint, $u \in L(\mathcal{A})$ implies $u \notin L(\mathcal{B})$ hence $g_{\mathcal{A}\times\mathcal{B}}(u) \in L \times \overline{L}$.

The argument for 2 is similar. Finally, again by disjointness of $L(\mathcal{A})$ and $L(\mathcal{B})$, we have $g_{\mathcal{A}\times\mathcal{B}}(u) \in \overline{L \times L}$, for any u, which completes the proof. □

4 Separation Property of ω-Languages

We recall the main concept of the paper.

Definition 6 A set Z *separates* a pair of sets X, Y if $X \subseteq Z$ and $Y \cap Z = \emptyset$ or, symmetrically, $Y \subseteq Z$ and $X \cap Z = \emptyset$.

We say that the *separation property* holds for a class $\varGamma_m \in \{\varSigma_m, \varPi_m\}$ if any two disjoint languages[4] from this class are separable by a language in class Δ_m; otherwise the separation property fails for $\varGamma_m$.

In this section we show that the separation property holds for classes $\varPi_m$ and fails for classes $\varSigma_m$, for $m \geq 1$. In fact, both properties will follow from a single construction (parametrized by $m \geq 2$).

The level 1 is somewhat exceptional. The class $\varSigma_1$ consists of a single language $\emptyset$, and the class $\varPi_1$ consists of languages of all words (for each finite alphabet). Therefore the class Δ_1 is empty. For any alphabet A, there are no disjoint languages in the class $\varPi_1$, hence the separation property holds trivially. On the other hand, there is a pair of disjoint sets in the class $\varSigma_1$ (namely $\emptyset$ and $\emptyset$), which is inseparable because the class Δ_1 is empty. From now on, we will only consider classes with $m \geq 2$.

The following is the first non-separability result.

Lemma 7 *Let $L = L_{i,k} \subseteq \{i, \dots, k\}^\omega$ (with[5] $i \in \{0, 1\}$) and let $m = k + 1 - i$. Then $L \times \overline{L}$ and $\overline{L} \times L$ (which are obviously disjoint) are not separable by a set in Δ_m.*

Proof Suppose to the contrary that there is a set C recognizable by an automaton of index (i, k), such that $\overline{L} \times L \subseteq C$ and $L \times \overline{L} \subseteq \overline{C}$.

By Lemma 5,

- if $u \in C$ then $g_{\mathcal{A}\times\mathcal{B}}(u) \in L \times \overline{L} \subseteq \overline{C}$,
- if $u \in \overline{C}$ then $g_{\mathcal{A}\times\mathcal{B}}(u) \in \overline{L} \times L \subseteq C$.

Note that in this case the function $g_{\mathcal{A}\times\mathcal{B}}$ maps $(\{i, \dots, k\}^2)^\omega$ into itself. We will show that it has a fixed point. Indeed, a fixed point f can be defined[6] by an inductive formula

$$f_0 = \big(\mathrm{rank}\big(q_I^{\mathcal{A}}\big), \mathrm{rank}\big(q_I^{\mathcal{B}}\big)\big)$$
$$f_{n+1} = \big(\mathrm{rank}\big(\hat{\delta}^{\mathcal{A}}\big(q_I^{\mathcal{A}}, f_0 \cdots f_n\big)\big), \mathrm{rank}\big(\hat{\delta}^{\mathcal{B}}\big(q_I^{\mathcal{B}}, f_0 \cdots f_n\big)\big)\big),$$

where $\hat{\delta}$ is the standard extension of δ from letters to finite words.

For this fixed point f, we have $f \in C \Rightarrow f \in \overline{C}$ and $f \in \overline{C} \Rightarrow f \in C$, a contradiction which completes the proof. □

The subsequent lemma is the heart of our proof.

Lemma 8 *Let $E = \{i, \dots, k\}$, where k is even, and let $m = k + 1 - i$. Let $L = L_{i,k} \subseteq E^\omega$ (hence L is in $\varPi_m$), and let $I \subseteq L_{i,k}$ consist of the words that contain*

[4]Strictly speaking, we assume here that the languages $X, Y \in \varGamma_m$ are over the same alphabet; which matters in the case of $\varPi_1$ discussed below.

[5]We will later omit this assumption as the results hold also without it. Clearly the languages $L_{\iota,\kappa}$ and $L_{\iota+2\ell,\kappa+2\ell}$, for $\ell \geq 0$, have the same properties.

[6]The existence and uniqueness of this fixed point can be also inferred from the Banach Fixed-Point Theorem, c.f. Sect. 5.1.

Fig. 2 The table for the automata for U_1 and U_2

$$\begin{tikzcd}
(up,\cdot,\cdot) \arrow[d, bend right, "{(k,.)}"'] \\
(down,\cdot,\cdot) \arrow[u, bend right, "{(.,k)}"']
\end{tikzcd}$$

infinitely many occurrences of k. Then there exist disjoint sets $U_1, U_2 \subseteq (E^2)^\omega$ in the class Σ_m, satisfying the following.

$$\begin{aligned}
&L \times \overline{L} \subseteq U_1\\
&\overline{L} \times L \subseteq U_2\\
&\overline{L \times L} \subseteq U_1 \cup U_2 = \overline{I \times I}.
\end{aligned}$$

Proof The automata for U_1 and U_2 will only differ by their ranks. Their set of states is $\{up\} \times \{i, \dots, k\} \times \{i, \dots, k+1\} \cup \{down\} \times \{i, \dots, k+1\} \times \{i, \dots, k\}$, the initial state is $\langle up, i, i\rangle$, and the transitions are

$$\langle up, p, q\rangle \xrightarrow{(r,s)} \begin{cases} \langle up, r, s\rangle & \text{if } r < k \\ \langle down, k+1, s\rangle & \text{if } r = k \end{cases} \tag{8}$$

$$\langle down, p, q\rangle \xrightarrow{(r,s)} \begin{cases} \langle down, r, s\rangle & \text{if } s < k \\ \langle up, r, k+1\rangle & \text{if } s = k. \end{cases} \tag{9}$$

Figure 2 represents the table of the automata.

With each state s we assign two ranks: $\text{rank}_1(s)$ and $\text{rank}_2(s)$, for U_1 and U_2, respectively, by

$$\text{rank}_1\big(\langle up, p, q\rangle\big) = \begin{cases} p+2 & \text{if } p \le k \\ k+1 & \text{if } p = k+1 \end{cases}$$

$$\text{rank}_1\big(\langle down, p, q\rangle\big) = \begin{cases} q+1 & \text{if } q \le k \\ k+1 & \text{if } q = k+1 \end{cases}$$

$$\text{rank}_2\big(\langle up, p, q\rangle\big) = \begin{cases} p+1 & \text{if } p \le k \\ k+1 & \text{if } p = k+1 \end{cases}$$

$$\text{rank}_2\big(\langle down, p, q\rangle\big) = \begin{cases} q+2 & \text{if } q \le k \\ k+1 & \text{if } q = k+1 \end{cases}$$

Each word $u \in (E^2)^\omega$ 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 k on both left and right track if and only if it visits infinitely often a state $\langle up, r, k+1\rangle$ and a state $\langle down, k+1, s\rangle$. By the definition of the rank functions such a word is accepted by neither of the automata. On the other hand, if the run on u stabilizes on some level then only 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 $U_1 \cup U_2 = \overline{I \times I}$. The inclusion $\overline{L \times L} \subseteq \overline{I \times I}$ is obvious.

If $u \in L \times \overline{L}$ then the run on u stabilizes in the upper or lower component (as clearly $u \notin I \times I$). Then the automaton for U_1, from some moment on, either reads a word in L in the upper component or a word in $\overline{L}$ in the lower component; in either case it accepts. An argument for the inclusion involving U_2 is similar. □

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 [15], who also gave[7] a hint [16] on how this result can be generalized for all classes Π_n.

Theorem 9 *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 constructed in Lemma 8, for any class Σ_m, $m \geq 2$.

Let $\mathcal{A}$ and $\mathcal{B}$ be as in Lemma 5. It follows from 1 and 2 that the inverse image of U_1 under the mapping $g_{\mathcal{A}\times\mathcal{B}}$, i.e.,

$$g^{-1}_{\mathcal{A}\times\mathcal{B}}(U_1) = \{u \in A^\omega : g_{\mathcal{A}\times\mathcal{B}}(u) \in U_1\}$$

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 $g_{\mathcal{A}\times\mathcal{B}}$; the construction is straightforward. In a similar vein we can show that $(g_{\mathcal{A}\times\mathcal{B}})^{-1}(U_2)$ 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 5 that they sum up to A^ω, hence they both are of class Δ_m.

Now, by Lemma 8, $L \times \overline{L} \subseteq U_1$ and $\overline{L} \times L \subseteq U_2$, where L is the parity language in Π_m. If the sets U_1 and U_2 in Σ_m were separable by a set in Δ_m, then $L \times \overline{L}$ and $\overline{L} \times L$ would be separable, a contradiction with Lemma 7. □

5 Tinkering with Trees

5.1 Contractions

Definition 10 Let $2^{<n} = \{v \in 2^* : |v| < n\}$ and $2^{\leq n} = \{v \in 2^* : |v| \leq n\}$. We say that a mapping $g : \mathrm{Tr}(A) \to \mathrm{Tr}(B)$ (for some alphabets A and B) is *contracting* if, for all $n < \omega$, and $t, t' \in \mathrm{Tr}(A)$,

$$t \restriction 2^{<n} = t' \restriction 2^{<n} \quad \Longrightarrow \quad g(t) \restriction 2^{\leq n} = g(t') \restriction 2^{\leq n},$$

where $F \restriction X$ denotes the restriction of a function F to a set X.

[7] More precisely, that author considered the *reduction* property for the dual classes Σ_m. See a comment after Proposition 3.5 in [16].

In the sequel, we will often use[8] the celebrated Banach Fixed-Point Theorem: If g is a contracting mapping over the same domain $\mathrm{Tr}(A)$, it has a (unique) fixed point.

For $\iota \leq \kappa < \omega$, we will use an abbreviation $\mathrm{GTr}(\{\iota, \ldots, \kappa\}) = \mathrm{GTr}_{\iota,\kappa}$. The following fact, first noted in [1], is very useful in analysis of tree automata.

Theorem 11 [1] *Let $\mathcal{A}$ be an alternating tree automaton of index (ι, κ) over the alphabet A. There exists a contracting mapping $g : \mathrm{Tr}(A) \to \mathrm{GTr}_{\iota,\kappa}$ such that $t \in L(\mathcal{A})$ iff $g(t) \in W_{\iota,\kappa}$.*

Proof (sketch) In the first step we replace $\mathcal{A}$ with another automaton $\mathcal{A}'$ of the same index (ι, κ) such that $L(\mathcal{A}) = L(\mathcal{A}')$ and in every state of $\mathcal{A}'$ the player has exactly two possible moves. We fix a map h which assigns to a given tree t a tree $h(t)$ of all possible gameplays between Adam and Eve on t on $\mathcal{A}'$. That is, for every label of $h(t)$ the first coordinate indicates who the owner of the state is (this is $\exists$ or $\forall$) and the second coordinate is the rank of the state q. The sons of a given vertex are induced by transitions from the state q.

This is almost what we need, that is $t \in L(\mathcal{A})$ if and only if $h(t) \in W_{\iota,\kappa}$. In order to make h contracting, we add a dummy initial level to $h(t)$, that is $g(t)(\epsilon) = g(t)(\exists, \iota)$ and $g(t)(iv) = h(t)(v)$ for $i = 0, 1$. $\square$

The above theorem also implies its dual variant.

Corollary 12 *Let $\mathcal{A}$ be a tree automaton of index (ι, κ) over the alphabet A. There exists a contracting mapping $g : \mathrm{Tr}(A) \to \mathrm{GTr}_{\iota,\kappa}$ such that $t \in L(\mathcal{A})$ iff $g(t) \in W^{\forall}(L_{\iota,\kappa})$.*

Proof Since $W_{\iota,\kappa} = W^{\exists}(L_{\iota,\kappa})$, it is enough to compose a contraction from Theorem 11 with a mapping which switches roles between Eve and Adam. $\square$

5.2 Product of Trees

In this section we define the product of two trees in the sets GTr—a crucial technical concept needed later in Sect. 6. In a sense, it is a notion similar to the direct product of infinite words.

Consider two alphabets, A and B. Recall that the label of a vertex of a tree $t_0 \in \mathrm{GTr}(A)$ has two coordinates, the first one tells us who the owner of the state is, and the second coordinate is a letter from A; similarly, for a tree in $t_1 \in \mathrm{GTr}(B)$. The product operation on trees assigns to a pair t_0, t_1 a tree $t_0 \otimes t_1 \in \mathrm{GTr}(A \times B)$. A vertex of the tree $t_0 \otimes t_1$ is also labeled by two coordinates, but the second coordinate is a pair of two letters, one from A and one from B. The following inductive definition of the tree $t_0 \otimes t_1 \in \mathrm{GTr}(A \times B)$ defines precisely the labeling of the vertices (see Fig. 3).

[8] This makes the proofs simpler, although it would be possible to explicitly construct the fixed points like in the proof of Lemma 7.

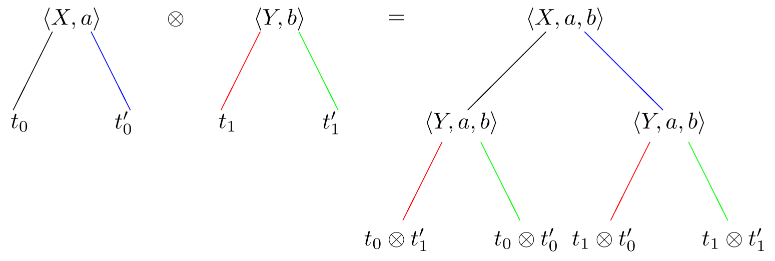

Fig. 3 Product of trees

We use the notation $\alpha(t, s)$ for a tree with the root labeled by α and the left and right subtrees t, s, respectively.

$$\begin{aligned}&\langle X, a\rangle(t_0, t_0) \otimes \langle Y, b\rangle\big(t_1, t'_1\big)\\ &\quad = \langle X, a, b\rangle\big(\langle Y, a, b\rangle\big(t_0 \otimes t_1, t_0 \otimes t'_1\big), \langle Y, a, b\rangle\big(t'_0 \otimes t_1, t'_0 \otimes t'_1\big)\big).\end{aligned}$$

As usual we can extend this operation to sets of trees; $U \otimes V = \{t \otimes s : t \in U, s \in V\}$.

Lemma 13 *If* $L_0 \subseteq A^\omega$ *and* $L_1 = B^\omega$ *are stuttering, then* $\mathrm{Win}^\exists(L_0) \otimes \mathrm{Win}^\exists(L_1) \subseteq \mathrm{Win}^\exists(L_0 \times L_1)$ *and* $\mathrm{Win}^\forall(L_0) \otimes \mathrm{Win}^\forall(L_1) \subseteq \mathrm{Win}^\forall(L_0 \times L_1)$.

Proof First let us observe that the sequence of labels in $A \times B$ along any infinite path $d_1 d'_1 d_2 d'_2 \cdots d_n d'_n \cdots$ in the product tree $t = t_0 \otimes t_1$ is $\langle a_0, b_0\rangle\langle a_0, b_0\rangle\langle a_1, b_1\rangle \cdots$, where path $d_1 d_2 \cdots d_n \cdots$ has labels $a_0 a_1 \cdots$ in t_0, and path $d'_1 d'_2 \cdots d'_n \cdots$ has labels $b_0 b_1 \cdots$ in t_1.

Now let σ_i be a strategy for Eve in t_i. In the case of games on trees, we can simplify the presentation of a strategy (c.f. Sect. 2.2) and view it just as a mapping from positions of Eve (i.e., nodes marked $\exists$) to $\{0, 1\}$, indicating a move to the left or right successor of the current node. We construct the strategy σ for Eve on t below. For any v such that $t(v) = \langle \exists, a, b\rangle$, let

$$\sigma(v) = \begin{cases} \sigma_0(\varepsilon) & \text{if } v = \varepsilon\\ \sigma_1(\varepsilon) & \text{if } v = d_1\\ \sigma_0(d_1 d_2 \cdots d_n) & \text{if } v = d_1 d'_1 \cdots d_n d'_n\\ \sigma_1(d'_1 d'_2 \cdots d'_n) & \text{if } v = d_1 d'_1 \cdots d_n d'_n d_{n+1}. \end{cases}$$

If $d_1 d'_1 d_2 d'_2 \cdots$ is any path played along the strategy σ on t then $d_1 d_2 \cdots$ is played along σ_0 in t_0 and $d'_1 d'_2 \cdots$ is played along σ_1 in t_1.

To prove that $\mathrm{Win}^\exists(L_0) \otimes \mathrm{Win}^\exists(L_1) \subseteq \mathrm{Win}^\exists(L_0 \times L_2)$, we have to show that if σ_0 and σ_1 are winning then so is σ. Let τ be any strategy for Adam and let $w = \langle a_0, b_0\rangle\langle a_1, b_1\rangle\langle a_2, b_2\rangle \cdots$ be the label of the path $d_1 d'_1 d_2 d'_2 \cdots$ played along σ and τ. Then $a_0 a_1 a_2 \cdots$ is the label of the path $d_1 d_2 \cdots$ played along the winning strategy σ_0, thus it belongs to L_0, and since L_0 is stuttering, $w_0 = a_0 a_0 a_1 a_1 \cdots \in L_0$

as well. Similarly $w_1 = b_0 b_0 b_1 b_1 \cdots \in L_1$. Thus $w = w_0 \times w_1 \in L_0 \times L_1$, which proves that σ is winning.

The proof for $\mathrm{Win}^{\forall}$ is very similar. $\square$

6 Inseparable Pairs in Σ_m

Recall that, for trees, we use the notation Σ_m, Π_m for classes of tree languages recognized by alternating automata of appropriate indices; a tree language is in the class Δ_m if it is simultaneously in the classes Σ_m and Π_m.

We let $m \geq 2$; the case of $m = 1$ will be considered in Sect. 6.1. Let U_1^m and U_2^m be the sets constructed in Lemma 8. By Lemma 1, the sets $\mathrm{Win}^{\exists}(U_1^m)$ and $\mathrm{Win}^{\forall}(U_2^m)$ are in the class Σ_m of the tree automata hierarchy. We will show that they are inseparable by a set in Δ_m. Let us first verify that these two sets are disjoint: if $t \in \mathrm{Win}^{\exists}(U_1^m) \cap \mathrm{Win}^{\forall}(U_2^m)$ then t has a branch labelled by a word in $U_1^m \cap U_2^m$, a contradiction.

Lemma 14 *Let (ι, κ) be the index corresponding to the class Π_m and let $L = L_{\iota,\kappa}$. If T and T' are two disjoint subsets of $\mathrm{GTr}_{\iota,\kappa}$ in class Π_m then there exists a contracting mapping h such that*

$$t \in T \Rightarrow h(t) \in \mathrm{Win}^{\exists}(L \times \overline{L})$$

and

$$t \in T' \Rightarrow h(t) \in \mathrm{Win}^{\forall}(\overline{L} \times L).$$

Proof Since T is in the class Π_m, there exists (by Theorem 11) a contracting mapping g such that $t \in T$ iff $g(t) \in \mathrm{Win}^{\exists}(L)$. From Corollary 12 we get a contracting mapping g' such that $t \in T'$ iff $g'(t) \in \mathrm{Win}^{\forall}(L)$.

Since T and T' are disjoint, $t \in T$ implies $t \notin T'$ and thus $g(t) \in \mathrm{Win}^{\exists}(L)$ and $g'(t) \in \mathrm{Win}^{\exists}(\overline{L})$. Hence by Lemma 13 we have $g(t) \otimes g'(t) \in \mathrm{Win}^{\exists}(L \times \overline{L})$.

Similarly, if $t \in T'$ then $g(t) \in \mathrm{Win}^{\forall}(\overline{L})$, $g'(t) \in \mathrm{Win}^{\forall}(L)$, and $g(t) \otimes g'(t) \in \mathrm{Win}^{\forall}(\overline{L} \times L)$. Now let us define h by $h(t) = g(t) \otimes g'(t)$. It is easy to check that h is contracting. $\square$

Lemma 15 *Let (ι, κ) be the index corresponding to the class Π_m and let $L = L_{\iota,\kappa}$. Then $\mathrm{Win}^{\exists}(L \times \overline{L})$ and $\mathrm{Win}^{\forall}(\overline{L} \times L)$ cannot be separated by a set $C \in \Delta_m$.*

Proof Suppose there is such a set $C \in \Delta_m$. Note that both sets $C, \overline{C}$ are in Π_m. Then, by the previous lemma with $T = \overline{C}$, $T' = C$, there exists a contracting h such that $t \in \overline{C} \Rightarrow h(t) \in \mathrm{Win}^{\exists}(L \times \overline{L}) \subseteq C$, $t \in C \Rightarrow h(t) \in \mathrm{Win}^{\forall}(\overline{L} \times L) \subseteq \overline{C}$. Since h is contracting, it has a fixed point $t = h(t)$, and for this t, $t \in \overline{C} \Rightarrow t \in C$, and $t \in C \Rightarrow t \in \overline{C}$, a contradiction. $\square$

Theorem 16 *The sets $\mathrm{Win}^{\exists}(U_1^m)$ and $\mathrm{Win}^{\forall}(U_2^m)$ are not separable by a set in Δ_m, for $m \geq 2$.*

Proof Let (ι, κ) be the index corresponding to the class Π_m. Since, by Lemma 8, $\mathrm{Win}^{\exists}(L_{\iota,\kappa} \times \overline{L_{\iota,\kappa}}) \subseteq \mathrm{Win}^{\exists}(U_1^m)$ and $\mathrm{Win}^{\forall}(\overline{L_{\iota,\kappa}} \times L_{\iota,\kappa}) \subseteq \mathrm{Win}^{\forall}(U_2^m)$, a separator of $\mathrm{Win}^{\exists}(U_1^m)$ and $\mathrm{Win}^{\forall}(U_2^m)$ would also separate $\mathrm{Win}^{\exists}(L_{\iota,\kappa} \times \overline{L_{\iota,\kappa}})$ and $\mathrm{Win}^{\forall}(\overline{L_{\iota,\kappa}} \times L_{\iota,\kappa})$, a contradiction with Lemma 15. □

6.1 Level 1

In contrast to deterministic automata for words, the classes Σ_1 and Π_1 for alternating automata on trees are not completely trivial. For example, an automaton $\mathcal{A}$ of index $(1,1)$ may accept a tree t if Eve wins the game $G(\mathcal{A}, t)$ in finite time. Similarly, an automaton of index $(0,0)$ may fail to accept a tree if Adam wins the game in finite time. Not surprisingly, the separation question for classes Σ_1 and Π_1 is not difficult to solve. We use, however, different methods than for higher levels, including some basic topology.

For a finite alphabet *Alph*, we equip the set $\mathrm{Tr}(Alph)$ with a topology *à la Cantor*, which can be defined by pointwise convergence: a set of trees t_n converges to a tree t if, for any vertex v, the sequence $t_n(v)$ stabilizes on some letter in *Alph*. It is well known that this space is compact and the family of closed-open sets of the form

$$U_f = \{s \in \mathrm{Tr}(Alph) : s(v) = f(v) \text{ for } v \in I\} \tag{10}$$

where $I \subseteq 2^*$ is finite and $f : I \to Alph$ is an arbitrary function, forms its basis. It is a folklore knowledge[9] that tree languages accepted by alternating automata of index $(0,0)$ (i.e., with trivial acceptance condition) are *closed* in this topology. Consequently, the languages accepted by automata of index $(1,1)$ are *open*, and the languages in the class Δ_1 *clopen* (closed-open).

Proposition 17 *The separation property fails for the class Σ_1 and holds for the class Π_1.*

Proof For the negative result, we construct automata A_1 and A_2 as follows. Both automata read the leftmost branch of a given tree over the alphabet $\{a, b\}$. A_1 accepts a tree t if and only if the first appearance of the letter a is at an even level, and A_2 accepts t iff the first appearance of the letter a is at an odd level. Formally speaking, the automaton A_1 has two existential states q_0, q_1 of rank 1 and one "dead" universal state (with no move) also of rank 1. The initial state is q_0. In the game $G(\mathcal{A}, t)$, Eve alternates between her two states q_0, q_1 moving to the left when she reads letters b unless she is in the state q_0 and reads letter a. In this case she moves to the "dead" state of Adam, and thus wins the game. The same description works for A_2 except that the roles of q_0 and q_1 are switched.

Clearly the languages $L(A_1)$ and $L(A_2)$ are disjoint and of index $(1,1)$. Suppose $L(A_i)$ is contained in a certain language K_i of index $(0,0)$ for $i = 1, 2$. We claim that a tree t with every vertex labeled b belongs to $K_1 \cap K_2$. For, let t_n be a tree with every

[9]We may refer the reader to [6], where an analogous fact is shown for all levels of the weak hierarchy, c.f. Sect. 7.

vertex labelled b except of the vertex 0^n (i.e., the n-th vertex on the leftmost branch), which is labeled a. Clearly, t is a pointwise limit of trees $t_{2n} \in L(A_1)$ $(n < \omega)$ and it is also a pointwise limit of trees $t_{2n+1} \in L(A_2)$ $(n < \omega)$. But, as we have noted above, the sets K_i are closed, and hence $t \in K_1 \cap K_2$. This shows that it is impossible to separate $L(A_1)$ and $L(A_2)$ by a set in Δ_1.

Now let $L_1, L_2 \subseteq \mathrm{Tr}(Alph)$ be two disjoint languages recognizable by automata of index $(0, 0)$. We will show that they can be separated by a language C in class Δ_1.

For each tree $t \in L_1$ we fix a finite set of vertices $I_t \subset 2^*$ and a set

$$U_t = \{s \in \mathrm{Tr}(Alph) : s(v) = t(v) \text{ for } v \in I_t\},$$

such that $U_t \cap L_2 = \emptyset$. One can find such set I_t, since the set L_2 is closed and for a fixed $t \notin L_2$ the family

$$\{\{s \in \mathrm{Tr} : s(v) = t(v) \text{ for } v \in I\} : I \subseteq 2^*,\ I \text{ finite}\}$$

is a basis of open sets in the point t (c.f. Eq. (10)).

Since the space is compact and sets U_t are open, one can find $t_0, \dots, t_k$, such that $L_1 \subseteq C = U_{t_0} \cup \cdots \cup U_{t_k}$; by definition C separates L_1 and L_2. We claim that C is in the class Δ_1; clearly it is enough to show that this holds for each set of the form (10) (each $U_{t,i}$ is of such form).

The $(0, 0)$ and $(1, 1)$ automata recognizing U_f are identical except ranks. We describe them in terms of the respective games. Adam first makes an ε-move: he chooses a vertex in the finite set I, say v. Once the decision is made, the automaton moves from the root to v in a deterministic manner and then verifies whether the label $t(v)$ agrees with $f(v)$. If it is the case, the automaton enters a "dead" state of Adam, and Adam looses. Otherwise, it enters a "dead" state of Eve, and Eve looses. Note that any play in this game is finite (bounded by $\max\{|v| : v \in I\} + 2$), hence the ranks of the states do not matter.

This remark completes the proof. □

Remark From the above considerations, it is not hard to infer a simple characterizations of languages in the class Δ_1. Namely, the following conditions are equivalent for a tree language $L \subseteq \mathrm{Tr}(Alph)$.

- L is simultaneously recognized by an automaton of index $(0, 0)$ and by an automaton of index $(1, 1)$,
- L is closed and open in the above topology,
- L is accepted by an alternating automaton B (of any index), such that every gameplay between Adam and Eve in B is finite,
- L is accepted by an alternating automaton B, and there is a constant N such that every gameplay between Adam and Eve in B ends in less than N moves.

We now summarize the results of this section, Theorem 16 above, and the results on level 2 mentioned in Introduction [7, 13].

Corollary 18 *Consider the Rabin-Mostowski index hierarchy for alternating automata on trees. Separation property fails for all classes Σ_m, $m \geq 1$, and holds for the classes Π_1 and Π_2.*

The status of the classes Π_m, $m \geq 3$, remains open.

7 Weakly Recognizable Tree Languages

A *weak alternating parity tree automaton of index* (ι, κ) can be presented as an ordinary automaton introduced in Sect. 2.3, subject to an extra monotonicity condition: if (p, a, d, q) is a transition in δ then $\mathrm{rank}(q) \geq \mathrm{rank}(p)$. The Rabin-Mostowski indices of weak automata induce a hierarchy of tree languages similarly as for the ordinary automata; however we will not associate letters Σ/Π with this hierarchy. The strictness of the weak hierarchy has been established prior to the "strong" one by A.W.Mostowski [10]; an alternative proof can be given *via* the weak versions of languages $W_{\iota,\kappa}$ (see, e.g., [4]).

Note that the weak automata with indices $(0, 0)$ and $(1, 1)$ coincide with the ordinary ones, and we have already considered the respective classes in Sect. 6.1.

From now on we fix a pair (ι, κ) $(\iota < \kappa < \omega)$ and define $M = M_{\iota,\kappa}$ to be the set of all non–decreasing sequences in $\{\iota, \ldots, \kappa\}^{\omega}$. We let $\mathbb{A}M$ be the set of all trees in $\mathcal{T}_{\iota,\kappa}$, such that the labeling of any path projected at the second component is in M. Let $L = L_{\iota,\kappa}$. The mapping g constructed in Theorem 11 has the following property in the weak case (see, e.g., [4] Theorem 8.3.1).

Fact 19 *Let* $T \subseteq \mathrm{GTr}_{\iota,\kappa}$ *be a weak language of index* (ι, κ)*. Then there exists a contracting mapping* g *such that* $t \in \mathcal{T} \Leftrightarrow g(t) \in \mathrm{Win}^{\exists}(L)$ *and moreover, for all* t*,* $g(t) \in \mathbb{A}M$*. The analogous claim holds for* $\mathrm{Win}^{\forall}(L)$*.*

Now, as in the general case (c.f. Lemma 14), if $T, T' \subseteq \mathrm{GTr}_{\iota,\kappa}$ are two weak languages of index (ι, κ) then there exist g and g' such that

- if $t \in T$ then $g(t) \in \mathrm{Win}^{\exists}(L) \cap \mathbb{A}M$ and $g'(t) \in \mathrm{Win}^{\exists}(\overline{L}) \cap \mathbb{A}M$; it follows that $g(t) \otimes g'(t) \in \mathrm{Win}^{\exists}(L \times \overline{L}) \cap \mathbb{A}M^2 \subseteq \mathrm{Win}^{\exists}((L \times \overline{L}) \cap M^2)$;
- analogously, if $t' \in T'$ then $g(t') \otimes g'(t') \in \mathrm{Win}^{\forall}((\overline{L} \times L) \cap M^2)$.

Since $g \otimes g'$ is contracting, we get the following, for the same reasons as in Lemma 15.

Fact 20 $\mathrm{Win}^{\exists}((L \times \overline{L}) \cap M^2)$ *and* $\mathrm{Win}^{\forall}((\overline{L} \times L) \cap M^2)$ *cannot be separated by a set* C *such that* C *and* $\overline{C}$ *are weak languages of index* (ι, κ)*.*

From now on we assume that $\iota = 1$, hence $L = L_{1,\kappa}$ and $M = M_{1,\kappa}$, with $1 < \kappa < \omega$. We will prove an analogue of Lemma 8. We call a deterministic automaton on infinite words *weak* if its transitions do not decrease ranks.

Lemma 21 *There exist two disjoint sets* $V_1, V_2 \subseteq (\{1, \ldots, \kappa\}^2)^{\omega}$*, satisfying the following.*

$$(L \times \overline{L}) \cap M^2 \subseteq V_1$$
$$(\overline{L} \times L) \cap M^2 \subseteq V_2.$$

The languages V_1, V_2 *are accepted by weak deterministic automata of index* $(1, \kappa)$*.*

Proof Let $\top = \{1, \dots, \kappa\}^\omega$. For any language $Z \subseteq \top$ and $i \in \{1, \dots, \kappa\}$, we abbreviate $Z_i = Z \cap \{i, \dots, \kappa\}^\omega$.

Let us first consider the case of κ odd.

Let $X = L \cap M$ and $Y = \overline{L} \cap M$.

We let

$$V_1 = (1,1)^* \Bigl(\bigcup_{1<q} (1,q)(\top \times Y_q) \cup \bigcup_{\substack{1<p \\ 1\leq q}} (p,q)(X_p \times \top) \Bigr) \tag{11}$$

$$V_2 = (1,1)^* \Bigl(\bigcup_{1<q} (1,q)(\top \times X_q) \cup \bigcup_{\substack{1<p \\ 1\leq q}} (p,q)(Y_p \times \top) \Bigr) \tag{12}$$

Note that $X_p \cap Y_p = \emptyset$, for any p, which follows that V_1 and V_2 are disjoint. It is also straightforward to verify the inclusions claimed in the lemma.

We now describe a weak deterministic automaton of index $(1, \kappa)$ recognizing V_1. We use notation $p \overset{s}{\to} q$ to mean $\delta(p, s) = q$. The automaton has states $1, 2, \dots, \kappa$, $\overline{2}, \dots, \overline{\kappa}$, and ♠. The rank of state i is i, the rank of state $\overline{i}$ is $i - 1$, and the rank of ♠ is κ. The initial state 1 has transitions

$$1 \overset{1,1}{\to} 1 \qquad 1 \overset{1,q}{\to} \overline{q} \quad \text{for } 1 < q \qquad 1 \overset{p,q}{\to} p \quad \text{for } 1 < p, 1 \leq q.$$

The states $2 \leq i$ "read" the first component as long as monotonicity is preserved, i.e., the transitions are $i \overset{j,k}{\to} j$, for $j \geq i$. However, violating monotonicity is "punished" by entering ♠, i.e., $i \overset{j,k}{\to}$ ♠, whenever $j < i$. Similarly, the states $\overline{i}$ read the second component, i.e., $\overline{i} \overset{j,k}{\to} \overline{j}$, for $j \geq i$; but violating monotonicity is again punished by ♠. The state ♠ loops in itself independently of the input letter.

By definition, the automaton is weak of index $(1, \kappa)$; We leave it to the reader to check that it recognizes V_1. The construction of the automaton for V_2 is analogous.

For κ even, we again consider two cases. Let us first consider $\kappa \geq 4$. The sets V_1 and V_2 are defined by the formulas (11) and (12) above, but we change the definition of X, which is now

$$X = (L \cap M) \cup \overline{M}.$$

Note that $X_p \cap Y_p = \emptyset$ continues to hold, and hence the sets V_1 and V_2 are disjoint. They clearly satisfy the inclusions claimed in the lemma.

An automaton recognizing V_1 is similar as in the previous case, except for the rank of ♠ which is now $\kappa - 1$ (which is odd and greater than 1). Moreover, the automaton has a new self-looping state ♡ with the rank κ. The transitions are similar as before whenever monotonicity is preserved. The violation of monotonicity in state $\overline{i}$ is again punished by entering the state ♠. In contrast, the violation of monotonicity in a state i is not punished at all, but instead it is granted by entering a "winning" state ♡. Again, it is straightforward to verify that the automaton satisfies all requirements. An automaton for V_2 is analogous.

Finally for the index $(1,2)$, we let

$$V_1 = (1,1)^*(2,1)(\top,\top)$$
$$V_2 = (1,1)^*(1,2)(\top,\top).$$

It is immediate to see that the sets V_1 and V_2 are disjoint and the inclusions of the lemma are satisfied. An automaton recognizing V_1 has an initial state 1 and two self-looping states ♠ and ♡ ranked as before. The transitions for 1 are

$$1 \xrightarrow{1,1} 1 \qquad 1 \xrightarrow{2,1} \heartsuit \qquad 1 \xrightarrow{i,2} \spadesuit \quad \text{for } i = 1,2.$$

An automaton for V_2 is analogous. □

Now, by reading the proof of Lemma 1, we can easily see if a deterministic automaton recognizing a set of infinite words is *weak* of index (ι,κ) then the alternating automata recognizing sets $\mathrm{Win}^{\exists}(L)$ and $\mathrm{Win}^{\forall}(L)$ are also weak of index (ι,κ). Hence we can note the following.

Fact 22 *For languages V_1, V_2 constructed in Lemma 21, the sets of trees* $\mathrm{Win}^{\exists}(V_1)$ *and* $\mathrm{Win}^{\forall}(V_2)$ *can be recognized by weak alternating automata of index* $(1,\kappa)$.

Thus we obtain the following analogue of

Theorem 23 *For every $\kappa \geq 1$, the separation property fails for the class of weak regular languages of index $(1,\kappa)$.*

Proof For index $(1,1)$, the weak automata coincide with the ordinary ones, and this case has been already settled in Proposition 17.

For $\kappa > 1$, we claim that the sets $\mathrm{Win}^{\exists}(V_1)$ and $\mathrm{Win}^{\forall}(V_2)$ are inseparable. Indeed, as $\mathrm{Win}^{\exists}((L \times \overline{L}) \cap M^2) \subseteq \mathrm{Win}^{\exists}(V_1)$ and $\mathrm{Win}^{\forall}((\overline{L} \times L) \cap M^2) \subseteq \mathrm{Win}^{\forall}(V_2)$ (by Lemma 21), a separator of $\mathrm{Win}^{\exists}(V_1)$ and $\mathrm{Win}^{\forall}(V_2)$ would also separate $\mathrm{Win}^{\exists}((L \times \overline{L}) \cap M^2)$ and $\mathrm{Win}^{\forall}((\overline{L} \times L) \cap M^2)$, contradicting Fact 20. □

8 Reduction Property

Sets C, D *reduce* a given pair of sets A, B if $C \subseteq A$, $D \subseteq B$, $C \cup D = A \cup B$ and $C \cap D = \emptyset$. The *reduction property* holds for a class $\mathcal{C}$ of sets, if for every $A, B \in \mathcal{C}$, there exist $C, D \in \mathcal{C}$ which reduce A, B. The reduction property for a class $\mathcal{C}$ implies readily the separation property for the dual class $\{\overline{C} : C \in \mathcal{C}\}$. In set-theoretic hierarchies, usually the converse also holds, that is the separation property for a class $\mathcal{C}$ implies the reduction property for the dual class (see [9]). We will see that it is not the case for the index hierarchy of the alternating automata on trees.

Theorem 24 *The reduction property fails for all classes Σ_m and Π_m, for $m \geq 1$.*

The rest of this section is devoted to the proof of this theorem. We consider a class Σ_m with $m \geq 2$; the proof for Π_m is analogous.

Let us start with a simple combinatorial lemma.

Lemma 25 *Let $\top$ be any set and $W \subseteq \top$. Assume that*

$$(\top \times W) \cup (W \times \top) = X \cup Y,$$

where $X \subseteq \top \times W$, $Y \subseteq W \times \top$, and the sets X and Y are disjoint. Suppose further that

$$X = \bigcup_{i=1}^{m} a_i \times b_i, \qquad Y = \bigcup_{i=1}^{n} c_i \times d_i$$

for some sets $a_i, b_i, c_i, d_i \subseteq \top$. Then the set $\overline{W} = \top - W$ can be generated from the sets $a_1, \dots, a_m, d_1, \dots, d_n$, by (finite) union and intersection.[10]

Proof Let, for $t \in \top$,

$$A_t = \bigcap_{t \in a_i} a_i,$$

where $A_t = \top$ if the set $\{a_i : t \in a_i\}$ is empty. If

$$\overline{W}c = \bigcup_{t \in \overline{W}} A_t \tag{13}$$

then we are done, as there are only finitely many distinct sets A_t. Note that the inclusion $\subseteq$ in (13) always holds as $t \in A_t$, for any t. Thus (13) may fail only if there are some $t \in \overline{W}$, $t' \in W$, such that $t' \in A_t$. We will show that, in this case,

$$\overline{W} = \{s : (t', s) \in Y\}. \tag{14}$$

($\supseteq$) If $s \in \overline{W}$ then $(t', s) \in X \cup Y$ (because $t' \in W$) but $(t', s) \notin X$ (because $s \notin W$), hence $(t', s) \in Y$.

($\subseteq$) Let $(t', s) \in Y$. Suppose $s \in W$; then $(t, s) \in X$. (For, it is in $X \cup Y$ because of s, but not in Y because of t.) Now the assumption that $t' \in A_t$ implies that $(t', s) \in X$ as well. (For, if $(t, s) \in a_i \times b_i$ then $(t', s) \in a_i \times b_i$ as well.) But this is impossible as X and Y are disjoint; the contradiction proves that $s \in \overline{W}$.

To complete the proof, note that the right-hand side of (14) amounts to

$$\{s : (t', s) \in Y\} = \bigcup_{t' \in c_i} d_i.$$

□

The next lemma will introduce a decomposition of a recognizable tree language into "rectangles" in the same class. We fix an alphabet A and a letter $d \in A$. Recall

[10] We allow empty unions/intersections, i.e., $\emptyset$ and $\top$.

that $d(t_0, t_1)$ denotes a tree with the root labeled by d, and the left and right subtrees t_0 and t_1, respectively. For sets of trees K, L, let $d(K, L) = \{d(t_0, t_1) : t_0 \in K, t_1 \in L\}$. Clearly, if K and L are in the class Σ_m, so is $d(K, L)$.

Let $\top = \mathrm{Tr}(A)$ stand for all trees over alphabet A.

Lemma 26 *Let $T \subseteq d(\top, \top)$ be in the class Σ_m. Then*

$$T = \bigcup_{i \in I} d(A_i, B_i),$$

for some finite set of pairs $(A_i, B_i)_{i \in I}$ of sets in Σ_m.

Proof Let $\mathcal{A}$ be an alternating tree automaton of index (ι, κ) recognizing T (like in (5)). For a state $q \in Q$, let $\mathcal{A}_q$ be an automaton which differs from $\mathcal{A}$ only in that its initial state is q.

Let $t = d(t_0, t_1) \in T$, and let S be a winning strategy of Eve in the game $G(\mathcal{A}, t)$ (c.f. Sect. 2.3). Recall that nodes in the arena of this game have form (q, v), where q is a state of the automaton and v is a node of the tree; in particular the initial position of the game is (q_I, ε). Let $L(t)$ be the set of those histories in S, where the left successor of the root of the tree t (i.e., the node 0) is reached for the first time (possibly after a sequence of ε-moves). That is, π is in $L(t)$ if it is in the form $(v_0, v_1, \ldots, v_k)$, where $v_0 = (q_I, \varepsilon)$, $v_k = (q, 0)$, for some q, and $v_i = (p_i, \varepsilon)$, for $i < k$, for some states p_i. Let $R(t)$ be defined similarly for the right subtree.

Let $L_{state}(t) = \{q : \mathrm{last}(\pi) = (q, 0), \text{for some } \pi \in L(t)\}$, and let $R_{state}(t)$ be defined similarly. Note that $t_0 \in L(\mathcal{A}_q)$, whenever $q \in L_{state}(t)$, as Eve can use a substrategy of S (starting from an appropriate history) to win the game $G(\mathcal{A}_q, t_0)$. The similar holds for t_1 and $R_{state}(t)$. We claim that a kind of converse also holds: if $t_0' \in \bigcap_{q \in L_{state}(t)} L(\mathcal{A}_q)$ and $t_1' \in \bigcap_{q \in L_{state}(t)} R(\mathcal{A}_q)$ then $d(t_0', t_1') \in T$. Indeed, for each $q \in L_{state}(t)$, Eve has a winning strategy $S_{q,0}$ in the game $G(\mathcal{A}_q, t_0')$, and the similar holds for t_1'. Hence, Eve can combine these strategies with the strategy S. That is, she plays S until the play reaches a history $\pi \in L(t)$ with $\mathrm{last}(\pi) = (q, 0)$, for some q (or $\pi \in R(t)$ with $\mathrm{last}(\pi) = (q, 1)$), in which case she switches to the strategy $S_{q,0}$ (or $S_{q,1}$). Symbolic presentation of this new strategy in terms of concatenation of sets of histories is easy but cumbersome, and we omit it here.

Hence

$$T = \bigcup_{t \in T} d\left(\bigcap_{q \in L_{state}(t)} L(\mathcal{A}_q), \bigcap_{p \in R_{state}(t)} L(\mathcal{A}_p) \right).$$

Obviously, the number of pairs of sets which may occur as arguments of d in the formula above is finite. It is well-known (and easy to verify) that the class Σ_m is closed under finite meets. Hence the formula gives us a decomposition required in the lemma. □

We now fix an alphabet A and a tree language W over A belonging to the class Σ_m, but not in Π_m. For concreteness, for $\iota < \kappa$, we can take the set $W_{\iota,\kappa}$ for appropriate ι, κ, c.f. Sect. 2.3. For the classes Σ_1 and Π_1 (c.f. Sect. 6.1), we leave the construction of the respective examples to the reader. We claim that the pair $d(\top, W), d(W, \top)$ is not reducible.

For, suppose that

$$d(\top, W) \cup d(W, \top) = X \cup Y,$$

for some sets X, Y in class Σ_m, such that $X \subseteq d(\top, W)$, $Y \subseteq d(W, \top)$, and the sets X and Y are disjoint. Let

$$X = \bigcup_{i=1}^{m} d(a_i, b_i) \qquad Y = \bigcup_{i=1}^{n} d(c_i, d_i)$$

be decompositions given by Lemma 26; in particular all sets a_i, b_i, c_j, d_j are in class Σ_m. We apply Lemma 25 *via* an obvious correspondence between a tree $d(t_0, t_1)$ and a pair (t_0, t_1); hence $\overline{W}$ can be presented as a $\cup\cap$–combination of sets in Σ_m. By the closure properties of Σ_m, this would imply that $\overline{W}$ is in this class itself, contradicting the choice of W. This remark completes the proof of Theorem 24.

Acknowledgements D. Niwiński wishes to thank Victor Selivanov and Wolfgang Thomas for insightful comments after his talk at the Dagstuhl Seminar 11411 (October 2011), in particular for an encouragement to study the problem for weak automata.

References

1. Arnold, A.: The μ-calculus alternation-depth hierarchy is strict on binary trees. RAIRO Theor. Inform. Appl. **33**, 329–339 (1999)
2. Arnold, A., Michalewski, H., Niwiński, D.: On the separation question for tree languages. In: STACS, pp. 396–407 (2012)
3. Arnold, A., Niwiński, D.: Fixed point characterization of weak monadic logic definable sets of trees. In: Nivat, M., Podelski, A. (eds.) Tree Automata and Languages, pp. 159–188. Elsevier, Amsterdam (1992)
4. Arnold, A., Niwiński, D.: Rudiments of μ-Calculus. Elsevier Science, Studies in Logic and the Foundations of Mathematics, vol. 146. North-Holland, Amsterdam (2001)
5. Bradfield, J.C.: Simplifying the modal mu-calculus alternation hierarchy. In: Proc. STACS'98. Lect. Notes Comput. Sci., vol. 1373, pp. 39–49 (1998)
6. Duparc, J., Murlak, F.: On the topological complexity of weakly recognizable tree languages. In: FCT. LNCS, vol. 4639, pp. 261–273 (2007)
7. Hummel, S., Michalewski, H., Niwiński, D.: On the Borel inseparability of game tree languages. In: STACS, pp. 565–575 (2009)
8. Kaminski, M.: A classification of omega-regular languages. Theor. Comput. Sci. **36**, 217–229 (1985)
9. Kechris, A.S.: Classical Descriptive Set Theory. Springer, New York (1995)
10. Mostowski, A.W.: Hierarchies of weak automata and weak monadic formulas. Theor. Comput. Sci. **83**, 323–335 (1991)
11. Moschovakis, Y.N.: Descriptive Set Theory. North-Holland, Amsterdam (1980)
12. Muller, D.E., Saoudi, A., Schupp, P.E.: Alternating automata, the weak monadic theory of trees and its complexity. Theor. Comput. Sci. **97**(2), 233–244 (1992)
13. Rabin, M.O.: Weakly definable relations and special automata. In: Bar-Hillel, Y. (ed.) Mathematical Logic and Foundations of Set Theory, pp. 1–23 (1970)
14. Santocanale, L., Arnold, A.: Ambiguous classes in μ-calculi hierarchies. Theor. Comput. Sci. **333**, 265–296 (2005)

15. Selivanov, V.: Fine hierarchy of regular ω-languages. Theor. Comput. Sci. **191**, 37–59 (1998)
16. Selivanov, V.: Fine hierarchy and m-reducibilities theoretical computer science. Theor. Comput. Sci. **405**, 116–163 (2008)
17. Thomas, W.: Languages, automata, and logic. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, pp. 389–455. Springer, Berlin (1997)
18. Wagner, K.: On ω-regular sets. Inf. Control **43**, 123–177 (1979)