The Topological Complexity of MSO+U and Related Automata Models

Szczepan Hummel*

Michał Skrzypczak†

Institute of Informatics
University of Warsaw
Poland
{shummel, mskrzypczak}@mimuw.edu.pl

April 26, 2012

Abstract

This work shows that for each $i \in \omega$ there exists a $\mathbf{\Sigma}^1_i$-hard ω-word language definable in Monadic Second Order Logic extended with the unbounding quantifier (MSO + U). This quantifier was introduced by Bojańczyk to express some asymptotic properties. Since it is not hard to see that each language expressible in MSO + U is projective, our finding solves the topological complexity of MSO + U. The result can immediately be transferred from ω-words to infinite labelled trees.

As a consequence of the topological hardness we note that no alternating automaton with a Borel acceptance condition — or even with an acceptance condition of a bounded projective complexity — can capture all of MSO + U. The same holds for deterministic and nondeterministic automata since they are special cases of alternating ones.

We also give exact topological complexities of related classes of languages recognized by nondeterministic ωB-, ωS- and ωBS-automata studied by Bojańczyk and Colcombet. Furthermore, we show that corresponding alternating automata have higher topological complexity than nondeterministic ones — they inhabit all finite levels of the Borel hierarchy.

The paper is an extended journal version of [HST10]. The main theorem of that article is strengthened here.

*This paper has been partially supported by the Polish Ministry of Science grant no. N206 567840

†Author supported by ERC Starting Grant "Sosna" no. 239850

Introduction

Since the seminal paper of Büchi [Büc62] the class of ω-word languages definable in Monadic Second Order logic is of a great interest of computer scientists. The crucial result states that the emptiness problem is decidable in that class. Various properties of potentially infinite computations (like liveness and safety) can be expressed in MSO and therefore automatically verified. By its closure properties and computational tractability, the class of MSO definable ω-word languages is traditionally referred to as ω-regular languages.

Due to [Büc62], [McN66] and [Saf88] it is known that every ω-regular language is recognised by some deterministic Muller automaton. This entails that on ω-words the expressive power of MSO is equal to the one of Weak Monadic Second Order logic. The latter is a variant of MSO where set quantification is restricted to finite subsets of the domain.

Mikołaj Bojańczyk has proposed an extension of the class of ω-regular languages which is able to express some asymptotic properties of words. The extension was first introduced to tree languages in [Boj04], and then mainly studied for ω-words (see e.g. [BC06], [Boj11] and [Boj10]). The idea was to consider an additional set quantifier U, called the *unbounding quantifier*, which is defined so that the formula $\mathsf{U}X.\varphi(X)$ is equivalent to writing:

"$\varphi(X)$ is satisfied by arbitrarily large finite sets X of positions"

The canonical examples of the languages that can be described using this quantifier are:

$$L_B = \{a^{n_1}ba^{n_2}ba^{n_3}b\ldots \mid \limsup n_i < \infty\} \qquad L_S = \{a^{n_1}ba^{n_2}ba^{n_3}b\ldots \mid \liminf n_i = \infty\}$$

The most important result of [Boj11] states that the theory of WMSO extended with U (WMSO + U) is decidable over ω-words. The proof leads through the construction of equivalent model of deterministic automata — so called max-automata. It turns out that the emptiness problem for max-automata is decidable.

Automata Models

The problem whether full MSO + U is decidable remains open. The difference versus WMSO + U is that MSO + U allows quantifiers ranging over arbitrary subsets of the domain. Existential quantification corresponds to a projection of an alphabet or to the nondeterminism on the automata side.

In [BC06], ωBS-automata were defined as nondeterministic automata equipped with counters which can be incremented or reset, but not read. The acceptance condition may require a counter to be bounded (the B-condition) or convergent to ∞ (the S-condition). Thanks to the nondeterminism, ωBS-automata are capable of capturing full existential quantification, therefore, they are more expressive than max-automata. Unfortunately the class defined by these automata is not closed under the complementation. This is why the authors consider two restrictions of the class: ωB-automata using only the B-condition and ωS-automata using only the S-condition. The main technical result of [BC06] shows that the complement of a language defined by an ωB-automaton is accepted by an ωS-automaton, and vice versa.

Since none of the above models is closed under both boolean operations and projections, one might want to consider alternating ωBS-automata. Such automata are an extension of nondeterministic ones and they are closed under boolean operations. However, the decidability of the emptiness problem for them is still open.

Monadic Second Order Logic with U

In [HST10] the authors have given an example of a $\mathbf{\Sigma}_1^1$-complete language definable in $\mathrm{MSO} + \mathrm{U}$. This result has already excluded all nondeterministic automata with Borel acceptance conditions as a potential automata models for $\mathrm{MSO} + \mathrm{U}$. After that result there was still hope that alternating ωBS-automata are the desired model.

This paper extends the result of [HST10]. We give exact estimations of the topological complexity of $\mathrm{MSO} + \mathrm{U}$. It turns out to be as high as possible — there are $\mathrm{MSO} + \mathrm{U}$ definable languages arbitrarily high in the projective hierarchy.

To apply the above result to potential automata models for $\mathrm{MSO} + \mathrm{U}$ we recall the fact that the topological complexity of the language $L(\mathcal{A})$ recognised by an alternating automaton $\mathcal{A}$ is at most two projective levels higher then the complexity of the acceptance condition of $\mathcal{A}$. Therefore, no alternating automata model with a fixed projective acceptance condition is able to capture $\mathrm{MSO} + \mathrm{U}$. In particular alternating ωBS-automata are not an automata model for $\mathrm{MSO} + \mathrm{U}$.

Of course $\mathrm{MSO} + \mathrm{U}$ may still be decidable. However, most of the decidability results in language theory (including ω-regular languages, regular tree languages [Rab68], $\mathrm{WMSO} + \mathrm{U}$ [Boj11], and $\mathrm{WMSO} + \mathrm{R}$ [BT09]) lead through a construction of an appropriate automata model. Results of this paper show that there is no such model that is *simple* from the descriptive point of view.

Results

The following list summarises results presented in this work.

1. All languages definable by ωB, ωS, ωBS-automata are respectively in $\mathbf{\Sigma}_3^0, \mathbf{\Pi}_3^0, \mathbf{\Sigma}_4^0$.

2. There are languages definable by ωB, ωS, ωBS-automata that are hard for their respective classes.

3. All languages definable by alternating ωBS-automata are at the second level of the projective hierarchy.

4. Alternating ωBS-automata recognise languages complete for arbitrarily high finite levels of the Borel hierarchy.

5. The $\mathrm{MSO} + \mathrm{U}$ logic defines languages arbitrarily high in the projective hierarchy.

In particular these results show that:

- Alternating ωBS-automata have strictly greater expressive power then the boolean combinations of nondeterministic ωBS-automata.
- The MSO + U logic defines languages not recognised by alternating ωBS-automata.

1 Basic Notions

By ω we will denote the set of natural numbers, as well as the smallest infinite ordinal.

1.1 Logic

We assume familiarity with the *Monadic Second Order Logic* (MSO). Fix an alphabet A. We denote positions of ω-words using symbols $x, y, \ldots$ and sets of positions with symbols $X, Y, \ldots$. For $a \in A$, the unary predicate P_a holds in all positions of the word where an a stands. It is well known that languages that can be described by this logic, called ω-regular languages, are exactly the sets recognized by nondeterministic Büchi automata or, equivalently, deterministic Muller or parity automata (see [Tho96] for a survey reference).

MSO + U allows building formulae using MSO constructs and an additional quantifier U, called the *unbounding quantifier*, defined as follows. The formula $\mathsf{U}X.\varphi(X)$ holds in a word w if $\varphi(X)$ is satisfied for arbitrarily large finite sets X of positions. Formally, $\mathsf{U}X.\varphi(X)$ is equivalent to:

$$\bigwedge_{n\in\omega} \exists X.\ (\varphi(X)\ \wedge\ n < |X| < \infty)$$

For example the language L_B defined in the introduction can be expressed by the formula:

$$\neg\mathsf{U}X.\,(\forall x{\in}X.P_a(x) \wedge \forall x{<}y{<}z.\,(x{\in}X \wedge z{\in}X) \implies y{\in}X)$$

1.2 Topology

Fix an alphabet A — any finite or countable set of *letters*. By A^* we denote the set of finite words over A, i.e. finite sequences of elements in A, whereas A^ω denotes the set of infinite sequences. For a word $s \in A^* \cup A^\omega$, the restriction of s to its first n letters is denoted by $s\restriction_n$. For the length of s we use a notation $|s|$. If $s \in A^*$ and $t \in A^* \cup A^\omega$ by $s \cdot t$ we explicitly denote the concatenation of s and t.

We treat A^ω as a topological space. A basic open set is determined by a prefix $s \in A^*$ and is of the form $s \cdot A^\omega$. Other open sets are obtained by taking unions of basic open sets. If A is finite, this topological space is homeomorphic (i.e. topologically isomorphic) to the Cantor space. If A is countably infinite then the space is homeomorphic to the Baire space ω^ω.

1.3 Borel and Projective Hierarchy

Let us fix a topological space X. The Borel hierarchy is defined inductively:

- $\mathbf{\Sigma}^0_1(X)$ denotes the class of open subsets of X,
- $\mathbf{\Pi}^0_1(X)$ denotes the class of closed subsets of X (the complements of open sets),

for a countable ordinal α:

- $\mathbf{\Sigma}^0_\alpha(X)$ is the class of countable unions of sets from $\bigcup_{\beta<\alpha} \mathbf{\Pi}^0_\beta(X)$,
- $\mathbf{\Pi}^0_\alpha(X)$ is the class of countable intersections of sets from $\bigcup_{\beta<\alpha} \mathbf{\Sigma}^0_\beta(X)$.

Note that for each α the class $\mathbf{\Sigma}^0_\alpha(X)$ consists exactly of the complements of the languages from $\mathbf{\Pi}^0_\alpha(X)$. The classes constitute a hierarchy — each class is included in all classes with greater subindex (see Figure 1). An important fact about the hierarchy states that the inclusions are strict. The class of *Borel sets*, defined as

$$\mathrm{Bor}(X) = \bigcup_{\alpha<\omega_1} \mathbf{\Sigma}^0_\alpha(X) = \bigcup_{\alpha<\omega_1} \mathbf{\Pi}^0_\alpha(X), \quad \text{where } \omega_1 \text{ is the smallest uncountable ordinal,}$$

is the least class closed under countable boolean operations that contains all open sets. Proofs and details about the Borel hierarchy can be found e.g. in [Sri98, Chapter 3.6]. If the space is clear from the context we will omit it and write Bor, $\mathbf{\Sigma}^0_\alpha$, $\mathbf{\Pi}^0_\alpha$, etc.

The class of Borel sets is not closed under projection. Each set that is a projection of a Borel set is called *analytic*. The class of analytic sets is denoted by $\mathbf{\Sigma}^1_1$. Formally:

$$\mathbf{\Sigma}^1_1(X) = \left\{P \subseteq X : \exists B {\in} \mathrm{Bor}(\omega^\omega {\times} X).\ P{=}\pi_2(B)\right\},$$

where π_2 is the projection on the second coordinate. The superscript 1 means that the class is a part of the projective hierarchy. The rest of the projective hierarchy is defined as follows:

$$\begin{array}{ll} \mathbf{\Pi}^1_i & \text{consists of the complements of the sets from } \mathbf{\Sigma}^1_i, \\ \mathbf{\Sigma}^1_{i+1} & \text{consists of the projections of the sets from } \mathbf{\Pi}^1_i. \end{array}$$

The sets from the class $\mathbf{\Pi}^1_1$ are called *co-analytic*.

The Borel hierarchy together with the projective hierarchy constitute the so-called *boldface hierarchy*, see the diagram on Figure 1.

1.4 Topological Complexity

A *topological complexity class* $\mathbf{C}$, for the needs of this paper is any of the classes of the boldface hierarchy. Analogously to the complexity theory, we have the notions of *reductions* and *completeness*. Let A, B be two alphabets and let

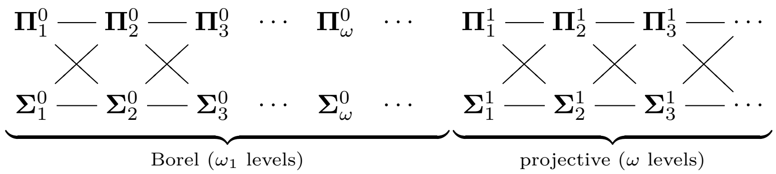

Figure 1: The boldface hierarchy.

$K \subseteq A^\omega$ and $L \subseteq B^\omega$. We say that a continuous mapping $f\colon A^\omega \to B^\omega$ is a *reduction* of K to L if $K{=}f^{-1}(L)$. It is a simple property of continuous mappings that if L belongs to a topological complexity class $\mathbf{C}$ then so does K. The language L is called $\mathbf{C}$-*hard* iff any set $K \in \mathbf{C}$ can be reduced to L. We say that L is $\mathbf{C}$-*complete* if additionally $L \in \mathbf{C}$.

The following fact presents a standard way of using the above notions.

Fact 1.1. *If $\mathcal{C} \subsetneq \mathcal{D}$ are two complexity classes and L is $\mathcal{D}$-hard, then $L \notin \mathcal{C}$.*

Proof. Assume to the contrary that $L \in \mathcal{C}$. Take any language $K \in \mathcal{D} \setminus C$. Since L is $\mathcal{D}$-hard, we can write $K{=}f^{-1}(L)$ for an appropriate continuous mapping f. By the above observation, it implies that $K \in \mathcal{C}$, which gives a contradiction. ■

2 Construction of $\mathbf{\Sigma}^1_i$-hard Languages

In this section we inductively construct a sequence of languages $(H_i)_{i\in\omega}$. We show that for each $i \in \omega$ the language H_i is $\mathrm{MSO}+\mathrm{U}$ definable and $\mathbf{\Sigma}^1_i$-hard. Therefore, we prove the following theorem.

Theorem 2.1. *For every $i > 0$ there exists an* $\mathrm{MSO}+\mathrm{U}$ *formula φ_i such that the language $L(\varphi_i)$ is $\mathbf{\Sigma}^1_i$-hard.*

In our construction we use a sequence IF^i of languages of “multi-branching” trees (i.e. trees on ω^i). First, we prove that for each i the language IF^i is $\mathbf{\Sigma}^1_i$-hard. Then we inductively show that the languages can be reduced to ω-word languages H_i definable in $\mathrm{MSO}+\mathrm{U}$. We use a function r_{i-1} reducing IF^{i-1} to H_{i-1} to construct a reduction c_i of IF^i to the language $\mathrm{EPath}\left(\overline{H_{i-1}}\right)$ of trees that have a branch labelled with a word $w \notin H_{i-1}$. Then we again code such labelled trees in ω-words.

To give some more details let us fix a finite alphabet $B_0 = \{a, |_0, b\}$ and define inductively $B_i = B_{i-1} \cup \{[_{i-1}, |_i,]_{i-1}\}$.

The inductive construction begins from step $i = 1$ and in each step the picture looks as follows:

$$\begin{array}{ccccc}
\mathrm{Tr}^i & \xrightarrow{c_i} & \mathrm{Tr}_{B^+_{i-1}} & \xrightarrow{d_i} & (B^+_i)^\omega \\
\cup\!| & & \cup\!| & & \cup\!| \\
\mathrm{IF}^i & & \mathrm{EPath}\left(\overline{H_{i-1}}\right) & & H_i
\end{array}$$

The construction ensures that $d_i^{-1}(H_i) = \mathrm{EPath}\left(\overline{H_{i-1}}\right)$ and $c_i^{-1}\left(\mathrm{EPath}\left(\overline{H_{i-1}}\right)\right) = \mathrm{IF}^i$. The elements of the above diagram are defined in following sections.

The rest of this chapter is devoted to defining elements of the above diagram and proving their properties.

2.1 Trees

Definition 2.2. *Let* Tr^i *be the set of all trees on* ω^i*, i.e. prefix-closed subsets of* $\left(\omega^i\right)^*$*.*

Additionally we consider labellings of the full ω*-branching tree. For a given set* X *an* X*-labelled* ω*-branching tree is any function* $t\colon \omega^* \to X$*. The set of all such trees is denoted by* Tr_X*.*

Fix an order $\sqsubseteq$ of type ω on ω^*, such that $\omega^* = \{v_0, v_1, \ldots\}$. Additionally assume that for all $n \in \omega$ we have $|v_n| \leq n$. There are infinitely many vertices of length 1 so it is possible.

Definition 2.3. *Consider* $i > 0$*, a tree* $t \in \mathrm{Tr}^{i+1}$ *and a finite or infinite word* $w \in \omega^* \cup \omega^\omega$*. We define the section* $t\restriction_w \in \mathrm{Tr}^i$ *of the tree* t *as follows*

$$t\restriction_w = \left\{w' \in (\omega^i)^* : |w'| \leq |w| \wedge (w\restriction_{|w'|} \times w') \in t\right\},$$

where

$$(w_0, w_1, w_2, \ldots) \times (w'_0, w'_1, w'_2, \ldots) = (w_0 \cdot w'_0, w_1 \cdot w'_1, w_2 \cdot w'_2, \ldots).$$

The dots in the above definition can stand for a finite or an infinite sequence.

Figure 2 presents the first two levels of a tree t on ω^2 i.e. $t \in \mathrm{Tr}^2$. The children of the root are arranged into a two dimensional grid. Given a sequence $w \in \omega^* \cup \omega^\omega$ the section $t\restriction_w \in \mathrm{Tr}^1$ is defined as the one dimensional tree obtained by selecting particular rows from the grids of children on every level. The position of the selected row is defined by successive values of w. For example the children of the root in $t\restriction_w$ come from w_0'th row of the presented grid.

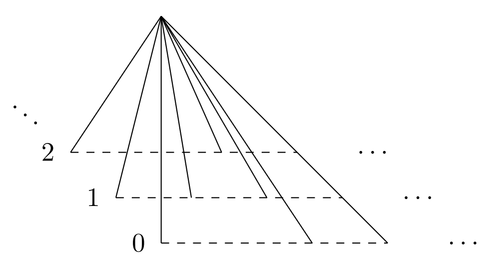

Figure 2: A multi branching tree on ω^2.

Observe that if w is a finite word, $t\restriction_w$ is a finite-depth tree — its depth is bounded by $|w|$.

Definition 2.4. *For a tree $t \in \mathrm{Tr}_X$ and an infinite word $\alpha \in \omega^\omega$, let*

$$t(\alpha) = (t(\alpha\restriction_0), t(\alpha\restriction_1), \ldots) \in X^\omega.$$

Definition 2.5. *We define inductively $\mathrm{IF}^i \subseteq \mathrm{Tr}^i$.*

Let IF^1 be the set of all trees $t \in \mathrm{Tr}^1$ that contain an infinite branch.

Take $i > 0$. Let IF^{i+1} be a set of all trees $t \in \mathrm{Tr}^{i+1}$ such that there exists an infinite word $\alpha \in \omega^\omega$ such that

$$t\restriction_\alpha \notin \mathrm{IF}^i.$$

Fact 2.6. *For each $i \geq 1$ the set IF^i is a $\boldsymbol{\Sigma}^1_i$-complete subset of Tr^i.*

Proof. (Upper bound) IF^1 is a set of Ill-Founded trees (usually denoted by IF), that is well known to be $\boldsymbol{\Sigma}^1_1$-complete (see e.g. [Kec95, Theorem 27.1]). We proceed by induction. Assume that $\mathrm{IF}^i \in \boldsymbol{\Sigma}^1_i$. Then the set

$$P_i = \left\{(\alpha, t) \in \omega^\omega \times \mathrm{Tr}^{i+1} : t\restriction_\alpha \notin \mathrm{IF}^i\right\} \in \boldsymbol{\Pi}^1_i.$$

Note that IF^{i+1} is a projection of P_i, so it is in $\boldsymbol{\Sigma}^1_{i+1}$.

(Hardness) We have to show that each $\boldsymbol{\Sigma}^1_i$ set in ω^ω continuously reduces to IF^i.

As we know (see e.g. [Kec95, Exercise 14.3]), each analytic ($\boldsymbol{\Sigma}^1_1$) set in a space X is a projection of a closed set in $\omega^\omega \times X$. Recall that, by definition, each $\boldsymbol{\Sigma}^1_{i+1}$ set is a projection of some $\boldsymbol{\Pi}^1_i$ set. Therefore, each $\boldsymbol{\Sigma}^1_i$ set in ω^ω is of the form[1]:

$$S = \{x : \exists x_1 \in \omega^\omega . \neg\exists x_2 \in \omega^\omega . \neg\exists x_3 \in \omega^\omega \ldots . \neg\exists x_i \in \omega^\omega . (x_1, x_2, \ldots, x_i, x) \in F_S\},$$

for some closed set $F_S \in (\omega^\omega)^{i+1}$. The formula unravels to:

$$\begin{array}{ll} \exists x_1 . \forall x_2 . \exists x_3 . \ldots . \exists x_i . (x_1, x_2, \ldots, x_i, x) \in F_S & \text{if } i \text{ is odd, and to:} \\ \exists x_1 . \forall x_2 . \exists x_3 . \ldots . \forall x_i . (x_1, x_2, \ldots, x_i, x) \notin F_S & \text{if } i \text{ is even.} \end{array}$$

The set F_S can be seen as a set in the space $\left(\omega^{i+1}\right)^\omega$, by simple transposition. This space is obviously homeomorphic to the Baire space ω^ω. Each closed set in the Baire space can be expressed as the set of branches of some tree (see e.g. [Kec95, Proposition 2.4]). So there is $t_S \in \mathrm{Tr}^{i+1}$ such that:

$$F_S = \left\{(x_1 \times x_2 \times \cdots \times x_{i+1}) \in \left(\omega^{i+1}\right)^\omega : \forall n \in \omega . \, (x_1\restriction_n \times x_2\restriction_n \times \cdots \times x_{i+1}\restriction_n) \in t_S\right\} \tag{2.1}$$

To simplify the notation, for a tree t on set X, by $[t] \subseteq X^\omega$ we denote the set of infinite branches of t. Using this notation, the above equation can be formulated as

$$F_S = [t_S].$$

[1] Formally, for $i = 1$ the formula takes the form $S = \{x : \exists x_1 \in \omega^\omega . (x_1, x) \in F_S\}$.

We will use the tree t_S to define the needed reduction. Let $f : \omega^\omega \to \mathrm{Tr}^i$ be defined as follows:

$$f(x) = \left\{(v_1 \times v_2 \times \cdots \times v_i) \in \left(\omega^i\right)^k : (v_1 \times v_2 \times \cdots \times v_i \times x\restriction_k) \in t_S, k \in \omega\right\}$$

To determine whether a vertex at some level k belongs to $f(x)$ we only need to know the first k numbers in the sequence x, so the function is continuous. To prove that this is a reduction of S to IF^i we need:

$$f(x) \in \mathrm{IF}^i \iff x \in S \tag{2.2}$$

Now we will take a closer look at the sets IF^i. Observe that:

$$\mathrm{IF}^i = \{t : \exists x_1.\forall x_2.\exists x_3.\dots\exists x_i.(x_1 \times x_2 \times \cdots \times x_i) \in [t]\} \qquad \text{if } i \text{ is odd, and:}$$
$$\mathrm{IF}^i = \{t : \exists x_1.\forall x_2.\exists x_3.\dots\forall x_i.(x_1 \times x_2 \times \cdots \times x_i) \notin [t]\} \qquad \text{if } i \text{ is even.}$$

So the quantifier structure is the same as in case of the above representation of S. Therefore, to obtain (2.2), it *suffices* to show that for any fixed x_1, x_2, $\dots$, x_i:

$$(x_1 \times x_2 \times \cdots \times x_i) \in [f(x)] \iff (x_1, x_2, \dots, x_i, x) \in F_S.$$

By (2.1) it is equivalent to:

$$(x_1 \times x_2 \times \cdots \times x_i) \in [f(x)] \iff (x_1 \times x_2 \times \cdots \times x_i \times x) \in [t_S].$$

But the latter follows immediately from the definition of f. ■

2.2 Functions c_i, d_i

In this section we define functions c_i, d_i. The idea is that both these functions are continuous and $1-1$. Their task is to present a tree $t \in \mathrm{Tr}^i$ as an infinite word in such a way that an $\mathrm{MSO}+\mathrm{U}$ formula can determine whether $t \in \mathrm{IF}^i$ or not.

Recall our inductively defined alphabets $B_0 = \{a, |_0, b\}$, $B_i = B_{i-1} \cup \{[_{i-1}, |_i,]_{i-1}\}$ and define:

Definition 2.7. *For a node $u = (u_1, u_2, \dots, u_m) \in \omega^*$ of a tree, we will call the word $a^{u_1}ba^{u_2}b\dots ba^{u_m}b$ the* address *of u in the tree.*

Let an i-block *be a word of the form $[_i w|_i w']_i$ where $w \in \{a,b\}^*$ and $w' \in (B_i \setminus \{|_i\})^+$. We will call the word w the* address *of this i-block (since it will be interpreted as an address of a node in a tree) and the word w' the* body *of this i-block.*

Functions d_i

Take any $i > 0$. We encode a tree $t \in \mathrm{Tr}_{B^+_{i-1}}$ into a word $d_i(t) \in (B^+_i)^\omega$ in the following way. Take a tree $t \in \mathrm{Tr}_{B^+_{i-1}}$ and a vertex v_n, i.e the n'th vertex with respect to the order $\sqsubseteq$. Let $v_n = (u_1, u_2, \ldots, u_m)$ and let $w_0, w_1, \ldots, w_m \in B^+_{i-1}$ be the list of labels of t on the path from the root to v_n. Then

$$d_i(t)_n := a^{u_1} b a^{u_2} b \ldots b a^{u_m} b \,|_i\, [_{i-1} w_0]_{i-1} \cdot [_{i-1} w_1]_{i-1} \cdot \ldots \cdot [_{i-1} w_m]_{i-1} \in B^+_i.$$

Intuitively $d_i(t)_n$ encodes the vertex v_n in t. Such an encoding consists of two parts: the part before $|_i$ is the address of v_n in the tree, while the part after $|_i$ is intended to store labels in t on the path from the root to v_n as $(i-1)$-blocks. The fact that we store not only the label but also the address of the given vertex in this coding will be crucial for the following parts of the construction.

Functions r_i, c_i

Now we can inductively define functions $c_i \colon \mathrm{Tr}^i \to \mathrm{Tr}_{B^+_{i-1}}$ and $r_i = d_i \circ c_i$.

Take a tree $t \in \mathrm{Tr}^1$ and a vertex $v = (u_1, u_2, \ldots, u_m) \in \omega^*$. Define $c_1(t) \in \mathrm{Tr}_{B^+_0}$ by an equation

$$c_1(t)(v) = \begin{cases} a^{u_1} b a^{u_2} b \ldots b a^{u_m} b \,|_0\, a & \text{if } v \in t, \\ a^{u_1} b a^{u_2} b \ldots b a^{u_m} b \,|_0\, b & \text{if } v \notin t. \end{cases}$$

For $i > 1$ take a tree $t \in \mathrm{Tr}^i$ and a vertex $v \in \omega^*$. Let

$$c_i(t)(v) = (r_{i-1}(t\restriction_v))_{|v|} \in B^+_{i-1}.$$

Lemma 2.8. *Functions c_i, d_i defined above are continuous.*

Proof. For d_i it holds by the definition. The continuity of c_i can be proved by induction together with the continuity of r_i, since they cyclically depend on each other. Function r_{i+1} is continuous as a composition of continuous functions, likewise c_i at each coordinate v is a composition of continuous operations: $\_\restriction_v$, r_{i-1}, $\_{}_{|v|}$. ■

The following lemma states that functions r_i are in some sense *sequential*.

Lemma 2.9. *For any $i > 0$ and any $m \in \omega$:*
if $t_1, t_2 \in \mathrm{Tr}^i$ agree on all $v \in (\omega^i)^*$ such that $|v| \leq m$ then

$$r_i(t_1)_m = r_i(t_2)_m.$$

Proof. Recall that $r_i(t) = d_i(c_i(t))$. First observe that for a given tree $t' \in \mathrm{Tr}_X$, by the definition of d_i, the value $d_i(t')_m$ depends only on v_m and the labels of t' on the path from the root to v_m.

Now use an induction on i and consider labels of $c_i(t_1)$ and $c_i(t_2)$ on the path from the root to v_m. For $i = 1$ they depend only on t_1, t_2 up to the depth of $|v_m|$, and $|v_m| \leq m$, thanks to our assumption about the order $\sqsubseteq$.

Take $i > 1$ and a vertex $v \preccurlyeq v_m$ (where $\preccurlyeq$ denotes the prefix order). By the definition $c_i(t)(v) = r_{i-1}(t\restriction_v)_{|v|}$. So, by the inductive assumption, this value also depends only on t at the depth of at most $|v| \leq |v_m| \leq m$. ■

From the above lemma we conclude that the labels on each branch $\alpha \in \omega^\omega$ in $c_i(t)$ code the tree $t\restriction_\alpha$. Formally:

Lemma 2.10. *For $i > 1$, a given tree $t \in \mathrm{Tr}^i$ and an infinite branch $\alpha \in \omega^\omega$ we have:*

$$c_i(t)(\alpha) = r_{i-1}(t\restriction_\alpha) \in \left(B^+_{i-1}\right)^\omega .$$

Proof. Take any $m \in \omega$ and consider $v = \alpha\restriction_m \in \omega^m$. By the definition

$$\left(c_i(t)(\alpha)\right)_m = c_i(t)(\alpha\restriction_m) = \left(r_{i-1}(t\restriction_v)\right)_m .$$

Since $t\restriction_v$ and $t\restriction_\alpha$ agree on all vertices up to the depth m, by Lemma 2.9, we have

$$\left(r_{i-1}(t\restriction_v)\right)_m = \left(r_{i-1}(t\restriction_\alpha)\right)_m .$$

■

2.3 Languages H_i

In this section we define MSO + U formulae φ_i. The i'th formula φ_i expresses properties of infinite words over B_{i+1}.

All the above work is done in the spaces $(B^+_i)^\omega$. Since we want to build MSO + U formulae over finite signatures, we need to work with finite alphabets. To achieve this we will use one additional encoding which is simply a kind of concatenation.

For $i \geq 0$ consider $j_i \colon (B^+_i)^\omega \to B^\omega_{i+1}$ defined as follows

$$j_i(w_0, w_1, \ldots) = [_i w_0]_i \cdot [_i w_1]_i \cdot \ldots$$

Of course functions j_i defined above are continuous and $1-1$.

Recall that the address of an i-block is supposed to represent a node of a tree (see Definition 2.7). We say that such an i-block (or its address) *corresponds* to this node.

We will call a set A of addresses of nodes:

deep if the number of letters b in elements of A is unbounded,

thin if for any set P of some prefixes of elements of A such that the number of letters b in elements of P is bounded, the lengths of sequences a^* in elements of P are bounded.

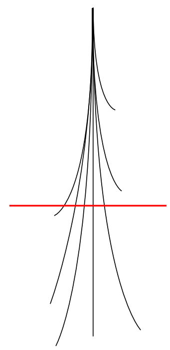

Figure 3: An illustration of the thin property — any section of finite depth contains only finitely many prefixes of branches in A.

The following remark provides a way of using the above properties.

Remark 2.11. *A tree $t \subseteq \omega^*$ has an infinite branch if and only if there is a thin and deep set A of addresses of some nodes in t.*

Proof. First assume that t has an infinite branch $\alpha \in \omega^\omega$. Take as A the set of addresses of vertices in $\{\alpha\restriction_n : n \in \omega\}$. Of course such A is deep. We show that A is thin. Consider any set P of prefixes of addresses in A, such that the number of letters b in elements of P is bounded by some number $k \in \omega$. In that case, lengths of sequences a^* in P are bounded by $\max_{n \le k}\ \alpha_n$: in each element of A the sequence a^* before the n'th letter b has length α_{n-1}.

Now take a thin and deep set A of addresses of some nodes of t. We identify elements of A with those nodes, i.e. $A \subseteq t$. Consider as T the closure of A under prefixes, i.e.:

$$T = \{v \in \omega^* : \exists_{v' \in A}\ v \preccurlyeq v'\}\,.$$

Then T is an infinite tree, because A is deep. Additionally, at each level $k \in \omega$, there are only finitely many vertices in $T \cap \omega^k$, by thinness of A. So T is a finitely branching tree. Therefore, by König's Lemma, T contains an infinite branch α. But $T \subseteq t$, so α is also an infinite branch of t. ■

Formulae

Observe that both properties *deepness* and *thinness* of a set of addresses of a sequence of i-blocks can be expressed in $\mathrm{MSO} + \mathrm{U}$. It is because in those definitions we only use regular properties and properties like *the number of letters b is unbounded* or *the length of sequences a^* is bounded*.

We now define a series of $\mathrm{MSO} + \mathrm{U}$ formulae φ_i. It is easy to see that we can express in MSO that a given word $\alpha \in (B_{i+1})^\omega$ is of the form $b_0 \cdot b_1 \cdot \ldots$ such that each b_n is an i-block. We implicitly assume that all formulae φ_i express it.

Let φ_0 additionally express that a given word is not of the form

$$([_0\ (a^* b)^*\ |_0\ a\ ]_0)^\omega\,.$$

For $i > 0$, let φ_i express the following property:

There exists a set G containing only whole i-blocks such that:

1. the set of addresses of the i-blocks of G is deep,
2. the set of addresses of the i-blocks of G is thin,
3. the bodies of the i-blocks of G, when concatenated, form an infinite word that satisfies $\neg\varphi_{i-1}$.

Take $i \geq 0$. Since $L(\varphi_i) \subseteq B_{i+1}^{\omega}$, we can define

$$H_i = j_i^{-1}\left(L(\varphi_i)\right) \subseteq (B_i^{+})^{\omega}.$$

Languages H_i defined above are (up to the j operator) $\mathrm{MSO}+\mathrm{U}$ definable. We will use one important property of languages H_i.

Definition 2.12. *A language $L \subseteq X^{\omega}$ is monotone if for any $\alpha, \beta \in X^{\omega}$*

$$\{\alpha_n : n \in \omega\} \subseteq \{\beta_n : n \in \omega\} \quad \Longrightarrow \quad (\alpha \in L \Rightarrow \beta \in L)\,.$$

Note, that belonging to a monotone language depends only on the set of letters occurring in a word, namely

Fact 2.13. *If $L \subseteq X^{\omega}$ is a monotone language, then for any $\alpha, \beta \in X^{\omega}$ the following holds*

$$\{\alpha_n : n \in \omega\} = \{\beta_n : n \in \omega\} \quad \Longrightarrow \quad (\alpha \in L \Leftrightarrow \beta \in L)\,.$$

Lemma 2.14. *Languages $H_i \subseteq (B_i^{+})^{\omega}$ are monotone.*

Proof. For $i = 0$ it is obvious. For $i > 0$ formula φ_i expresses that there exists a set of i-blocks such that this set satisfies some additional property. Moreover, it does not matter in what order the i-blocks appear. ■

2.4 Reductions

In this section we show that r_i is a reduction of IF^i to H_i. We do it in two steps.

Definition 2.15. *For $L \subseteq X^{\omega}$ let $\mathrm{EPath}\,(L) \subseteq \mathrm{Tr}_X$ be a set of such trees t that there exists an infinite word $\alpha \in \omega^{\omega}$ such that*

$$t(\alpha) \in L.$$

In other words $\mathrm{EPath}\,(L)$ is the set of trees that contain an infinite branch such that labels on this branch form a word in L.

Lemma 2.16. *For $i > 0$ the function $d_i \colon \mathrm{Tr}_{B_{i-1}^{+}} \to (B_i^{+})^{\omega}$ is a reduction of $\mathrm{EPath}\left(\overline{H_{i-1}}\right)$ to H_i.*

Proof. We have to prove that for any $t \in \mathrm{Tr}_{B_i^+}$

$$t \in \mathrm{EPath}\left(\overline{H_{i-1}}\right) \iff d_i(t) \in H_i.$$

First assume that $t \in \mathrm{EPath}\left(\overline{H_{i-1}}\right)$. Let $\alpha \in \omega^\omega$ be a branch such that $t(\alpha) \notin H_{i-1}$. Let $w = j_i(d_i(t)) \in (B_{i+1})^\omega$. We show that $w \models \varphi_i$. Take as G the set containing i-blocks corresponding to vertices of α. Then the set of addresses of i-blocks of G is obviously thin and deep (one vertex at each level of the tree). Additionally, the set of $(i-1)$-blocks occurring in bodies of i-blocks in G is exactly the set

$$\left\{ [_{i-1} \cdot \ (t(\alpha))_n \ \cdot]_{i-1} \ : n \in \omega \right\}.$$

Language H_{i-1} is monotone, so, by Fact 2.13, since $t(\alpha) \notin H_{i-1}$, the set G satisfies point 3 in the definition of φ_i.

The other direction is a little more tricky. Assume that $j_i(d_i(t)) \models \varphi_i$. Let G be as in the definition of φ_i. Then the set of addresses of i-blocks of G is deep and thin. Let $B \subseteq \omega^*$ be the set of nodes corresponding to these addresses and let T be the closure of B under prefixes, i.e.:

$$T = \left\{ v \in \omega^* : \exists_{v' \in B} \ v \preccurlyeq v' \right\}.$$

As in Remark 2.11, there exists an infinite branch $\alpha \in \omega^\omega$ of T. Observe that the set

$$\left\{ [_{i-1} \cdot \ (t(\alpha))_n \ \cdot]_{i-1} \ : n \in \omega \right\}$$

is contained in the set of $(i-1)$-blocks in bodies of i-blocks in G. Because of the monotonicity of H_{i-1} and point 3 in the definition of φ_i, $t(\alpha) \notin H_{i-1}$. ■

Lemma 2.17. *For $i > 0$ the function c_i is a reduction of IF^i to $\mathrm{EPath}\left(\overline{H_{i-1}}\right)$.*

Proof. Take $i = 1$. A tree $t \in \mathrm{Tr}^1$ contains an infinite branch if and only if $c_1(t)$ contains a branch labelled by words of the form $(a^*b)^*|_0 a$ if and only if $c_1(t) \in \mathrm{EPath}\left(\overline{H_0}\right)$.

Induction step: $i > 1$. Take a tree $t \in \mathrm{Tr}^i$. The following conditions are equivalent:

	$t \in \mathrm{IF}^i$	
$\exists_{\alpha \in \omega^\omega}$	$t \restriction_\alpha \notin \mathrm{IF}^{i-1}$	by the definition of IF^i
$\exists_{\alpha \in \omega^\omega}$	$c_{i-1}(t \restriction_\alpha) \notin \mathrm{EPath}\left(\overline{H_{i-2}}\right)$	by the inductive assumption
$\exists_{\alpha \in \omega^\omega}$	$r_{i-1}(t \restriction_\alpha) \notin H_{i-1}$	by Lemma 2.16
$\exists_{\alpha \in \omega^\omega}$	$c_i(t)(\alpha) \notin H_{i-1}$	by Lemma 2.10
	$c_i(t) \in \mathrm{EPath}\left(\overline{H_{i-1}}\right)$	by the definition of $\mathrm{EPath}\,(L)$.

■

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1 Take $i \in \omega$ and φ_i as defined above. Functions c_i, d_i, j_i are continuous by Lemma 2.8 and the definition of j_i. Moreover, using the definition of H_i and Lemmas 2.17, 2.16 their composition reduces IF^i to $L(\varphi_i)$. Thanks to Fact 2.6, the set IF^i is $\mathbf{\Sigma}^1_i$-hard. ■

3 Nondeterministic ωBS-automata

In this section we give exact estimations on the topological complexity of ωBS-, ωB- and ωS-regular languages.

First we present ωBS-automata as described in [BC06,Boj10]. They define a strict subclass of MSO + U, but, as far as we know, it is the greatest considered subclass of MSO + U with decidable emptiness.

An *ωBS-automaton* $\mathcal{A}$, as other nondeterministic finite automata, has a finite input alphabet A, a finite set Q of states and an initial state q_I. Apart from that it is equipped with a finite set Γ of counters. The counters can only be updated and cannot be read during the run. They are used by the acceptance condition. A transition of the automaton is a transformation of states, as in standard NFA's, and additionally a finite sequence of counter updates. A counter update can be either an increment or a reset of a counter $c \in \Gamma$.

The value of a counter c is initially set to 0 and is incremented or reset according to the transitions in a run. For $c \in \Gamma$ and a run ρ we define a sequence $val_\rho(c)$, where $val_\rho(c)_i$ is the value of counter c right before its i-th reset in the run ρ. Note that if the counter c is reset only finitely many times then the sequence $val_\rho(c)$ is finite.

The acceptance condition of an ωBS-automaton is a boolean combination of constraints that can be of one of the forms:

$$\limsup_i val_\rho(c)_i < \infty \qquad\qquad \liminf_i val_\rho(c)_i = \infty$$

The first constraint is called the *B-condition* (bounded), the second — the *S-condition* (strongly unbounded). In order that lim inf and lim sup make sense, the constraints implicitly require the corresponding sequences to be infinite.

It is a simple observation that the negation of a B-condition can be simulated using an S-condition and nondeterminism, and vice versa. Thanks to this fact we can consider automata with acceptance conditions that are *positive* boolean combinations of S- and B-conditions, without loss of expressive power.

We will use the notation $B(c)$ for the B-condition and $S(c)$ for the S-condition imposed on a counter c.

If the acceptance condition of an automaton is a positive boolean combination of B-conditions, the automaton is called an *ωB-automaton*. We similarly define *ωS-automata*.

Languages recognized by ωBS-automata (respectively ωB-automata, ωS-automata) are called *ωBS-regular* (respectively *ωB-regular*, *ωS-regular*). An important result of [BC06] is that the complement of an ωB-regular language is an ωS-regular language, and vice versa. The result is much more involved than the above remark of the duality of B-condition and S-condition, because, by the straightforward reduction, while negating a nondeterministic automaton we obtain a co-nondeterministic (universal) one, not a nondeterministic one. Both the classes are extensions of the class of ω-regular languages since the Büchi condition can be simulated by either a B-condition or an S-condition.

Example 3.1. *The language L_S defined in the introduction can be recognized by an ωS-automaton. The automaton has one state and uses one counter that is increased when reading a letter a and is reset after each b. The acceptance condition is simply an S-condition on the only counter.*

3.1 Complexity of ωB- and ωS-regular Languages

Theorem 3.2. *Each ωB-regular language is in $\mathbf{\Sigma}^0_3$.*

Proof. Fix an ωB-automaton $\mathcal{A}$ recognizing a language L, and let us first assume that its accepting condition is a conjunction of B-conditions, i.e. is of the form:

$$\bigwedge_{c \in \Gamma_B} B(c)$$

Each of the considered counters is bounded iff there is a common bound k for all of them. Therefore L can be defined as:

$$\begin{aligned} L &= \left\{w : \exists \rho. \textstyle\bigwedge_{c\in\Gamma_B} val_\rho(c) \text{ is infinite but bounded}\right\} \\ &= \bigcup_k \underbrace{\left\{w : \exists \rho. \textstyle\bigwedge_{c\in\Gamma_B} val_\rho(c) \text{ is infinite and bounded by } k \right\}}_{L_k}, \end{aligned}$$

where the quantification is over the set of all runs of $\mathcal{A}$ on w.

It is easy to see that for a fixed k, L_k can be recognized by a nondeterministic Büchi automaton. We simply store counter values in states and do not allow them to be incremented above k. The acceptance condition requires each of the counters $c \in \Gamma_B$ to be reset infinitely often. Hence L_k is ω-regular. Since each ω-regular language is a boolean combination of $\mathbf{\Sigma}^0_2$ sets and L is a countable union of such sets, $L \in \mathbf{\Sigma}^0_3$.

In the general form, the acceptance condition of an ωB-automaton is a positive boolean combination of B-conditions. We can write such a condition in disjunctive normal form (DNF). The language accepted by this automaton is a union of languages corresponding to each disjunct. Hence it is in $\mathbf{\Sigma}^0_3$. ■

Thanks to the complementation result of [BC06], we have:

Corollary 3.3. *Each ωS-regular language is in $\mathbf{\Pi}^0_3$.*

The complexity bounds given by Theorem 3.2 and Corollary 3.3 are tight.

Fact 3.4. *There is a $\mathbf{\Sigma}^0_3$-complete set that is ωB-regular and a $\mathbf{\Pi}^0_3$-complete set that is ωS-regular.*

Proof. Because ωB-regular languages are complements of ωS-regular languages, it suffices to show only one of the claims.

We recall that the language L_S is in $\mathbf{\Pi}^0_3$ and ωS-regular. $\mathbf{\Pi}^0_3$-hardness of L_S follows from $\mathbf{\Pi}^0_3$-hardness of set C_3 from Exercise 23.2 in [Kec95] via an obvious reduction. ■

3.2 Complexity of ωBS-regular Languages

In this section we show that the reasoning presented in the previous section can be lifted to the case of automata that can use both S- and B-conditions. This important observation is by Szymon Toruńczyk.

Theorem 3.5. *Each ωBS-regular language is in $\mathbf{\Sigma}^0_4$.*

Proof. The proof, on one hand, will use the result of Corollary 3.3 and, on the other hand, will repeat a reasoning similar to the one from the proof of Theorem 3.2.

Let us fix an ωBS-regular language L and an automaton $\mathcal{A}$ recognizing it. First assume that an acceptance condition of $\mathcal{A}$ is of the form:

$$\bigwedge_{c \in \Gamma_B} B(c) \quad \wedge \quad \bigwedge_{c \in \Gamma_S} S(c)$$

The language L can then be defined by:

$$L = \bigcup_k \underbrace{\left\{ w : \exists \rho. \begin{array}{ll} \bigwedge_{c \in \Gamma_B} val_\rho(c) & \text{is infinite and bounded by } k \\ \bigwedge_{c \in \Gamma_S} val_\rho(c) & \text{converges to } \infty \end{array} \right\}}_{L_k}$$

Note that each L_k language is ωS-regular, hence, by Corollary 3.3, it is in $\mathbf{\Pi}^0_3$. Therefore L, as a countable union of such languages, is in $\mathbf{\Sigma}^0_4$.

A general acceptance condition can be written in disjunctive normal form (DNF). Again, the language accepted by such an automaton is a union of languages corresponding to each disjunct, so it is in $\mathbf{\Sigma}^0_4$. ■

Now we show that the bound is tight. For that we consider the language, that was used in [BC06, Corollary 2.8] to show that the class of ωBS-regular languages is not closed under complements. Let

$$G = \left\{ a^{n_1} b a^{n_2} b \ldots : \begin{array}{l} \text{the sequence } n_1, n_2, \ldots \text{ can be partitioned into} \\ \text{a (possibly finite) bounded subsequence and} \\ \text{a subsequence that is empty or tends to } \infty \end{array} \right\}$$

The following fact is presented as an example in [TL93, page 595]. It can be shown using language S_4 from Exercise 23.6 in [Kec95].

Fact 3.6. *The language G is $\mathbf{\Sigma}^0_4$-complete.*

Now it suffices to note that the language G is ωBS-regular. It is proven in [BC06] (by showing an appropriate ωBS-regular expression), but it is straightforward to construct a nondeterministic ωBS-automaton recognizing it.

4 Alternating ωBS-automata

On the way towards finding a model of automata for the logic MSO + U, alternating ωBS -automata were considered. Thanks to Theorem 2.1 we know that the model is too weak (this is discussed in detail in Chapter 5), but we give here a lower bound for the topological complexity of alternating ωBS -automata.

Alternating ωBS-automata are defined similarly as nondeterministic ωBS-automata. The difference is that the state space Q is partitioned into $Q_\forall$ (universal states) and $Q_\exists$ (existential states). We use standard game semantics for such automata. For a given alternating automaton $\mathcal{A}$ and a word $w \in A^\omega$ we define a two-player game. A play in this game starts in the initial state of the automaton and in the first position of the word and proceeds by applying transitions of the automaton on the word w consistent with the current state and the letter in the current position in the word. Player $\forall$ (respectively $\exists$) chooses transitions when the automaton is in a state from $Q_\forall$ (respectively $Q_\exists$). Finally the play produces an infinite sequence of transitions consistent with consecutive letters of the word. Such a play is winning for $\exists$ if it satisfies the acceptance ωBS-condition of $\mathcal{A}$ — a boolean combination of B- and S-conditions. The word w is accepted by the automaton iff Player $\exists$ has a winning strategy in the above game.

4.1 Languages Complete for the Classes $\mathbf{\Pi}^0_{2n}$

We will now present examples of languages of infinite words complete for the Borel classes $\mathbf{\Pi}^0_{2n}$, which are recognized by alternating ωBS-automata.

To make proofs easier, we will work with the spaces of sequences of vectors of numbers $\mathcal{N}_n = (\omega^n)^\omega$. An easy embedding, described below, will transfer the results into the space of infinite words. For $n = 0$, the above definition gives the space of infinite sequences of empty tuples, i.e. $\mathcal{N}_0 \simeq \{\omega\}$.

Let us fix an alphabet $A = \{a, b, c\}$. We encode a sequence of vectors in the space A^ω. Each vector $(z_n, z_{n-1}, \ldots, z_1)$ is mapped to the word $a^{z_n} b a^{z_{n-1}} b \ldots a^{z_1} c$, and the codes of consecutive vectors are concatenated. We will call the embedding defined this way $W_n \colon \mathcal{N}_n \to A^\omega$.

We will use the following notations to easily operate on sequences of vectors.

- For $\eta \in \mathcal{N}_n$ and $m \in \omega$, let $\eta\restriction_{=m}$ be a subsequence of η consisting of those vectors that have value m at the first coordinate. We will also use the notation $\eta\restriction_{\in S}$ to restrict to set S of values at the first coordinate.
- Let $\pi_{\bar{1}} \colon \mathcal{N}_n \to \mathcal{N}_{n-1}$ be the projection that cuts off the first coordinate from each vector in a given sequence.

Definition 4.1. *For $n > 0$ we define*

$$L_n = \left\{\eta \in \mathcal{N}_n : \exists^\infty_{m_n} \exists^\infty_{m_{n-1}} \ldots \exists^\infty_{m_1} \exists^\infty_x \eta(x) = (m_n, m_{n-1}, \ldots, m_1)\right\},$$

where $\exists^\infty$ stands for "exists infinitely many". Additionally, let $L_0 = \{\omega\} = \mathcal{N}_0$.

The following fact describes the languages L_n in an inductive fashion.

Fact 4.2. *For $n > 0$, a sequence $\eta \in \mathcal{N}_n$ belongs to L_n iff there exist infinitely many $m \in \omega$ such that $\eta\restriction_{=m}$ is an infinite sequence and $\pi_{\bar{1}}(\eta\restriction_{=m}) \in L_{n-1}$.*

We note some important properties of languages L_n:

- *monotonicity*: If $\eta \in \mathcal{N}_n$ and ν is a subsequence of η, then $\nu \in L_n \implies \eta \in L_n$,
- *prefix independence*: For $\eta \in \mathcal{N}_n$ and $\nu \in (\omega^n)^*$, $\eta \in L_n$ iff $\nu\eta \in L_n$.
- *pigeonhole property*: Let $\nu_1, \nu_2, \ldots, \nu_k$ be a partition of a sequence $\eta \in L_n$ into subsequences, then for some $j \in \{1, 2. \ldots, k\}$, $\nu_j \in L_n$.

We give yet another presentation of languages L_n — this time in logical terms. It will serve as a guideline for us in the construction of alternating automata recognizing the languages.

Let $\mathrm{Bnd}_n(X)$ be a second order predicate expressing that the set X of positions in a sequence from $\mathcal{N}_n$ has bounded first coordinate. We inductively build a sequence of MSO formulae using this predicate:

$$\psi_n \;\equiv\; \forall X.\, \mathrm{Bnd}_n(X) \implies \exists Y.\, \mathrm{Bnd}_n(Y) \wedge (X \cap Y = \emptyset) \wedge (\psi_{n-1}|_Y)\,, \qquad (4.1)$$

where $\psi_{n-1}|_Y$ is ψ_{n-1} with all quantifiers restricted to Y and operating on $\mathcal{N}_n$ by ignoring the first coordinates of vectors, and ψ_0 simply states that a sequence is infinite.

Observe that:

Fact 4.3. *For each $n \in \omega$, $L(\psi_n) = L_n$.*

The formulae (4.1) deal with sequences of vectors, but it is easy to rewrite it in such a way that it works on ω-words over A and defines $W_n(L_n)$. It is possible because properties like "being a maximal block of consecutive a's that correspond to the k-th coordinate of one of the vectors in a sequence" are expressible in MSO. It is not hard to observe that the predicate Bnd_n can be defined in MSO + U in this context. We do not discuss it here in detail because it is out of the scope of the paper.

4.2 Topological Complexity

The topological complexity of languages $W_n(L_n)$ is presented as an example (without proof) in [TL93, pages 595–596]. For the sake of completeness, we sketch a proof of the following fact here.

Fact 4.4. *For every $n{>}0$, the language L_n is $\mathbf{\Pi}^0_{2n+2}$-complete.*

The proof is inductive. The following lemma is the basis for the induction.

Lemma 4.5. *Language L_1 is $\mathbf{\Pi}^0_4$-complete.*

Proof. This is an easy consequence of Exercise 23.6 in [Kec95]. Language L_1 is equivalent to the language P_4 presented there. ■

In the induction step we use the following Lemma, that is stated as Theorem 2 in [Kur66, §30.V].

Lemma 4.6. *For every $n > 0$ and set $Y \in \mathbf{\Sigma}^0_n$, there exist pairwise disjoint sets $Y_i \in \mathbf{\Pi}^0_{n-1}$, such that*

$$\bigcup_i Y_i = Y.$$

Proof of Fact 4.4 For $n = 1$ this is a consequence of Lemma 4.5, as mentioned above. For every n, the language L_n is in class $\mathbf{\Pi}^0_{2n+2}$, because quantifier $\exists^\infty_m$ can be written as $\forall_k \exists_{m>k}$.

Let us take $n > 1$, and any $M \in \mathbf{\Pi}^0_{2n+2}(X)$. We construct a continuous reduction of M to L_n.

There is a decreasing sequence of sets $M_i \in \mathbf{\Sigma}^0_{2n+1}$ such that $M = \bigcap_i M_i$. Using Lemma 4.6 we can define sets $M^{(i)}_k \in \mathbf{\Pi}^0_{2n}$ that are (for fixed i) pairwise disjoint, and

$$\bigcup_k M^{(i)}_k = M_i.$$

By inductive assumption, language L_{n-1} is $\mathbf{\Pi}^0_{2n}$-hard, so there are continuous reductions $R^{(i)}_k : X \to \mathcal{N}_{n-1}$ of sets $M^{(i)}_k$ to L_{n-1}.

Let $\iota: \omega^2 \to \omega$ be any bijection. Let us define function $R: X \to \mathcal{N}_n$ that takes $x \in M$ and maps it into the sequence having at any given position $z = \iota(\iota(i,k), m) \in \omega$ value

$$\left(\iota(i,k), \left(R^{(i)}_k(x)\right)_m\right) \in \omega^n,$$

where the first element in braces is a number and the second is an $(n-1)$-tuple of numbers, so they form an n-vector.

This is easy to see that the function defined this way is continuous. Now, it is enough to show that $x \in M \Leftrightarrow R(x) \in L_n$.

By the definition of L_n we know that $R(x) \in L_n$ iff for infinitely many pairs (i,k), we have $R^{(i)}_k(x) \in L_{n-1}$. For a fixed i, sets $M^{(i)}_k$ are pairwise disjoint, so for given i there can be at most one such pair (i,k). Therefore, $R(x) \in L_n$ iff for infinitely many i there exists k, such that $x \in M^{(i)}_k$. This is equivalent to the fact that for infinitely many i, $x \in M_i$. But the sequence M_i is decreasing, so this is equivalent to the fact that $x \in M$. This shows that R is, indeed, a reduction of M to L_n. ■

4.3 Automata Construction

Theorem 4.7. *For each $n \in \omega$ there is an alternating ωBS-automaton recognizing a $\mathbf{\Pi}^0_{2n+2}$-hard language.*

Proof. For a fixed n, it is possible to construct an alternating ωBS-automaton recognizing exactly the language $W_n(L_n)$. However, to avoid some technical inconveniences, we construct an automaton $\mathcal{A}_n$ for which we only require that it accepts a word $W_n(\eta)$ if and only if $\eta \in L_n$. The latter is sufficient for the proof of hardness.

The automaton will mimic the formula ψ_n (see (4.1)): Player $\forall$ chooses a set X and Player $\exists$ chooses a set Y. The problem that we face is that alternation in automata and quantifier alternation in logic have different semantics. In logic, using the second order quantifier refers to choosing a set all at once, while in automata, players make decisions step by step (position by position). We will be able to overcome this problem using properties of the B-condition.

Automaton. Let us define the automaton $\mathcal{A}_n$ in the following way.

While reading the code of a sequence of vectors, before reading each vector Player $\forall$ decides if he chooses the first component of the vector. If $\forall$ has not chosen the component, $\exists$ can choose it. If the component was chosen by $\forall$, counter a_n counts its length and then resets. If the component was chosen by $\exists$, counter e_n counts its length and then resets.

If the first component was chosen by $\exists$ then the procedure is repeated for the second component and for the counters a_{n-1} and e_{n-1}. We continue with the following components until Player $\exists$ does not choose a component or all components of the vector are selected by $\exists$.

The whole process is repeated for all the vectors in a word.

Player $\forall$ can additionally reset any of a_i counters at any time (except the moment when it is actually incremented). This is to allow Player $\forall$ to select a finite (even empty) set.

The acceptance condition (winning condition for $\exists$ in the game) requires that among the counters $a_n, e_n, a_{n-1}, e_{n-1}, \ldots, a_1, e_1$, the left-most which is unbounded (or reset finitely many times) is an a-counter, or all counters are reset infinitely many times and are bounded.

Soundness. For a given word $w = W_n(\eta)$ such that $\eta \in L_n$, we have to prove that the existential player has a winning strategy in $\mathcal{A}_n$ on w. We proceed by induction. As stated above, $\eta \in L_n$ if and only if there exist infinitely many $m \in \omega$ such that

$$\eta\restriction_{=m} \text{ is infinite} \quad \text{and} \quad \pi_{\bar{1}}\,(\eta\restriction_{=m}) \in L_{n-1} \tag{4.2}$$

Player $\exists$ uses the following strategy. Let k be the greatest value of the first component among vectors selected by Player $\forall$ so far. Let m_k be the least m greater than k, for which condition (4.2) holds. Player $\exists$ selects a vector if its first component is equal to m_k.

Note that we may assume that k is increased only finitely many times during the run (otherwise Player $\forall$ loses). Hence, there exists a value m_{k0} that occurs at the first component of almost all vectors selected by Player $\exists$. By the assumption, $\pi_{\bar{1}}(\eta\restriction_{=m_{k0}}) \in L_{n-1}$, i.e. η restricted to vectors having m_{k0} as the first component, with the first component erased, belongs to L_{n-1}. Player $\exists$ selects almost all vectors with m_{k0} as the first component. Therefore, since L_{n-1}

is prefix-independent, also η restricted to the vectors selected by Player $\exists$, with the first component erased, belongs to L_{n-1}. It follows by inductive assumption that $\exists$ has a strategy on further components of vectors of this restricted sequence.

Induction basis: Since $W_0(L_0)=\{c^\omega\}$, it is straightforward to construct an automaton recognizing it.

Correctness. Now let us take $w = W_n(\eta)$ such that $\eta \notin L_n$. We have to prove that the universal player has a winning strategy in $\mathcal{A}_n$ on w. Note that a strategy of $\forall$ (as well as of $\exists$) in $\mathcal{A}_n$ is simply a selection procedure of vectors or components of vectors. For the induction purposes we strengthen the claim, and prove the following:

Lemma 4.8. *If $w = W_n(\eta)$ such that $\eta \notin L_n$, then Player $\forall$ has a winning strategy σ in $\mathcal{A}_n$ on w such that if ν is a subsequence of η then the strategy σ restricted to positions of ν is winning in $\mathcal{A}_n$ on $W_n(\nu)$.*

Note that if $\eta \notin L_n$, then there exists m_0 such that for all $m \geq m_0$

$$\eta\restriction_{=m} \text{ is finite} \quad \text{or} \quad \pi_{\bar{1}}\left(\eta\restriction_{=m}\right) \notin L_{n-1} \tag{4.3}$$

Player $\forall$ can use the following strategy: Select all the vectors with the first coordinate less than m_0. If there are only finitely many such vectors, $\forall$ uses additional resets. During the game, Player $\forall$ remembers the largest first coordinate M of vectors selected by Player $\exists$.

For every $i \in \{m_0, \ldots, M\}$ we have $\eta\restriction_{=i}$ is finite or $\pi_{\bar{1}}\left(\eta\restriction_{=i}\right) \notin L_{n-1}$, so

$$\overline{\eta_M} := \eta\restriction_{\in\{m_0,m_0+1,\ldots,M\}} \quad \text{is finite} \quad \text{or} \quad \eta_M := \pi_{\bar{1}}\left(\overline{\eta_M}\right) \notin L_{n-1}.$$

The above holds, because of the pigeonhole property of L_{n-1}.

If $\overline{\eta_M}$ is finite, Player $\exists$ will lose the game (if she does not increase M). We can then assume that $\eta_M \notin L_{n-1}$. Then, by the inductive assumption, Player $\forall$ has a winning strategy σ on η_M that satisfies the condition from Lemma 4.8. Player $\forall$ can use a restriction of the strategy σ to the vectors selected by Player $\exists$ to win the game, until Player $\exists$ selects something greater than M.

The value M can increase only finitely many times during the game (otherwise Player $\exists$ loses). Using prefix independence of the winning condition, we obtain that $\forall$ wins the game.

The inductive basis is trivial here, since there is no $\eta \in \mathcal{N}_0 \setminus L_0$. ■

5 Conclusions

The languages presented in Section 2 enable us to give exact estimations of the topological complexity of MSO + U definable sets.

Theorem 5.1. *For every* MSO + U *formula φ over infinite words or trees, the language $L(\varphi)$ is in $\mathbf{\Sigma}^1_i$ for some i. Additionally, for every $i \in \omega$ there is an* MSO + U *definable language that is hard for $\mathbf{\Sigma}^1_i$.*

Proof. Quantifiers $\exists, \forall$ correspond to projection and co-projection. Quantifier U can be interpreted as a countable intersection of countable unions ranging over all finite sets. Therefore, for a given MSO + U formula we can inductively show that $L(\varphi) \in \mathbf{\Sigma}^1_{|\varphi|}$, no matter whether ω-word or infinite tree languages are concerned.

Using Theorem 2.1, we obtain examples of MSO + U languages that are hard in classes at arbitrarily high projective levels. Of course those examples may also be interpreted in infinite binary trees (e.g. on the leftmost branch). ■

Note that the theorem is also true for languages of infinite graphs, digraphs, grids, or any other structures that do not have non-projective predicates and that we can encode ω-words in (in an MSO definable way).

Additionally, the following remarks summarise the topological complexity of WMSO + U.

Remark 5.2 (See [CDFM09])**.** *Every* WMSO + U *definable ω-word language is a boolean combination of $\mathbf{\Sigma}^0_2$ sets.*

Remark 5.3. WMSO + U *over infinite trees defines languages at finite levels of the Borel hierarchy.*

Proof. Weak quantifiers $\exists, \forall$ correspond to countable unions and intersections. Quantifier U, as mentioned in the previous proof, can be expressed as countable intersection of countable unions. Therefore for every WMSO + U formula φ we can show that $L(\varphi) \in \mathbf{\Sigma}^0_{2|\varphi|}$.

On the other hand, over trees even WMSO without the unbounding quantifier is able to define languages at arbitrarily high finite levels of the Borel hierarchy (see [Sku93]). ■

Therefore, this paper gives the last element needed to estimate the topological complexity of U in all four contexts: weak or full MSO logics over words or trees.

In the statement of the following theorem we use term *projective accepting condition*. By this we mean any condition on possible runs (or plays) of the automaton that is in some class $\mathbf{\Sigma}^1_i$. The following example shows that the accepting condition of nondeterministic tree automata is projective.

Example 5.4. *The accepting condition of nondeterministic parity automata on trees is in $\mathbf{\Pi}^1_1$.*

Proof. The condition can be expressed in the following way: a run ρ is accepting iff for every infinite branch of a tree, the $\limsup$ of the ranks of ρ on this branch is even. Since the property „$\limsup$ of the ranks is even" is Borel, the above condition is $\mathbf{\Pi}^1_1$ as a co-projection of a Borel set. ■

Theorem 5.5. *There is no model of alternating (neither nondeterministic nor deterministic) automata with a fixed projective accepting condition that can capture the whole expressive power of* MSO + U *on ω-words.*

Proof. Assume that there is one. Since alternating automata are the most general, we focus on them. The accepting condition is a subset $\mathcal{T} \subseteq C^\omega$, where C denotes the (possibly infinite) set of configurations of an automaton and C^ω is a set of all possible plays in the game induced by the automaton.

In that case $L(\mathcal{A})$ can be written as a set of such words $\alpha \in A^\omega$ on which there exists a strategy σ of Player $\exists$ such that for every possible play τ consistent with this strategy we have $\tau \in \mathcal{T}$. It is easy to observe that properties „σ is a strategy for $\exists$" and „τ is a run consistent with σ" are Borel (in fact closed). Therefore, by the above definition, if $\mathcal{T} \in \mathbf{\Sigma}^1_i$ then $L(\mathcal{A}) \in \mathbf{\Sigma}^1_{i+2}$. But, by Theorem 2.1, there are MSO + U definable languages that are not in $\mathbf{\Sigma}^1_{i+2}$, that yields a contradiction. ■

We have also shown the topological complexity of automata models that were considered in the context of MSO + U logic before. In Theorem 3.2, Corollary 3.3 and Theorem 3.5 we show that nondeterministic ωB-, ωS-, ωBS -automata recognize languages in $\mathbf{\Sigma}^0_3$, $\mathbf{\Pi}^0_3$, $\mathbf{\Sigma}^0_4$, respectively. Additionally in Fact 3.4 and Fact 3.6 we show that there are appropriate automata recognizing languages hard in their respective topological classes. Thanks to this upper and lower complexity bounds we can say that the topological complexity of these automata models is solved.

In Chapter 4 we showed that for each finite level of the Borel hierarchy there is a language recognized by an alternating ωBS -automaton hard for this level. This, in particular, implies:

Corollary 5.6. *Alternating ωBS -automata are more expressive than boolean combinations of nondeterministic ωBS -automata.*

As far as the authors know, it was never observed before the paper [HST10].

6 Further Work

This paper gives exact estimations on the topological complexity of the quantifier U and automata models: ωB, ωS and ωBS. However, there are still some open questions in this subject. The following list contains some of them.

1. What does the Wadge hierarchy (see [Wad83]) look like for the MSO + U definable languages?

2. Is there any MSO + U definable language that is a boolean combination of $\mathbf{\Sigma}^0_2$ sets and is not Wadge-equivalent to any ω-regular language (compare to [CDFM09])?

3. Is there any gap property for MSO + U logic (see [NW03])?

There is a partial and potentially interesting answer for the last question. The conjecture about the gap property for nondeterministic tree languages says:

Conjecture 6.1. *Every regular tree language is either non-Borel or in* $\mathbf{\Sigma}^0_n$ *for some* $n \in \omega$.

It turns out that this is false in the case of MSO + U definable languages.

Example 6.2. *There exists an* MSO + U *definable language of labelled infinite binary trees, that is at an infinite level of the Borel hierarchy.*

Proof. Let L_∞ be the language of infinite binary trees t over the alphabet $\{x, +, -, b\}$, such that: there exists a subtree (prefix-closed subset of nodes) $T \subseteq \{L, R\}^*$ labelled in the following way

- inner nodes of T are labelled by x,
- leaves of T are labelled by $+$ or $-$,
- vertices in $\{L, R\}^* \setminus T$ are labelled by b,

such that

- if $v \in T$ then v is either a leaf of T or
  - if v is a left child then the leftmost infinite branch starting in v is in T,
  - if v is a right child or a root then the rightmost infinite branch starting in v is in T,
- the number of turns (alternations of L and R) on branches of T is bounded (this number is called the *depth* of T),

and that there exists $S \subseteq T$, such that:

- $\varepsilon \in S$,
- for any $v \in S$ that is not a leaf of T, we have
  - if v is a left child then $vL^*R \subseteq S$ and
  - if v is a right child or a root then $vR^nL \in S$ for some $n \in \omega$,
- for any $v \in S$ that is a leaf of T, we have $t(v) = +$.

It is easy to check that all these properties can be expressed in MSO + U. Additionally, for a fixed *depth* of T, the above language is at a finite level of the Borel hierarchy. So $L_\infty \in \mathbf{\Sigma}^0_\omega$. But for any $i < \omega$ we can easily reduce any $\mathbf{\Sigma}^0_i$ language to L_∞ using standard techniques (see e.g. [TL93]). Therefore L_∞ is not finite Borel. ■

There is also one question on the automata side that we leave open. There is a huge gap between the upper and the lower bound for the complexity of alternating ωBS -automata that we provide. On one hand we know that they inhabit at least all finite levels of the Borel hierarchy. On the other hand, using reasoning as in the proof of Theorem 5.5, we obtain that each language recognized by such an automaton is in $\mathbf{\Sigma}^1_2$. The gap is significant, however, the importance of the model has decreased, as we know that it is not sufficient to cover the whole expressive power of MSO + U logic.

Acknowledgements The authors would like to thank the anonymous referees for a number of helpful suggestions that came from very careful lecture and thorough analysis of our results.

References

[BC06] Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS*, pages 285–296, 2006.

[Boj04] Mikołaj Bojańczyk. A bounding quantifier. In *CSL*, pages 41–55, 2004.

[Boj10] Mikołaj Bojańczyk. Beyond ω-regular languages. In *STACS*, pages 11–16, 2010.

[Boj11] Mikołaj Bojańczyk. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.*, 48(3):554–576, 2011.

[BT09] Mikołaj Bojańczyk and Szymon Toruńczyk. Deterministic automata and extensions of weak mso. In *FSTTCS*, pages 73–84, 2009.

[Büc62] J. Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

[CDFM09] Jérémie Cabessa, Jacques Duparc, Alessandro Facchini, and Filip Murlak. The wadge hierarchy of max-regular languages. In *FSTTCS*, pages 121–132, 2009.

[HST10] Szczepan Hummel, Michał Skrzypczak, and Szymon Toruńczyk. On the topological complexity of MSO+U and related automata models. In *MFCS*, pages 429–440, 2010.

[Kec95] Alexander Kechris. *Classical descriptive set theory.* Springer-Verlag, New York, 1995.

[Kur66] Kazimierz Kuratowski. *Topology. Vol. I.* Academic Press, New York, 1966.

[McN66] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

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

[Rab68] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Bull. Amer. Math. Soc.*, 74:1025–1029, 1968.

[Saf88] Samuel Safra. On the complexity of omega-automata. In *FOCS*, pages 319–327, 1988.

[Sku93] Jerzy Skurczyński. The borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

[Sri98] Sashi M. Srivastava. *A Course on Borel Sets*, volume 180 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.

[Tho96] Wolfgang Thomas. Languages, automata and logics. Technical Report 9607, Institut für Informatik und Praktische Mathematik, Christian-Albsechts-Universität, Kiel, Germany, 1996.

[TL93] Wolfgang Thomas and Helmut Lescow. Logical specifications of infinite computations. In *REX School/Symposium*, pages 583–621, 1993.

[Wad83] William Wadge. *Reducibility and determinateness in the Baire space*. PhD thesis, University of California, Berkeley, 1983.