Fundamenta Informaticae XXI (2001) 1001–1013
IOS Press

On the Borel complexity of MSO definable sets of branches*

Mikołaj Bojańczyk, Damian Niwiński
Institute of Informatics
University of Warsaw
Poland
bojan@mimuw.edu.pl, niwinski@mimuw.edu.pl

Alexander Rabinovich
The Blavatnik School of Computer Science
Tel Aviv University
Israel
rabinoa@post.tau.ac.il

Adam Radziwończyk–Syta, Michał Skrzypczak
Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw
Poland
ar237576@students.mimuw.edu.pl, mskrzypczak@mimuw.edu.pl

Abstract. An infinite binary word can be identified with a branch in the full binary tree. We consider sets of branches definable in monadic second-order logic over the tree, where we allow some extra monadic predicates on the nodes. We show that this class equals to the Boolean combinations of sets in the Borel class $\mathbf{\Sigma}^0_2$ over the Cantor discontinuum. Note that the last coincides with the Borel complexity of ω-regular languages.

Keywords: Monadic 2nd-order logic, infinite words, trees, Borel hierarchy, automata

*The authors affiliated at the University of Warsaw have been supported by the Polish government grant no. N206 008 32/0810. D.Niwiński and A.Rabinovich have been supported by the ESF grants AutoMathA and Games.

1. Introduction

It is well known that a great part of automata theory extends quite well from words to trees. But, not surprisingly, the analogous results become often more difficult in the tree case, as trees have a richer structure than words. A celebrated example is decidability of SkS, i.e., the monadic second order (MSO) theory of the full k-ary tree $\mathbf{t_k} = \langle \{1,\dots,k\}^*, succ_1, \dots, succ_k \rangle$, where $succ_i(w) = wi$. Rabin's proof [11] for $k \geq 2$ needed an essentially new insight into the subject, although it built on an idea of reducing formulas to automata, previously used by Büchi in his proof of decidability of S1S. The increase of conceptual difficulty is also reflected by the computational complexity of the related decision problems. For example, the non-emptiness problem for automata with the Rabin acceptance criterion over infinite words is in P, while the analogous problem for trees is NP-complete [4].

A good context where the two kinds of objects can be compared is topology. Indeed both trees and words can be naturally represented as elements of the Cantor discontinuum $\{0,1\}^\omega$. Then the complexity of respective concepts is compared in the frame of the classical hierarchies of set-theoretic topology. For instance, finite-state recognizable sets of infinite words are known to be on the 3rd level of the Borel hierarchy, more precisely they are Boolean combinations of sets in $\mathbf{\Sigma}^0_2$ [7] (see also [15]). In contrast, finite-state automata on infinite trees can recognize some Borel sets on any finite level [13], as well as some non-Borel sets in $\mathbf{\Delta}^1_2$ [10].

In this context, we consider the following question. Since an infinite word α over an alphabet $\{1,\dots,k\}$ can be represented as a branch in a k-ary tree, it is possible to define a language of infinite words by an MSO formula with one free set variable interpreted in the structure $\mathbf{t_k}$ as the set of prefixes of α. It is easy to see that a language definable that way must be ω-regular, i.e., recognizable by a Büchi automaton. This of course need not be the case if we extend the tree structure $\mathbf{t_k}$ by some additional monadic predicates. Recently, Bárány, Kaiser, and Rabinovich [1] considered languages definable in that way in context of an uncountability quantifier over trees, and discovered that they are always Borel. In the present paper we show that these languages of infinite words have the same Borel complexity as ω-regular languages, that is, they are in the class $Boole\left(\mathbf{\Sigma}^0_2\right)$ of the Boolean combinations of sets in $\mathbf{\Sigma}^0_2$. Moreover, if we range over all possible predicates, the languages in consideration exhaust the whole class $Boole\left(\mathbf{\Sigma}^0_2\right)$.

To this end, we observe that our languages can be captured by non-deterministic automata with the Büchi acceptance condition, additionally equipped with an *advice* telling which transitions are recommended after reading a finite prefix of an infinite word. We note that a similar concept of automata recognizing languages of *finite* words has been recently considered by Fratani [5] who showed an analogous characterization for languages of finite words definable in tree structures. A useful property is the determinization result which, for infinite words, is analogous to the McNaughton Theorem [9] for the ordinary Büchi automata: the automata with advice can be made deterministic if we replace Büchi condition by some more general acceptance criteria, like the parity acceptance condition (see also [15]).

To complete the proof we note that the languages recognized by automata with advice are closed under continuous reductions. As they also form a Boolean algebra and contain $\mathbf{\Sigma}^0_2$-complete sets (which is well-known already for the ordinary ω-regular languages), we obtain the desired characterization.

Finally we note that the MSO definability of sets of infinite words in a k-ary tree with predicates cannot be reduced to definability in the structure $\langle \omega, succ \rangle$ (i.e., the underlying structure of S1S) with additional predicates; we exhibit a language definable in the former sense but not in the latter.

In this paper, we usually present our proofs for binary trees; an extension to k-ary trees, for $k \geq 2$, is routine. We note that another approach to the MSO definability is possible which, instead of tree automata, uses the Composition Theorem by Lifsches and Shelah [8]. This approach has capacity to extend to more general structures, like trees with infinite branching, in order to estimate the Borel complexity of sets of paths there. It will be the subject of further research.

2. Borel complexity of Büchi automata with advice

In this section, we consider an extension of non-deterministic Büchi automata on infinite words by the concept of advice, and show that the topological complexity of the recognized languages is the same as for ordinary Büchi automata.

Topological preliminaries Throughout the paper, ω denotes the set of natural numbers which we identify with the first infinite ordinal. (Thus the writings $n < \omega$ and $n \in \omega$ are equivalent.) For a set X, X^* denotes the set of finite words over X, including the empty word ε, and X^ω the set of infinite words, i.e., mappings $\omega \to X$. When applied to words, the symbol $\leq$ denotes prefix ordering. The length of a finite word w is denoted by $|w|$. The m-th letter of a word $u \in X^\omega$ is denoted $u(m)$ or u_m interchangeably. The prefix of length m of a word u will be denoted $u \restriction m$, that is

$$u \restriction m = u(0)u(1)\dots u(m-1) = u_0 u_1 \dots u_{m-1},$$

(in particular, $u \restriction 0 = \varepsilon$). determined by the context. We consider X^ω with a topology induced by the metric given by the distance function

$$d(u, u') = \begin{cases} 0 & \text{if } u = u' \\ 2^{-n} \text{ with } n = \min\{i : u(i) \neq u'(i)\} & \text{otherwise.} \end{cases} \tag{1}$$

Note that the open sets are of the form WX^ω, for some set of finite words $W \subseteq X^*$. It is easy to see that if X is *finite* and contains at least two elements then X^ω is homeomorphic with the *Cantor discontinuum* $\{0,1\}^\omega$. (For the concepts of set-theoretic topology, see, e.g., [6].)

We use the notation $\mathbf{\Sigma}^0_n$ and $\mathbf{\Pi}^0_n$, with $1 \leq n < \omega$, for *finite* levels of the Borel hierarchy over $\{0,1\}^\omega$. That is, $\mathbf{\Sigma}^0_1$ and $\mathbf{\Pi}^0_1$ are classes of open and closed sets, respectively. Next, $\mathbf{\Sigma}^0_{n+1}$ consists of *countable* unions of sets in $\mathbf{\Pi}^0_n$, and $\mathbf{\Pi}^0_{n+1}$ consists of countable intersections of sets in $\mathbf{\Sigma}^0_n$. Note that the sets in $\mathbf{\Pi}^0_n$ are complements of the sets in $\mathbf{\Sigma}^0_n$.

2.1. Advised automata

A Büchi automaton on infinite words over an input alphabet A can be presented by $\mathcal{B} = \langle A, Q, q_I, F, Tr \rangle$, where Q is a finite set of *states* with an *initial state* q_I and a subset of *accepting states* $F \subseteq Q$, and $Tr \subseteq Q \times A \times Q$ is a set of (non-deterministic) *transitions*. We write $q \xrightarrow{a} p$ to mean $\langle q, a, p \rangle \in Tr$. A *run* of $\mathcal{B}$ on a word $u \in A^\omega$ is a word $r \in Q^\omega$ such that $r_0 = q_I$, and, for $m < \omega$, $r_m \xrightarrow{u_m} r_{m+1}$. It is accepting if $r_m \in F$, for infinitely many values of m. The language $L(\mathcal{B})$ recognized by $\mathcal{B}$ consists of those words $u \in A^\omega$ which have an accepting run. Languages of infinite words recognized by Büchi automata are called *ω-regular*.

We now generalize the above concept of automata, so that the transition relation will depend on the prefix of a word read so far. We note that a similar concept of automata running on finite words has been considered by Séverine Fratani [5] (called *automates à oracles* there) in the context of automata with nested pushdown stores.

A non-deterministic Büchi *automaton with advice* (or *advised automaton*) can be presented by $\mathcal{B} = \langle A, Q, q_I, F, \rho \rangle$, where Q, q_I, and F are as above, and $\rho : A^* \to \wp(Q \times A \times Q)$ is the advice function which associates a set of transitions with each finite word over A. We write $v, q \xrightarrow{a} p$ to mean $\langle q, a, p \rangle \in \rho(v)$. A *run* of $\mathcal{B}$ on a word $u = u_0 u_1 \ldots \in A^\omega$ is a word $r \in Q^\omega$ such that $r_0 = q_I$, and, for $m \in \omega$,

$$u_0 \ldots u_{m-1}, r_m \xrightarrow{u_m} r_{m+1}.$$

The concept of acceptance is defined similarly as in the previous case. An ordinary Büchi automaton as presented above can be viewed as an automaton with advice defined by

$$\rho(w) = Tr, \quad \text{for } w \in A^*.$$

Parity automata An ordinary (non-deterministic) *parity automaton*[1] differs from a Büchi automaton only by the acceptance condition which, instead of F, takes form of a *ranking* function $rank : Q \to \omega$. A run r is considered accepting if the highest rank occurring infinitely often is *even*, in other words, $\limsup_{n\to\infty} rank(r_n)$ is even. Note that a Büchi automaton can be viewed as a parity automaton with $rank(q) = 2$, for $q \in F$, and $rank(q) = 1$ otherwise.

A *parity automaton with advice* is defined analogously to the Büchi automaton, with the acceptance given in terms of the ranking function.

It is well known that non-deterministic parity automata accept only ω-regular languages. We note that a straightforward transformation from parity to Büchi automata applies also to automata with advice.

Lemma 2.1. For any parity automaton with advice, there exists a Büchi automaton with advice accepting the same language.

Proof:
Let $\mathcal{B} = \langle A, Q, q_I, rank, \rho \rangle$, and suppose that $rank$ takes the values in $\{0, 1, \ldots, m\}$. We construct a Büchi automaton $\mathcal{B}'$ with the set of states

$$Q \cup \bigcup_{2i \le m} \{q : rank(q) \le 2i\} \times \{i\}.$$

The initial state remains q_I, and the accepting states are $F = \{(q, i) : rank(q) = 2i\}$. The advise of $\mathcal{B}'$ is given by the following rules:

- $v, p \xrightarrow{a} q$, whenever it was the case in $\mathcal{B}$,
- $v, p \xrightarrow{a} (q, i)$, whenever $v, p \xrightarrow{a} q$ in $\mathcal{B}$, and $rank(q) \le 2i$,
- $v, (p, i) \xrightarrow{a} (q, i)$, whenever $v, p \xrightarrow{a} q$ in $\mathcal{B}$, and $rank(q) \le 2i$.

[1] Currently most frequently used in the literature, the parity acceptance criterion is well-known to be equivalent to the historically previous Muller and Rabin criteria, see [15].

Intuitively, in some moment of the computation, the automaton "decides" that the highest rank to occur infinitely often should be $2i$. Since that moment on, the automaton cannot enter the states with higher rank, and it accepts if the rank $2i$ occurs infinitely often. The equivalence of $\mathcal{B}$ and $\mathcal{B}'$ follows readily from the definition. $\square$

Determinization An ordinary Büchi automaton is *deterministic* if Tr represents a function from $Q \times A$ to Q; that is, for each q and a, there is exactly one p, such that $q \xrightarrow{a} p$. It is easy to see that Büchi automata cannot, in general, be determinized, but from the celebrated McNaughton Theorem [9], we know that any Büchi automaton is equivalent to a deterministic automaton with parity condition; an elegant and optimal construction has been provided by Safra [12] (see also [15]).

By analogy, an advised automaton is deterministic if each $\rho(v)$ is a function from $Q \times A$ to Q; consequently, for each q, a, and v, there is exactly one p, such that $v, q \xrightarrow{a} p$. As for the ordinary automata, this guarantees that the automaton has exactly one run on each word $u \in A^\omega$. In particular, for each $v \in A^*$, there is exactly one state, say $\rho'(v)$, which the automaton reaches after reading v, starting from the initial state. This leads to a simpler presentation of deterministic automata: instead of $\rho : A^* \to \wp(Q \times A \times Q)$, we can consider the function $\rho' : A^* \to Q$ defined above that we call *state-advice*. Indeed, the function ρ' fully determines the language recognized by the automaton, as a word u is accepted if and only if the sequence $\rho'(u \restriction n)$, $n < \omega$, forms an accepting run. On the other hand, *any* function $f : A^* \to Q$ is a state-advice of some automaton, it is enough to let

$$v, f(v) \xrightarrow{a} f(va)$$

(transitions for $q \neq f(v)$ may be defined arbitrarily). Since now on, we usually present deterministic automata by state-advices.

We now show that the determinization result carries over to automata with advice. A similar results for languages of finite words has been shown by Fratani [5].

Proposition 2.1. For any advised Büchi automaton, there is a deterministic advised parity automaton accepting the same language.

Proof:
Let $\mathcal{B} = \langle A, Q, q_I, F, \rho \rangle$ be a non-deterministic Büchi automaton with advice. We say that an infinite word $\alpha \in \wp(Q \times A \times Q)^\omega$ *favours* an infinite word $u = u_0 u_1 \ldots \in A^\omega$, whenever there exists a sequence of states $q_0, q_1, \ldots$, such that

1. $q_0 = q_I$,
2. $\langle q_n, u_n, q_{n+1} \rangle \in \alpha_n$, for all $n < \omega$,
3. $q_n \in F$, for infinitely many n's.

For any $u \in A^\omega$, let $\rho(u)$ be an infinite word over the alphabet $\wp(Q \times A \times Q)$, defined by

$$\rho(u)(n) \quad = \quad \rho(u \restriction n).$$

Note that u is accepted by $\mathcal{B}$ if and only if $\rho(u)$ favours u.

For any $u \in A^\omega$ and $\alpha \in \wp(Q \times A \times Q)^\omega$, let $u \star \alpha$ be the word over the product alphabet $A \times \wp(Q \times A \times Q)$, defined by

$$u \star \alpha(n) \quad = \quad \langle u(n), \alpha(n) \rangle.$$

The crucial property is that the set

$$\{u \star \alpha : \alpha \text{ favours } u\}$$

is ω-regular in the usual sense. Indeed, a suitable non-deterministic Büchi automaton (over the alphabet $A \times \wp(Q \times A \times Q)^\omega$) can borrow Q, F, and q_I from $\mathcal{B}$, and assume the transitions

$$q \stackrel{\langle a, R \rangle}{\longrightarrow} p,$$

whenever $\langle q, a, p \rangle \in R$. By the McNaughton Theorem, there is an equivalent deterministic automaton with parity condition, say $\mathcal{M}$. We are ready to define a deterministic advised parity automaton recognizing $L(\mathcal{B})$. Its set of states and ranking function are the same as in $\mathcal{M}$. The state-advice sends each finite word $u_0 u_1 \dots u_n$ on the unique state that the automaton $\mathcal{M}$ reaches after reading the word

$$\langle u_0, \rho(\varepsilon) \rangle, \langle u_1, \rho(u_0) \rangle, \langle u_2, \rho(u_0 u_1) \rangle, \dots, \langle u_n, \rho(u_0 u_1 \dots u_{n-1}) \rangle$$

(the empty word ε is sent on the initial state of $\mathcal{M}$). Hence the run this automaton assumes on an infinite word $u \in A^\omega$ coincides with the run of the automaton $\mathcal{M}$ on $u \star \rho(u)$. But $\mathcal{M}$ accepts $u \star \rho(u)$ if and only if $\mathcal{B}$ accepts u. $\square$

Note that, by the above proof, the increase of the number of states induced by determinization is the same as in the classical construction.

Borel complexity We first note that automata with advice are not more powerful than ordinary automata as far as the Borel complexity is concerned. Indeed, let $\mathcal{B}$ be a deterministic parity automaton with a state-advice $\rho : A^* \to Q$, and a ranking function $rank : Q \to \{0, 1, \dots, m\}$. Let us abbreviate $\underline{m} = \{0, 1, \dots, m\}$. We can simplify the automaton further, by taking $\underline{m}$ as the set of states with $rank(i) = i$, and the state-advice given by $rank \circ \rho$; clearly the new automaton is equivalent to the previous one. This further induces a continuous (even Lipschitz) mapping from A^ω to $\underline{m}^\omega$

$$u \mapsto rank(\rho(u \restriction 0)),\ rank(\rho(u \restriction 1)),\ rank(\rho(u \restriction 2)),\ \dots$$

Clearly the set $L(\mathcal{B})$ is an inverse image under this mapping of the set of strings satisfying the parity criterion

$$Parity_m \quad = \quad \{\alpha \in \underline{m}^\omega : \limsup_{n \to \infty} \alpha_n \text{ is even }\}.$$

The last set is a Boolean combination of sets defined by the conditions "i occurs only finitely often", and hence is in the Boolean closure of the Borel class $\mathbf{\Sigma}^0_2$. (This also follows from the Landweber bound on the ω-regular languages [7].) Hence, any set of infinite words recognized by a non-deterministic parity automaton with advice has at most this Borel complexity.

It turns out that the converse is also true.

Theorem 2.1. A language $L \subseteq A^\omega$ is presentable as a Boolean combination of sets in $\mathbf{\Sigma}^0_2$ if and only if it is recognized by a deterministic parity automaton with advice, and consequently also by a (non-deterministic) Büchi automaton with advice.

Proof:
The *if* implication has been observed above. To show the *only if* part, we will use deterministic automata with the states coinciding with their ranks (see page 1006). A state-advice of the form $\rho : A^* \to \underline{m}$ may be viewed as a coloring of the tree A^* by the ranks in $\underline{m}$. Therefore, for the sake of this proof, we call the set recognized an automaton a *rainbow*. The strategy of the proof is to show that rainbows comprise the whole class $\mathbf{\Sigma}^0_2$ and are closed under Boolean operations.

We first show that each continuous reduction induces a rainbow.

Lemma 2.2. Let $f : A^\omega \to \underline{m}^\omega$ be a continuous function and $K = f^{-1}(Parity_m)$. Then K is a rainbow.

Proof:
For $w \in A^*$, let $\hat{f}(w)$ be the largest common prefix of the words in $\{f(wu) : u \in A^\omega\}$. Note that it can be finite or infinite (if the prefix w determines the value of f). It follows from continuity of f that, for any $u \in A^\omega$, the sequence of lengths $|\hat{f}(u \restriction n)|$ diverges to infinity (it may also reach it, for some n). Hence there is a unique infinite word having all $\hat{f}(u \restriction n)$'s as prefixes, which must be $f(u)$.

To define a state-advice ρ for an automaton recognizing K, we proceed by induction on the length of an argument w. Let $\rho(\varepsilon) = 0$. For $w > \varepsilon$, we consider two cases. If $\hat{f}(w)$ is an infinite word, we let

$$\rho(w) \quad = \quad \limsup_{n\to\infty} \hat{f}(w)(n).$$

Otherwise let $w = w'a$, with $a \in A$. Clearly $\hat{f}(w) = \hat{f}(w')\Delta$, for some $\Delta \in \underline{m}^*$. If $\Delta = \varepsilon$, we let $\rho(w) = \rho(w')$. Otherwise, if $\Delta = \delta_1 \dots \delta_k$, for some $k \geq 1$, we let

$$\rho(w) \quad = \quad \max\{\delta_1, \dots, \delta_k\}.$$

Now it is enough to show that, for each $u \in A^\omega$,

$$\limsup_{n\to\infty} \rho(u \restriction n) \quad = \quad \limsup_{n\to\infty} f(u)(n).$$

If, for some n, $\hat{f}(u \restriction n)$ is infinite then it must equal $f(u)$. Then the sequence on the left-hand side stabilizes on the value that equals precisely to the right-hand side. Otherwise, $f(u)$ can be decomposed

$$f(u) \quad = \quad \Delta_0 \Delta_1 \dots$$

where $\hat{f}(u \restriction n+1) = \hat{f}(u \restriction n)\Delta_n$. Then the claim follows from a simple observation that if in a sequence $\alpha \in \underline{m}^\omega$ we replace any number of (pairwise disjoint) subwords by their maxima, the $\limsup$ remains the same. $\square$

It follows from the above lemma that rainbows are closed under continuous reductions, i.e., if $f : A^\omega \to A^\omega$ is a continuous mapping and $K \subseteq A^\omega$ a rainbow then $f^{-1}(K)$ is also a rainbow. Indeed, if $\rho : A^* \to \underline{m}$ is an advice recognizing K then the mapping

$$u \mapsto \rho(f(u) \restriction 0),\ \rho(f(u) \restriction 1),\ \dots$$

is a continuous reduction of $f^{-1}(K)$ to $Parity_m$, hence $f^{-1}(K)$ is a rainbow by Lemma 2.2. Hence, to show that rainbows comprise the whole class $\mathbf{\Sigma}^0_2$, it is enough to exhibit a rainbow complete in this class (w.r.t. continuous reductions). It is well known that there are (ordinary) ω-regular languages complete in $\mathbf{\Sigma}^0_2$. For concreteness, suppose that A contains letters $0, 1$, and consider the set $Parity_1$ which is then included in A^ω. It is straightforward to see that $Parity_1$ cannot be recognized by a deterministic Büchi automaton and hence, by Landweber's characterization [7] (see also [14], Theorem 5.3) belongs to $\mathbf{\Sigma}^0_2 - \mathbf{\Pi}^0_2$. By the result of Wadge (see [6], Theorem 22.10), this implies that $Parity_1$ is complete in $\mathbf{\Sigma}^0_2$. (A direct proof of this fact is also not difficult.)

To conclude the proof of the theorem, it is enough to show that rainbows form a Boolean algebra. It is easy to see that if $\rho : A^* \to \underline{m}$ is an advice for an automaton recognizing K then the formula $\tilde{\rho}(w) = \rho(w) + 1$ gives an advice $\rho : A^* \to \underline{m}$ recognizing the complement of K. Next it suffices to show that rainbows are closed under binary union. If K_1 and K_2 are rainbows, it is straightforward to construct a non-deterministic automaton with advice recognizing $K_1 \cup K_2$. By Proposition 2.1, it can be determinized, hence $K_1 \cup K_2$ is a rainbow. This remark completes the proof. □

3. Defining words in trees

In this section, we show that a set of infinite words is MSO definable in a k-ary tree if and only if it is recognizable by a parity automaton with advice. Together with Theorem 2.1, this yields the desired topological characterization.

We restrict our considerations to binary trees; extension of the results to k-ary trees, for $k \geq 2$, is routine. (For $k = 1$ the result is trivial.)

Monadic second-order logic A (relational) *signature* is a finite set τ of relation symbols; each R in τ given with a (finite) *arity* $ar(R) \geq 1$. The formulas of *monadic second order (MSO) logic* over signature τ use two kinds of variables : *individual variables* $x_0, x_1, \ldots$, and *set variables* $X_0, X_1, \ldots$. Atomic formulas are $x_i = x_j$, $R(x_{i_1}, \ldots, x_{i_{ar(R)}})$, and $X_i(x_j)$. The other formulas are built using propositional connectives $\vee, \neg$, and the quantifier $\exists$ ranging over both kinds of variables.

Formulas are interpreted in relational structures over the signature τ, which we present by $\mathbf{A} = \langle A, \{R^{\mathbf{A}} : R \in \tau\}\rangle$, where A is the *universe* of $\mathbf{A}$, and $R^{\mathbf{A}} \subseteq A^{ar(R)}$ is an $ar(R)$-ary relation on A. A *valuation* is a mapping v from the set of variables (of both kinds), such that $v(x_i) \in A$, and $v(X_i) \subseteq A$. The *satisfaction relation* of a formula φ in a structure $\mathbf{A}$ under the valuation v is defined by induction on φ in the usual manner and denoted $\mathbf{A}, v \models \varphi$ (see, e.g., [3]).

A variable (of any kind) is *free* in φ if it has an occurrence not bound by a quantifier. We write $\varphi(\xi_1, \ldots, \xi_k)$ to indicate that the free variables of φ are among $\xi_1, \ldots, \xi_k$. Clearly, the satisfaction of a formula depends only on the valuation of its free variables. We write $\mathbf{A} \models \varphi[\alpha_1, \ldots, \alpha_k]$ to mean that $\mathbf{A}, v \models \varphi$, for a valuation v, such that $v(\xi_i) = \alpha_i$, for $i = 1, \ldots, k$.

A (binary) *tree with predicates* is a structure with the universe $\{1, 2\}^*$, over the signature consisting of binary symbols $succ_1, succ_2$, and unary symbols $P_1, \ldots, P_m$, for some $m < \omega$. It can be presented

$$\mathbf{t} = \langle \{1, 2\}^*,\, P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}},\, succ_1^{\mathbf{t}}, succ_2^{\mathbf{t}} \rangle.$$

We further assume that the symbols $succ_i$ are interpreted as the *successor* relations $succ_i^{\mathbf{t}} = \{(w, wi) : w \in \{1,2\}^*\}$, whereas the symbols P_i are interpreted as arbitrary sets $P_i^{\mathbf{t}} \subseteq \{1,2\}^*$, which we usually call *predicates*.

We refer to finite words over the alphabet $\{1,2\}$ as to *nodes* of the tree, with the empty word ε coinciding with the root. An infinite word $u \in \{1,2\}^\omega$ can be viewed as a path in the tree. As far as MSO definability is concerned, it is convenient to identify it with the set of nodes

$$\hat{u} = \{u \restriction n : n \in \omega\}.$$

Definition 3.1. *A set $L \subseteq \{1,2\}^\omega$ is* MSO definable in $\mathbf{t}$*, if there exists an MSO formula $\varphi(X)$, such that, for any set $Z \subseteq \{1,2\}^*$,*

$$\mathbf{t} \models \varphi[Z] \text{ iff } Z = \hat{u}, \text{ for some } u \in L.$$

Automata on trees A non-deterministic (binary) tree automaton with a parity acceptance condition is presented by $\mathcal{D} = \langle A, Q, q_I, Tr, rank\rangle$, where A is a finite alphabet of input symbols, Q is a finite set of states with an initial state q_I, $Tr \subseteq Q \times A \times Q \times Q$ is a set of transitions, and $rank : Q \to \omega$ is the ranking function. A transition (q, a, p_1, p_2) is usually written $q \stackrel{a}{\to} p_1, p_2$.

An input to an automaton is an infinite (binary) A–valued tree, which can be presented as mapping $t : \{1,2\}^* \to A$. We let T_A denote the set of all such trees. A *run* of $\mathcal{D}$ on a tree $t \in T_A$ is itself a Q–valued tree $r : \{1,2\}^* \to Q$ such that $r(\varepsilon) = q_I$, and, for each $w \in dom\,(r)$, $r(w) \stackrel{t(w)}{\to} r(w1), r(w2)$ is a transition in Tr. A *path* $P = p_0p_1 \ldots \in \{1,2\}^\omega$ in r is *accepting* if $\limsup_{n\to\infty} rank(r(p_0p_1 \ldots p_n))$ is even.

A *run is accepting* if so are all its paths. The tree language $L(\mathcal{D})$ *recognized* by $\mathcal{D}$ consists of those trees in T_A which admit an accepting run.

The correspondence between MSO formulas and automata constitutes a key step in Rabin's proof of decidability of S2S ([11], see also [14]). For a set $Z \subseteq \{1,2\}^*$, a characteristic mapping $\chi_Z : \{1,2\}^* \to \{0,1\}$ is given by $\chi_Z(v) = 1$ if $v \in Z$, and $\chi_Z(v) = 0$, otherwise. For a vector of sets $Z_1, \ldots, Z_k \subseteq \{1,2\}^*$, a *characteristic tree* $t_{\vec{Z}} : \{1,2\}^* \to \{0,1\}^k$ is given by

$$t_{\vec{Z}}(v) \quad = \quad \langle \chi_{Z_1}(v), \ldots, \chi_{Z_k}(v)\rangle.$$

Rabin proved [11] that, for an MSO formula φ without predicate symbols and with the free variables among $X_1, \ldots, X_k$, one can always construct an automaton $\mathcal{D}_\varphi$ over the input alphabet $\{0,1\}^k$, such that, for all $Z_1, \ldots, Z_k \subseteq \{1,2\}^*$,

$$\mathbf{t_2} \models \varphi[Z_1, \ldots, Z_k] \quad \text{iff} \quad t_{\vec{Z}} \in L(\mathcal{D}_\varphi), \tag{2}$$

where $\mathbf{t_2}$ is the full binary tree without predicates (see also [14]).

Now, let us replace some variables in φ, say $X_1, \ldots, X_m$ ($m \leq k$), by the monadic relation symbols $P_1, \ldots, P_m$, thus obtaining a new formula φ' over an extended signature. Then, for a tree $\mathbf{t}$, where the new symbols are interpreted by predicates $P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}}$, we have

$$\mathbf{t} \models \varphi'[Z_{m+1}, \ldots, Z_k] \quad \text{iff} \quad \mathbf{t_2} \models \varphi[P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}}, Z_{m+1}, \ldots, Z_k]. \tag{3}$$

The equivalences (2) and (3) allow us to rephrase Definition 3.1 in terms of automata. Namely, a set $L \subseteq \{1,2\}^\omega$ is MSO definable in a tree $\mathbf{t}$ (with predicates $P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}}$) iff there exists a tree automaton $\mathcal{D}$ over the alphabet $\{0,1\}^{m+1}$, such that, for any set $Z \subseteq \{1,2\}^*$,

$$t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},Z} \in L(\mathcal{D}) \quad \text{iff} \quad Z = \hat{u} \text{ for some } u \in L. \tag{4}$$

This last characterization is useful to prove the following characterization. We note that a similar results for languages of finite words has been shown by Fratani (see chapter 4 in [5]).

Proposition 3.1. A set $L \subseteq \{1,2\}^\omega$ is MSO definable in a tree with predicates if and only if it is recognized by a parity automaton with advice.

Proof:
Only if. Suppose L is definable in a tree $\mathbf{t} = \langle \{1,2\}^*,\ P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}},\ \mathit{succ}\,_1^{\mathbf{t}}, \mathit{succ}\,_2^{\mathbf{t}}, \rangle$, and let an automaton $\mathcal{D} = \langle A, Q, q_I, \mathit{Tr}, \mathit{rank} \rangle$ witness this definability in the sense of (4). The automaton $\mathcal{B}$ recognizing L will have the same set of states as $\mathcal{D}$, the same initial state and the rank function. The advice function will depend on the values of the predicates $P_i^{\mathbf{t}}$. At first, for each node $v \in \{1,2\}^*$, we fix the set of states from which the automaton $\mathcal{D}$ would accept the subtree of $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$ rooted in v, provided that the path $\hat{u}$ did not enter this subtree. More specifically, let $t_v^{\emptyset} : \{1,2\}^* \to \{0,1\}^k$ be a tree defined by

$$t_v^{\emptyset}(w) \quad = \quad \langle P_1^{\mathbf{t}}(vw), \ldots, P_m^{\mathbf{t}}(vw), 0 \rangle,$$

where $P_i^{\mathbf{t}}(x)$ equals 1 if $x \in P_i^{\mathbf{t}}$, and 0 otherwise. Let $\mathcal{D}_q$, with $q \in Q$, be an automaton which coincides with $\mathcal{D}$, except for that its initial state is q. We let

$$\mathit{acc}\,(v) \quad = \quad \{q : t_v^{\emptyset} \in L(\mathcal{D}_q)\}.$$

The advice function ρ of the automaton $\mathcal{B}$ is defined by the following rule:

- $v, p \xrightarrow{1} q$, whenever the automaton $\mathcal{D}$ has a transition $p \xrightarrow{\langle P_1^{\mathbf{t}}(v),\ldots,P_m^{\mathbf{t}}(v),1\rangle} q, q'$, for some $q' \in \mathit{acc}\,(v2)$,
- $v, p \xrightarrow{2} q$, whenever the automaton $\mathcal{D}$ has a transition $p \xrightarrow{\langle P_1^{\mathbf{t}}(v),\ldots,P_m^{\mathbf{t}}(v),1\rangle} q'', q$, for some $q'' \in \mathit{acc}\,(v1)$.

Intuitively, for an input u, the automaton $\mathcal{B}$ follows a hypothetical run of $\mathcal{D}$ on the characteristic tree $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$, along the path $\hat{u}$. Note that the input letters for the automaton $\mathcal{B}$ correspond to directions in the tree (not to labels). For a transition $p \to q, q'$ of $\mathcal{D}$, the automaton $\mathcal{B}$ "chooses" one direction: left or right, depending on its actual input letter: 1 or 2, respectively. The advice makes sure that the run corresponds indeed to an accepting run of $\mathcal{D}$.

We now show that $\mathcal{B}$ accepts an infinite word u if and only if $\mathcal{D}$ accepts the tree $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$. Let r be an accepting run of $\mathcal{D}$ on this tree. Consider the sequence of states

$$r(\varepsilon), r(u_0), r(u_0u_1), r(u_0u_1u_2), \ldots$$

It follows directly from the definitions that this is an accepting run of $\mathcal{B}$ on u.

Conversely, let $s = s_0 s_1 s_2 \ldots$ be an accepting run of $\mathcal{B}$ on a word u. By assumption, $s_0 = q_I$ and $u \upharpoonright n, s_n \stackrel{u_n}{\to} s_{n+1}$, for $n < \omega$. We construct a run r of $\mathcal{D}$ on $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$, as follows. We first let $r(u \upharpoonright n) = s_n$, for $n < \omega$. That is, the states assumed along the path u are the same as in the run s. Note that, whenever $u_n = 1$, there is a transition

$$s_n \stackrel{\langle P_1^{\mathbf{t}}(u \upharpoonright n),\ldots,P_m^{\mathbf{t}}(u \upharpoonright n),1\rangle}{\longrightarrow} s_{n+1}, q,$$

for some $q \in acc\ ((u \upharpoonright n)2)$. Hence, we can define an accepting run starting from q on the subtree of $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$ rooted in $(u \upharpoonright n)2$, which coincides with the tree $t^{\emptyset}_{(u \upharpoonright n)2}$. Similarly, if $u_n = 2$ then we can extend the run on the subtree rooted in $(u \upharpoonright n)1$. Thus we obtain an accepting run of $\mathcal{D}$ on $t_{P_1^{\mathbf{t}},\ldots,P_m^{\mathbf{t}},\hat{u}}$, as desired.

If. By Proposition 2.1 and the subsequent considerations, we may assume that L is recognized by a deterministic automaton with a state-advice $\rho : \{1,2\}^* \to \underline{m}$, for some m. Consider a tree $\mathbf{t}$ with predicates $P_1^{\mathbf{t}}, \ldots, P_m^{\mathbf{t}}$, defined by

$$v \in P_i^{\mathbf{t}} \quad \text{iff} \quad \rho(v) = i.$$

Clearly, $u \in L$ if and only if the highest i, such that $P_i^{\mathbf{t}}(u \upharpoonright n)$ holds for infinitely many i's, is even. This last property is readily expressible by an MSO formula over $\mathbf{t}$. □

Remark Note that, in the proof of the implication *If* of the above proposition, the model $\mathbf{t}$ depends on the advice ρ, but the actual MSO formula depends only on m. Hence, we have in fact a sequence of formulas φ_m (expressing the parity condition), such that each MSO definable set of infinite words is definable by some φ_m.

By combining Proposition 3.1 with Theorem 2.1, we obtain the following.

Corollary 3.1. A set $L \subseteq \{1,2\}^{\omega}$ is MSO definable in a tree with predicates if and only if it is presentable as a Boolean combination of sets in $\mathbf{\Sigma}_2^0$ w.r.t. the Cantor topology on $\{1,2\}^{\omega}$.

As we have mentioned above, the extension of this result to the alphabet $\{1, 2, \ldots, k\}$, for any $k < \omega$, is completely routine.

We complete our considerations by an observation that definability in binary trees with extra predicates in nevertheless more powerful than definability in ω with extra predicates, in the following sense.

Consider the structure

$$\mathbf{N} = \langle \omega, P_1^{\mathbf{N}}, \ldots, P_m^{\mathbf{N}}, \ succ^{\mathbf{N}} \rangle,$$

where $succ^{\mathbf{N}} = \{(n, n+1) : n < \omega\}$, and $P_i^{\mathbf{N}} \subseteq \omega$, for $i = 1, \ldots, m$, are arbitrary monadic predicates over ω. We are now interested in definability of languages of infinite words in this structure in the usual sense, i.e., by viewing words as characteristic functions of tuples of sets. More specifically, for a vector of sets $Z_1, \ldots, Z_k \subseteq \omega$, its *characteristic word* is an infinite word $u_{\vec{Z}} : \omega \to \{0,1\}^k$, defined by

$$u_{\vec{Z}}(n) \quad = \quad \langle Z_1(n), \ldots, Z_k(n) \rangle,$$

where $Z_i(n) = 1$ if $n \in Z_i$, and $Z_i(n) = 0$ otherwise.

Definition 3.2. *A language* $L \subseteq \left(\{0,1\}^k\right)^\omega$ *is* MSO definable in $\mathbf{N}$ *if there exists an MSO formula* $\varphi(X_1, \ldots, X_k))$*, such that, for any sets* $Z_1, \ldots, Z_k \subseteq \omega$*,*

$$\mathbf{N} \models \varphi[Z_1, \ldots, Z_k] \quad \text{iff} \quad u_{\vec{Z}} \in L.$$

Let $\underline{\omega}$ denote the structure $\mathbf{N}$ without any predicates. We use the correspondence between MSO formulas over $\underline{\omega}$ and Büchi automata analogous to (2), originally established by Büchi [2] in his proof of decidability of S1S (see also [14]). We then have the following analogue to the equivalence (4) above. For a language $L \subseteq \left(\{0,1\}^k\right)^\omega$ definable by a formula $\varphi(X_1, \ldots, X_k)$ interpreted in a structure $\mathbf{N}$ with predicates $P_1^{\mathbf{N}}, \ldots, P_m^{\mathbf{N}}$, we can find a non-deterministic Büchi (or deterministic parity) automaton $\mathcal{B}$ over the alphabet $\{0,1\}^{m+k}$, such that, for any $Z_1, \ldots, Z_k \subseteq \omega$,

$$u_{P_1^{\mathbf{N}}, \ldots, P_m^{\mathbf{N}}, Z_1, \ldots, Z_k} \in L(\mathcal{B}) \quad \text{iff} \quad u_{\vec{Z}} \in L. \tag{5}$$

Note that the topological complexity of L is not higher than that of $L(\mathcal{B})$, as the mapping $u_{\vec{Z}} \mapsto u_{P_1^{\mathbf{N}}, \ldots, P_m^{\mathbf{N}}, Z_1, \ldots, Z_k}$ (for fixed $P_i^{\mathbf{N}}$'s) is a continuous reduction. Hence, by Corollary 3.1, any language definable in the sense of Definition 3.2, is also definable in the sense of Definition 3.1, adapted, if necessary, to ℓ-ary trees, for sufficiently large $\ell < \omega$.

We note that the converse is not true. Let

$$L_0 \quad = \quad \{(0^n 1)^\omega : n < \omega\}.$$

Proposition 3.2. The language L_0 is definable in the sense of Definition 3.1 (up to a renaming), but not in the sense of Definition 3.2.

Proof:
For the first part of the claim, we rename L_0 to the language $\{(1^n 2)^\omega : n < \omega\}$. It is easily definable, e.g., in a tree with one predicate P holding precisely in the nodes $v \in (1^n 2)^*$, for $n < \omega$. The defining formula ensures that the predicate holds infinitely often on the path.

For the second part, suppose the contrary and let $\mathcal{B}$ be a deterministic parity automaton satisfying (5). Like in the proof of Proposition 2.1, we use notation $u \star \alpha$ for the product of words $u \in (\{0,1\}^m)^\omega$ and $\alpha \in \{0,1\}^\omega$. Let $u_{\vec{P}}$ be the characteristic word of the tuple $P_1^{\mathbf{N}}, \ldots, P_m^{\mathbf{N}}$. Then $\mathcal{B}$ accepts $u_{\vec{P}} \star \alpha$ iff $\alpha = \alpha_n =_{def} (1^n 2)^\omega$, for some n. But then we can easily fool the automaton by swapping the prefixes of equal length of two different accepted words. More specifically, let K be greater than the number of states of $\mathcal{B}$. Then there are $0 \leq i < j \leq K$, such that the automaton assumes the same state q after reading the prefix of length $2K$ of the words $u_{\vec{P}} \star \alpha_i$ and $u_{\vec{P}} \star \alpha_j$. Decompose $\alpha_i = (\alpha_i \restriction 2K)\, \beta_i$. Then the automaton would also accept the word $u_{\vec{P}} \star (\alpha_j \restriction 2K)\, \beta_i$, violating (5). □

Acknowledgments. We thank Łukasz Kaiser and Filip Murlak for helpful comments on the preliminary version of this paper.

References

[1] Bárány, V., and Kaiser, Ł., and Rabinovich, A., Cardinality quantifiers in MLO over trees. In *Proc. 18th EACSL Annual Conference on Computer Science Logic*, Vol. 5771 of LNCS, pages 117-131, 2009.

[2] Büchi, J.R., On a decision method in restricted second order arithmetic. In *Logic, Methodology, and Philosophy of Science*, pp. 1–11, Stanford Univ. Press, 1960.

[3] Ebbinghaus, H.D., and Flum, J., and Thomas, W., *Mathematical Logic*. Springer, 1984.

[4] Emerson, E.A., and Jutla, C.S., The complexity of tree automata and logics of programs. In *Proc. 29th Foundations of Computer Science*, pp. 368–377, 1988.

[5] Fratani, S., *Automates à piles de piles... de piles*. PhD thesis, Université Bordeaux I, 2005 (in French).

[6] Kechris, A.S., *Classical Descriptive Set Theory*. Springer, 1995.

[7] Landweber, L.H. Decision problems for ω-automata. *Math. Systems Theory* 3 (1969) 376–384.

[8] Lifsches, S., and Shelah, S., Uniformization and Skolem functions in the class of trees. *J.Symbolic Logic*, Vol. 63(1) (1998), 103-127.

[9] McNaughton, R., Testing and generating infinite sequences by finite automaton. *Information and Control* **9** (1966), 521–530.

[10] Niwiński, D., and Walukiewicz, I., A gap property of deterministic tree languages. *Theoret. Comput. Sci.* **303** (2003), 215–231.

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

[12] Safra, S., On the complexity of ω-automata. In *Proc. 29th Foundations of Computer Science*, pp. 319–327, 1988.

[13] J. Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoret. Comput. Sci.* **112** (1993) 413–418.

[14] Thomas, W., Automata on infinite objects. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science* , volume B, pages 133–191. Elsevier Science Pub., 1990.

[15] Thomas, W., Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Springer-Verlag, 1997, pp. 389–455.