Regular languages of thin trees

Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak

University of Warsaw*
{bojan,idziaszek,mskrzypczak}@mimuw.edu.pl

Abstract

An infinite tree is called thin if it contains only countably many infinite branches. Thin trees can be seen as intermediate structures between infinite words and infinite trees. In this work we investigate properties of regular languages of thin trees.

Our main tool is an algebra suitable for thin trees. Using this framework we characterize various classes of regular languages: commutative, open in the standard topology, closed under two variants of bisimulational equivalence, and definable in WMSO logic among all trees.

We also show that in various meanings thin trees are not as rich as all infinite trees. In particular we observe a parity index collapse to level $(1, 3)$ and a topological complexity collapse to co-analytic sets. Moreover, a *gap property* is shown: a regular language of thin trees is either WMSO-definable among all trees or co-analytic-complete.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases infinite trees, regular languages, effective characterizations, topological complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.562

1 Introduction

Since the decidability results by Büchi [7] and Rabin [17], regular languages of infinite words and trees have been intensively studied. Those languages can be equivalently described in monadic second-order (MSO) logic, by nondeterministic finite automata, or in terms of homomorphisms to finite algebras. Apart from the emptiness problem, which is known to be decidable, one ask about decidability for other, more subtle properties of a given language.

Suppose that X is a subclass of regular languages of infinite trees, e.g. X can be the languages that are definable in first-order (FO) logic with descendant; or definable in weak MSO (WMSO); or recognized by a nondeterministic parity automaton with priorities $\{i, \ldots, j\}$. An *effective characterization for* X is an algorithm which inputs a regular language of infinite trees and answers if the language belongs to X. As far as decidability is concerned the representation of the language is not very important, since there are decidable translations between the many ways of representing regular languages of infinite trees.

Effective characterizations are a lively and important topic in the theory of regular languages. In the case of finite words there are many celebrated results, e.g. characterizations of FO [18], two-variable FO [21] or piecewise testable languages [19]. Many of these results carry over to infinite words, see [23], [16], or [12]. For finite trees much less is known, but still there are some techniques [3]. The main reason why effective characterizations are studied is that an effective characterization of a class X requires a deep insight into the structure of the class. Usually this insight is achieved through an algebraic framework, such as semigroups for finite words, Wilke semigroups for infinite words, or forest algebra for finite trees. Apart

* All authors were supported by ERC Starting Grant "Sosna" no. 239850

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 562–573

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


from having a well-developed structure theory, another advantage of algebra is that many effective characterizations can be elegantly stated in terms of identities.

Effective characterizations are technically challenging, and in fact there are very few effective characterizations for languages of infinite trees: for languages recognized by top-down deterministic automata one can compute the Wadge degree [14], for arbitrary regular languages one can decide definability in the temporal logic EF [4] or in the topological class of Boolean combinations of open sets [5]. One of the reasons why effective characterizations are so difficult for infinite trees is that, so far, there is no satisfactory algebraic approach to infinite trees, or even a canonical way to present a regular language. Proposed algebras (see [4], [2]) either have no finite representation or yield no effective characterizations.

In this paper, we propose to study *thin trees*, which generalize both finite trees and infinite words, but which are still simpler than arbitrary infinite trees. A tree is called thin if it has only countably many infinite branches (or equivalently, it does not contain a full binary tree as a minor). We believe that thin trees are a good stepping stone on the way to understanding regular languages of arbitrary infinite trees.

Our contributions can be divided into two sets:

Effective characterizations. We characterize the following classes of regular languages of thin trees in terms of finite sets of identities:

- closed under rearranging of siblings,
- closed under bisimulation equivalence (in two variants),
- open in the standard topology,
- definable in the temporal logic EF,
- definable among all forests in WMSO logic.

The crucial ingredient of these characterizations is an observation that a regular language of thin trees can be canonically represented by a finite algebraic object, called its syntactic thin-forest algebra. For general trees no such representation is known.

Upper bounds. We show that in various contexts thin trees are not as rich as generic trees:

- The Rabin-Mostowski index hierarchy collapses to level $(1, 3)$ on thin trees.
- The projective hierarchy of regular languages collapses to level $\mathbf{\Pi}^1_1$ on thin trees (comparing to $\mathbf{\Delta}^1_2$ in the case of all trees).
- We observe a *gap property* (see [15]): a regular language of thin trees treated as a subset of all trees is either definable in WMSO logic or non-Borel.
- If we treat thin trees as our universe then no regular language is topologically harder than Borel sets.

2 Preliminaries

This section introduces basic notions and facts used in the proofs. To avoid technical difficulties when introducing algebras, we operate on finitely branching forests instead of partial binary trees. The difference is only technical, all the results can be naturally transferred back to the framework of partial binary trees.

2.1 Forests

Fix a finite alphabet A. By A^{For} we denote the set of all A-labelled forests. Formally a forest is a partial mapping from its set of nodes $\mathrm{dom}(t) \subset \omega^+$ into A. We additionally assume that a forest is finitely branching: for every $w \in \omega^*$ there are only finitely many nodes of the form $w0, w1, w2, \ldots, wn$ in $\mathrm{dom}(t)$. For $w = \epsilon$ those nodes are called *roots* of the forest

t and for $w \neq \epsilon$ these are *children* of the node w. In both cases the list of nodes of the form wn ordered by n is called a *list of siblings in* t.

A node $w \in \mathrm{dom}(t)$ is *branching* if it has at least two distinct children $wn_1, wn_2 \in \mathrm{dom}(t)$. A node in $\mathrm{dom}(t)$ is a leaf of t if it has no children in t.

A forest with exactly one root is called a *tree*. The empty forest is denoted as 0. For a given forest t and a node $x \in \mathrm{dom}(t)$ by $t\restriction_x$ we denote the subtree of t rooted in x: $\mathrm{dom}(t\restriction_x) = \{0 \cdot w \in \omega^* : \ xw \in \mathrm{dom}(t)\}$, $t\restriction_x (0 \cdot w) = t(xw)$.

Let t be a forest. A sequence $\pi \in \omega^*$ is a *finite branch of* t if either $\pi = \epsilon$ and $t = 0$ or $\pi \in \mathrm{dom}(t)$ and π (as an element of ω^+) is a leaf of t. A sequence $\pi \in \omega^\omega$ is an *infinite branch of* t if for every sequence $w \in \omega^+$ such that $w \prec \pi$ we have that w is a node of t.

A forest is *regular* if it has finitely many distinct subtrees. A forest is *thin* if it has countably many branches. The set of all thin forests is denoted as $A^{\mathsf{ThinFor}} \subset A^{\mathsf{For}}$. A forest is thin if and only if it is a *tame tree* in the meaning of [13].

We say that a forest s is a *prefix* of a forest t if $\mathrm{dom}(s) \subseteq \mathrm{dom}(t)$ and for every $x \in \mathrm{dom}(s)$ we have $s(x) = t(x)$. We denote it by $s \subseteq t$.

Let t be a forest and $s \subseteq t$ be a prefix of t. A node $y \in t$ is *off* s if $y \notin s$ and either y is a root, or the parent of y is in s. Since a branch π of t can be treated as a prefix of t this definition also extends to branches.

An A-labelled *context* is a forest over the alphabet $A \cup \{\Box\}$, where the label $\Box$ is a special marker, called the *hole*, which occurs exactly once and in a leaf. A context is *guarded* if its hole is not in a root. For every letter $a \in A$ we denote by $a\Box$ the single-letter tree context with a in the root and the hole below it.

Since we are interested in algebraic frameworks for forests, we need a set of operations which will allow to build forest from basic elements. Following [6] we introduce following operations on forests. For a graphical presentation of these operations, compare Figure 1 and Figure 2 in [6]. We can

- concatenate two forests s, t, which results in the forest $s + t$,
- compose a context p with a forest t, which results in the forest pt, obtained from p by replacing the hole with t,
- compose a context p with a context q, which results in the context pq that satisfies $(pq)t = p(qt)$.

We write at, ap for a composition of a single-letter context $a\Box$ with t or p (thus $a0$ is a forest of one node labelled a). Additionally we have an operation which allows us to produce infinite forests:

- compose a guarded context p with itself infinitely many times, which results in the forest p^∞ that satisfies $p(p^\infty) = p^\infty$. Note that we exclude non-guarded contexts from this definition. (For example the result of $(\Box + a0)^\infty$, even if well-defined, is not finitely branching.)

2.2 Automata and regular languages

A (nondeterministic parity) *forest automaton* over an alphabet A is given by a set of states Q equipped with a monoid structure, a transition relation $\Delta \subseteq Q \times A \times Q$, a set of initial states $Q_I \subseteq Q$ and a parity condition $\Omega\colon Q \to \mathbb{N}$. We use additive notation $+$ for the monoid operation in Q, and we write 0 for the neutral element.

We say that a forest automaton $\mathcal{A}$ *has index* (i, j) (or shortly that $\mathcal{A}$ is (i, j)-automaton) if i is the minimal and j is the maximal value of Ω on Q.

A *run* of this automaton over a forest t is a labelling $\rho\colon \mathrm{dom}(t) \to Q$ of forest nodes with states such that for any node x with children $x_1, \ldots, x_n$

$$(\rho(x_1) + \rho(x_2) + \cdots + \rho(x_n), t(x), \rho(x)) \in \Delta.$$

Note that if x is a leaf, then the above implies $(0, t(x), \rho(x)) \in \Delta$.

A run is *accepting* if for every (infinite) branch π of t, the highest value of $\Omega(q)$ is even among those states q which appear infinitely often along the branch π. The *value* of a run over a forest t is obtained by adding, using $+$, all the states assigned to roots of the forest. A forest is *accepted* if it has an accepting run whose value belongs to Q_I. The set of forests accepted by an automaton is called the language *recognized* by the automaton.

A language is *regular* if it is definable by a formula of monadic second-order logic (MSO).

▶ **Theorem 1** ([10]). *A language of thin forests is regular if and only if it is recognized by some forest automaton. Every nonempty language of thin forests contains a regular forest.*

We use MSO logic to describe properties of infinite forests. An infinite forest is treated as a relational structure, where the universe is the nodes, and the predicates are: a binary child predicate, a binary next sibling predicate, and one unary predicate for each label in the alphabet. Additionally, we consider WMSO: the logic with the same syntax as MSO but with the semantical restriction that all set quantifiers range over finite subsets of the domain. Since the property that a given set is finite is MSO-definable on finitely branching infinite forests, so WMSO can be naturally embedded into MSO. There are examples of languages of infinite forests that are definable in MSO but not in WMSO.

2.3 Topology

A topological space X is *Polish* if it is separable and has a complete metrics. Polish topological spaces are the principal objects studied in descriptive set theory.

The set of forests A^{For}, equipped with the natural Tikhonov topology, is an uncountable Polish topological space. The base of the topology is given by the sets of the form $\{t : t\restriction_{\omega \leq d} = r\}$ for finite forests r and a number (depth) d.

Let X be an uncountable Polish topological space. The class of open sets in X is denoted as $\mathbf{\Sigma}^0_1(X)$. The class of complements of open sets (called closed) is denoted as $\mathbf{\Pi}^0_1(X)$. The Borel hierarchy is defined inductively, the building ingredients are countable unions and intersections. For a countable ordinal α let:

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

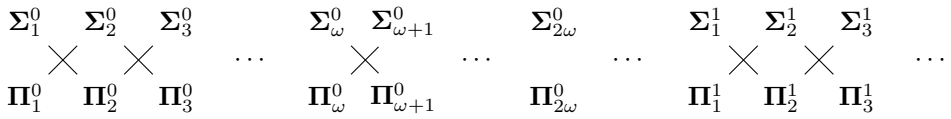
The class of Borel sets is the union of all classes $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$ for $\alpha < \omega_1$. A more detailed introduction to the Borel hierarchy can be found e.g. in [11, Chapter II]. If the space is clear from the context we will omit it and write just $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$.

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 $\mathbf{\Sigma}^1_1$. The superscript 1 means that the class is a part of the projective hierarchy. The rest of the projective hierarchy is defined as follows:

- $\mathbf{\Pi}^1_i$ consists of the complements of the sets from $\mathbf{\Sigma}^1_i$,
- $\mathbf{\Sigma}^1_{i+1}$ consists of the projections of the sets from $\mathbf{\Pi}^1_i$.

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*. The most important property of this hierarchy is strictness: all the inclusions on the following diagram are strict.



▶ Fact 2. Every regular language of forests is in the intersection of $\mathbf{\Sigma}^1_2$ and $\mathbf{\Pi}^1_2$ (denoted by $\mathbf{\Delta}^1_2$).

The set of thin forests A^{ThinFor} is $\mathbf{\Pi}^1_1(A^{\mathsf{For}})$-complete, thus non-Borel.

2.4 Ranks and skeletons

The crucial tool in our analysis of thin forests is structural induction — we inductively decompose a given forest into *simpler* ones. A measure of complexity of thin forests is called a *rank* — a function that assigns to each thin forest a countable ordinal number. The rank we use, denoted CB-rank (or shortly $\mathrm{rank}^{\mathrm{CB}}$), is based on the Cantor-Bendixson derivative on closed subsets of ω^ω.

Intuitively, a forest t has $\mathrm{rank}^{\mathrm{CB}}$ equal M if t contains M levels of infinite branches:

- The CB-rank of the empty forest is 0,
- The CB-rank of a forest with finitely many branches is 1,
- if s is a prefix of t of rank 1 and for every x that is off s we have $\mathrm{rank}^{\mathrm{CB}}(t\restriction_x) \leq M$, then $\mathrm{rank}^{\mathrm{CB}}(t) \leq M+1$.

The set of forests of CB-rank bounded by a given ordinal η is denoted as $A^{\mathsf{ThinFor}\leq\eta}$.

The second tool used to analyze structural properties of thin forests are *skeletons*. A skeleton can be seen as a witness that a given forest is thin. Moreover, a skeleton of a thin forest t represents a structural decomposition of t.

A subset of nodes $\sigma \subseteq \mathrm{dom}(t)$ of a given forest $t \in A^{\mathsf{For}}$ is a *skeleton of* t if:

- from every set of siblings in t exactly one is in σ,
- on every infinite branch π of the forest t almost all nodes $x \prec \pi$ belong to σ.

Observe that we can identify σ with its characteristic function — a labelling of nodes of t by $\{0,1\}$. Therefore, $\sigma \in \{0,1\}^{\mathsf{For}}$ and we can treat a pair of a forest and a skeleton (t,σ) as an element of $A \times \{0,1\}^{\mathsf{For}}$.

An easy inductive argument shows that a forest t has a skeleton if and only if t is a thin forest. Moreover, for every thin forest t one can define its *canonical skeleton* $\sigma(t)$.

3 Algebra

In this section we define thin-forest algebra. Its operations and axioms are constructed in such a manner that the free object of this algebra is the set of all regular thin forests and regular thin contexts. Thin-forest algebra is a common generalization of both Wilke algebra [22] and forest algebra [6].

A thin-forest algebra is a three-sorted algebra $(H, V_+, V_\Box, act, in_l, in_r, inf)$. It consists of two monoids H and $V = V_+ \cup V_\Box$ (partitioned into a subsemigroup V_+ and a submonoid $V_\Box$) along with an operation of left action $act\colon H \times V \to H$ of V on H, two operations $in_l, in_r\colon H \to V_\Box$ and an infinite loop operation $inf\colon V_+ \to H$. Instead of writing $act(h,v)$, we write vh (notice a reversal of arguments). Instead of writing $inf(v)$, we write v^∞. We will call H the *horizontal monoid* and V the *vertical monoid*.

The above construction is based on forest algebra (see [6]). In fact we take forest algebra and introduce the new operation *inf*; this operation corresponds to infinite composition

of contexts. However, since infinite composition is defined only for guarded contexts, we are forced to make a distinction between guarded and non-guarded objects, therefore we partition the sort V into two parts V_+ and $V_\Box$ respectively.

3.1 Axioms and free objects

A thin-forest algebra must satisfy the following axioms:

A1. $(H, +, 0)$ is a monoid with operation $+$ and neutral element 0,

A2. $(V, \cdot, \Box)$ is a monoid with operation $\cdot$ and neutral element $\Box$; it contains two disjoint subalgebras: $(V_\Box, \cdot, \Box)$ is a monoid and $(V_+, \cdot)$ is a semigroup,

A3. (action axiom) $(vw)h = v(wh)$ for every $v, w \in V$, $h \in H$,

A4. (insertion axiom) $in_l(h)g = h + g$, $in_r(h)g = g + h$ for every $h, g \in H$,

A5. $(vw)^\infty = v(wv)^\infty$ for $v, w \in V$, excluding the case when $v, w \in V_\Box$,

A6. $(v^n)^\infty = v^\infty$ for $v \in V_+$ and every $n \geq 1$.

Given an alphabet A we define the *free thin-forest algebra* over A, which is denoted by $A^{\mathsf{regThin}^\triangle}$, as follows:

1. the horizontal monoid is the set of regular thin forests over A, with the operation of forest concatenation;
2. the vertical monoid is the set of regular thin contexts over A (respectively guarded and non-guarded), with the operation of context composition;
3. the action is the substitution of forests in contexts;
4. the in_l operation takes a regular thin forest and transforms it into a regular thin context with the hole to the right of all the roots in the forest (similarly for in_r but the hole is to the left of the roots);
5. the infinite loop operation takes a regular thin context and transforms it into a regular thin forest by performing infinite composition.

▶ **Theorem 3.** *The algebra $A^{\mathsf{regThin}^\triangle}$ is a thin-forest algebra. Moreover it is the free algebra in the class of thin-forest algebras over the generator set $A\Box = \{a\Box : a \in A\}$.*

Since the insertion operations are somewhat cumbersome to use, we will use the operation $+$ to concatenate forests with contexts, meaning $h + v = in_l(h)v$, $v + h = in_r(h)v$.

We note that it is possible to introduce an algebra where the free object would be the set of all thin forests and all thin contexts (not only regular ones). This can be done by generalizing ω-semigroups. However, since regular languages of forests are uniquely described by regular forests which they contain, this more general algebra gives us the same information about the language as thin-forest algebra. See [10] for more details.

3.2 Recognizability by thin-forest algebra and regularity

A *morphism* between two thin-forest algebras is defined in a natural way. A set L of thin forests over an alphabet A is *recognized* by a morphism $\alpha\colon A^{\mathsf{regThin}^\triangle} \to (H, V)$ if $L = \alpha^{-1}(I)$ for some $I \subseteq H$.

We will consider terms in the signature of thin-forest algebra with typed variables. Variables can be of type τ_H, τ_V, or τ_{V_+}, which means that a valuation of a term should assign to the variable an element of the sort H, V or V_+ respectively. Similarly a term is of certain type if a valuation of this term results in an element from the corresponding sort.

Two thin forests t, s are *L-equivalent* if for every term σ over the signature of thin-forest algebra of type τ_H of one variable x of type τ_H, either both or none of the forests

$\sigma[x \leftarrow t], \sigma[x \leftarrow s]$ belong to L (note that we evaluate the term σ in the free thin-forest algebra). Similarly we define the L-equivalence of contexts (but now the variable x is of type τ_V).

The relation of L-equivalence is a congruence, and the quotient of $A^{\mathsf{regThin}^\triangle}$ with respect to L-equivalence is the *syntactic thin-forest algebra* for L. The *syntactic morphism* of L assigns to every element of $A^{\mathsf{regThin}^\triangle}$ its equivalence class in the syntactic thin-forest algebra of L.

▶ **Theorem 4.** *A language of thin forests is recognizable by a finite thin-forest algebra if and only if it is regular. Every regular language of thin forests is recognizable by its syntactic morphism. The syntactic thin-algebra and the syntactic morphism can be effectively calculated, based on a parity automaton.*

Let L be a regular language of thin forests and $\alpha\colon A^{\mathsf{regThin}^\triangle} \to (H, V)$ its syntactic morphism. We say that an element $h \in H$ is *the bottom element for* L if $\alpha^{-1}(h) \cap L = \emptyset$ and $vh = h$ for every $v \in V$.

Note that the bottom element is unique, since if h_1 and h_2 are both bottom elements, then $h_1 = (\Box + h_2)h_1 = h_1 + h_2 = (h_1 + \Box)h_2 = h_2$.

4 Applications of thin-forest algebra

In this section we show how thin-forest algebra can be used to give decidable characterizations of certain properties of languages. Many such characterizations boil down to checking whether the syntactic algebra of a given regular language satisfies a set of identities. An *identity* is a pair of terms (of the same type) in the signature of thin-forest algebra over typed variables. An algebra satisfies an identity if for every valuation the two terms have the same value. We usually assume that the operation $v \mapsto v^\omega$ is a part of the signature. This operation assigns to every $v \in V$ its *idempotent power*, i.e. such a power v^k that satisfies $v^k \cdot v^k = v^k$. For every v there exists a unique idempotent power, since V is a semigroup [16] (the number k is not unique, but the value v^k is).

In the following subsections we show how to decide whether a given regular language of thin forests is commutative, invariant under bisimulation, open in the standard topology, and definable by a formula of the temporal logic EF.

4.1 Commutative languages

The notion of *commutative language* of finite forests is quite natural: it is a language closed under rearranging of siblings. In the case of finite forests, a language is commutative if and only if its syntactic algebra satisfies the identity

$$h + g = g + h \qquad \text{for } g, h \in H. \tag{1}$$

In the case of infinite forests we have more flexibility. We get different "degrees of commutativity" by allowing rearranging of siblings finitely many times, finitely many times on every branch, or arbitrarily many times. We think that the last (unrestricted) definition is the most appealing. However, it is not captured by the identity (1). Consider the language $L =$ "every node has 0 or 2 children and every branch goes left only finite number of times". The language L does satisfy (1), but it is not commutative, as witnessed by two thin forests $a(a0 + a\Box)^\infty \in L$, $a(a\Box + a0)^\infty \notin L$. The problem with the above example is that we would like to be able not only to rearrange forests, but also to rearrange a forest with a context. This property is expressed by the following identity:

▶ **Theorem 5.** *A regular language of thin forests L is commutative if and only if its syntactic thin-forest algebra satisfies the identity*

$$h + v = v + h \qquad \text{for } h \in H \text{ and } v \in V.$$

Identity (1) corresponds to a weaker notion of commutativity, where on every branch we allow only finite number of rearrangements of siblings (see [10]).

4.2 Languages invariant under bisimulation

Two forests t_0 and t_1 are called *bisimilar* if Duplicator wins the following game, which is played by players Spoiler and Duplicator. Spoiler begins the game by choosing some $i \in \{0,1\}$ and a root node x_i of the forest t_i. Duplicator responds by chosing a root node x_{1-i} of the other forest t_{1-i}, which has the same label (if no such node exists, the game is terminated and Spiler wins). For $i \in \{0,1\}$, let s_i be the forest obtained by taking the subtree of t_i rooted in x_i and removing the root. If Duplicator did not lose, then a new round of the game is played with the forests being s_0 and s_1. Duplicator wins if infinitely many rounds are played without Spoiler winning.

A language of thin forests L is called *invariant under bisimulation* if for every forests which are bisimilar, either both or none belong to L.

▶ **Theorem 6.** *A regular language of thin forests L is invariant under bisimulation if and only if its syntactic thin-forest algebra satisfies the following identities:*

$$h + v = v + h, \qquad h + h = h, \qquad (w^\infty + w)^\infty = w^\infty \qquad \text{for } v \in V,\ w \in V_+ \text{ and } h \in H.$$

4.3 Open languages

In this section we give a characterization of the class of languages that are open in the standard topology on forests (see Section 2.3). An equivalent definition says that a forest language L is open if for every forest $t \in L$ there is a finite prefix of t such that changing nodes outside of the prefix does not affect membership in L. Checking whether a given regular forest language L is open was known to be decidable, our contribution lies in showing that for thin forests it can be done by testing the syntactic morphism of L:

▶ **Theorem 7.** *A regular language of thin forests L is open if and only if its syntactic morphism $\alpha\colon A^{\mathsf{regThin}^\triangle} \to (H, V)$ satisfies the following condition for $v \in V_+$ and $h \in H$:*

$$\text{if} \quad v^\infty \in \alpha(L) \quad \text{then} \quad v^\omega h \in \alpha(L).$$

The notion of open sets is also applicable to the case of infinite words. It is interesting to note that the above condition also characterizes open languages of infinite words.

Moreover, one can extend the theory of ordered algebras (see [16]) to thin-forest algebras. Then the above condition could be simply stated as $v^\infty \geq v^\omega h$.

4.4 Temporal logic EF

The logic EF is a simple temporal logic which uses only one operator EF, which stands for "Exists Finally". Formulas of the logic EF are defined as follows:

1. every letter a is an EF formula, which is true in trees with root label a,
2. EF formulas admit Boolean operations, including negation,

3. if φ is an EF formula, then $\mathsf{EF}\varphi$ is an EF formula, which is true in trees that have a proper subtree where φ is true.

A tree t satisfies an EF formula φ if φ holds in the root of the tree t. There are some technical difficulties with generalizing this definition to forests, therefore we will only allow Boolean combinations of formulas of the form $\varphi \vee \mathsf{EF}\varphi$ to describe forests (we call them forest EF formulas; a forest t satisfies such a formula if φ holds in any node of t).

A forest language L is *invariant under EF-bisimulation* if for every forests t_0, t_1 which are EF-bisimilar either both or none belong to L. The relation of EF-bisimilarity is similar to the relation of bisimilarity, but in the game Spoiler chooses an arbitrary node x_i of t_i (not necessarily a root), and Duplicator responds with an arbitrary node x_{1-i} of t_{1-i}. Note that if t_1, t_2 are EF-bisimilar and φ is an forest EF formula then $t_1 \models \varphi$ if and only if $t_2 \models \varphi$.

The following theorem (in a version for general infinite forests) was proved in [4]:

▶ **Theorem 8.** *A regular language of thin forests L can be defined by a forest EF formula if and only if*

1. *it is invariant under EF-bisimulation,*
2. *its syntactic thin-forest algebra satisfies the identity $v^\omega h = (v + v^\omega h)^\infty$ for $v \in V_+$ and $h \in H$.*

For forests that are not necessarily thin, we could not find how to express the first condition in terms of identities. We show how to do it in the case of thin forests:

▶ **Theorem 9.** *A regular language of thin forests L is invariant under EF-bisimulation if and only if its syntactic thin-forest algebra satisfies the identities for $v, u \in V$, $w \in V_+$, $h \in H$:*

$$h + v = v + h, \qquad vh = vh + h, \qquad (w + (wv)^\infty)^\infty = (wv)^\infty, \qquad (wvu)^\infty = (wuv)^\infty.$$

5 Descriptive properties

5.1 Automata

First we show that it is possible to recognize regular languages of thin forests using „simple" automata.

▶ **Theorem 10.** *Every regular language of thin forests can be recognized among all forests by a $(1,3)$-automaton.*

The principal idea is to guess a skeleton of a given forest and use nondeterministic Büchi automata on the branches of this skeleton to verify the types in the syntactic algebra.

The following theorem expresses that the collapse from Theorem 10 is the best we can get from the point of view of the alternating index hierarchy (also known as the Rabin-Mostowski hierarchy).

▶ **Theorem 11.** *There exists a regular language of thin forests L that is not recognizable among all forests by any alternating $(1,2)$-automaton nor any alternating $(0,1)$-automaton.*

The following theorem shows that regular languages of thin forests can be recognized by unambiguous automata *relatively* to thin forests. It is especially interesting, since there are regular languages of forests that are not unambiguous, one of the examples is the language „exists a node labelled by the letter a" (see [8]). The following theorem implies that the language of thin forests containing a letter a is unambiguous among thin forests.

▶ **Theorem 12.** *For every regular language of thin forests L there exists a nondeterministic forest automaton $\mathcal{A}$ such that $\mathrm{L}(\mathcal{A}) \cap A^{\mathsf{ThinFor}} = L$ and for every thin forest $t \in L$ there exists exactly one accepting run of $\mathcal{A}$ on t.*

The proof is based on a modification of a technique (called algebraic automata) proposed by Marcin Bilkowski [1]. The idea is the following: we construct an automaton $\mathcal{A}$ that guesses a marking τ of nodes of the given forest t by types in the syntactic algebra of L. Then $\mathcal{A}$ runs on top of τ a deterministic top-down automaton verifying the following property:

> For every node x and every infinite branch π that goes through x, the type guessed in x is consistent with the guessed types of nodes that are off π and letters of t on π.

5.2 Languages that are WMSO-definable among all forests

In this section we consider a nonstandard approach to restricting the family of all forests to thin ones. In this setting we show that it is decidable whether a given regular language of thin forests is WMSO-definable. The difference between the standard approach and the one used in this section is that we do not implicitly restrict our universe to thin forests.

▶ **Definition 13.** Let L be a regular language of thin forests and φ be a formula of WMSO. We say that φ *defines L among all forests* if $L = \{t \in A^{\mathsf{For}} : t \models \varphi\}$.

Note that the class of languages definable in WMSO among all forests is not closed under complement with respect to thin forests: the relative complement of the empty language $\emptyset \subseteq A^{\mathsf{ThinFor}}$ is A^{ThinFor} which is not WMSO-definable among all forests.

The following fact says that even in this restricted setting we can define languages as complicated as in the general case.

▶ Fact 14. The examples of WMSO-definable languages lying arbitrarily high on the finite levels of the Borel hierarchy (see [20]) can be encoded into thin forests in a way WMSO-definable among all forests.

The main result of this section is the following characterization.

▶ **Theorem 15.** *Let L be a regular language of thin forests. The following conditions are equivalent:*

1. *there exists $M \in \mathbb{N}$ such that every forest $t \in L$ satisfies $\mathrm{rank}^{CB}(t) \leq M$,*
2. *L is WMSO-definable among all forests,*
3. *L is not $\mathbf{\Pi}^1_1(A^{\mathsf{For}})$-hard,*
4. *the syntactic morphism for L satisfies the following condition:*

$$\begin{aligned} &\text{if } h = v(w+h)^\infty \text{ or } h = v(h+w)^\infty \text{ for some } v \in V, w \in V_+, \\ &\qquad \text{then } h \text{ is the bottom element for } L. \end{aligned} \tag{2}$$

The following list presents a sketch of the argumentation.

From 1 to 2. A direct construction of a formula.

From 2 to 3. Folklore.

From 3 to 4. A pumping argument: a counterexample to the equations can be used to construct a continuous function f from the space of trees over ω to A^{For}. If a given tree t is well-founded (does not contain an infinite branch) then the result $f(t)$ is in L. Otherwise the result $f(t)$ is not thin, therefore does not belong to L. Since the set of well-founded trees over ω is $\mathbf{\Pi}^1_1$-hard then so is L (f is a continuous reduction).

From 4 to 1. Estimating: condition (2) introduces an order on types in H. The height of this order bounds the maximal CB-rank of forests in the language L.

Note that the last condition in the theorem is effective, therefore we obtain the following corollary.

▶ **Corollary 16.** *It is decidable whether a given regular language of thin forests L is WMSO-definable among all forests.*

▶ Proposition 17. Assume that L is a regular language of forests that is recognized by a nondeterministic (or equivalently alternating) $(1,2)$-automaton. Assume additionally that L contains only thin forests. Then L can be defined in WMSO among all forests.

Proof. Since L is recognizable by a $(1,2)$-automaton so L is an analytic subset of A^{For}. Therefore, L cannot be $\mathbf{\Pi}^1_1$-hard, thus L satisfies the condition 3 in Theorem 15. ◀

5.3 Topological properties

In this section we give a couple of results showing that regular languages of thin forests are topologically simpler then generic regular languages of forests.

▶ **Theorem 18.** *Every regular language of thin forests L is co-analytic as a set of forests.*

Note that despite the fact that the space of thin forests A^{ThinFor} is co-analytic among all forests, it contains arbitrarily complicated subsets. In fact, already the family of forests of CB-rank equal 1 is an uncountable Polish topological space, so the whole boldface hierarchy (see Section 2.3) can be constructed using only such forests.

Theorems 15 and 18 imply the following dichotomy or *gap property* in the spirit of [15].

▶ Remark. For every regular language of thin forests L exactly one of the following possibilities holds, it can be effectively decided which one:

- L is WMSO-definable among all forests and lies on a finite level of the Borel hierarchy,
- L is $\mathbf{\Pi}^1_1(A^{\mathsf{For}})$-complete.

The following theorem shows that, when treating thin forests as our universe, there are no topologically hard regular languages.

▶ **Theorem 19.** *Let X be a Polish topological space, $f\colon X \to A^{\mathsf{ThinFor}}$ be continuous and L be a regular language of thin forests. Then $f^{-1}(L)$ is Borel in X.*

The following theorem can be seen as complementing Theorem 19.

▶ **Theorem 20.** *There exists a regular language of thin forests L_W over an alphabet A_W that is* Borel-hard*: for every Polish topological space X and every Borel set $B \subseteq X$ there exists a continuous function $f\colon X \to {A_W}^{\mathsf{ThinFor}}$ such that $f^{-1}(L_W) = B$.*

The principal concept of the above language is based on a construction proposed in [9]. Using the structure of the language L_W one can deduce the following corollary.

▶ **Corollary 21.** *The language L_W cannot be defined in WMSO among thin forests.*

This statement holds true even if we provide with every forest $t \in {A_W}^{\mathsf{ThinFor}}$ its canonical skeleton $\sigma(t)$: there is no WMSO formula φ over the alphabet $A_W \times \{0,1\}$ such that

$$L_W = \left\{ t \in {A_W}^{\mathsf{ThinFor}} :\ (t, \sigma(t)) \models \varphi \right\}.$$

Acknowledgements

The authors would like to thank Henryk Michalewski for posing a number of motivating problems and questions on the subject. Additionally, the authors thank the referees for suggestions and comments.

References

1 M. Bilkowski. Algebraic automata. Private communication, 2011.
2 A. Blumensath. Recognisability for algebras of infinite trees. *Theor. Comput. Sci.*, 412(29):3463–3486, 2011.
3 M. Bojańczyk. Effective characterizations of tree logics. In *PODS*, pages 53–66, 2008.
4 M. Bojańczyk and T. Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *CONCUR*, pages 131–145, 2009.
5 M. Bojańczyk and T. Place. Regular languages of infinite trees that are boolean combinations of open sets. In *ICALP*, pages 104–115, 2012.
6 M. Bojańczyk and I. Walukiewicz. Forest algebras. In *Logic and Automata*, pages 107–132, 2008.
7 J.R. 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.
8 A. Carayol, Ch. Löding, D. Niwiński, and I. Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *CEJM*, 8:662–682, 2010.
9 S. Hummel, H. Michalewski, and D. Niwiński. On the Borel inseparability of game tree languages. In *STACS*, pages 565–575, 2009.
10 T. Idziaszek. *Algebraic methods in the theory of infinite trees*. PhD thesis, University of Warsaw, 2013. Unpublished.
11 A. Kechris. *Classical descriptive set theory*. Springer-Verlag, New York, 1995.
12 M. Kufleitner and A. Lauser. Languages of dot-depth one over infinite words. In *LICS*, pages 23–32, 2011.
13 S. Lifsches and S. Shelah. Uniformization and skolem functions in the class of trees. *J. Symb. Log.*, 63(1):103–127, 1998.
14 F. Murlak. The Wadge hierarchy of deterministic tree languages. *LMCS*, 4(4), 2008.
15 D. Niwiński and I. Walukiewicz. A gap property of deterministic tree languages. *TCS*, 1(303):215–231, 2003.
16 D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.
17 M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Bull. Amer. Math. Soc.*, 74:1025–1029, 1968.
18 M.P. Schützenberger. On finite monoids having only trivial subgroups. *Inf. and Cont.*, 8(2):190–194, 1965.
19 I. Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222, 1975.
20 J. Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *TCS*, 112(2):413–418, 1993.
21 D. Thérien and T. Wilke. Over words, two variables are as powerful as one quantifier alternation. In *STOC*, pages 234–240, 1998.
22 T. Wilke. Classifying discrete temporal properties. Habilitationsschrift, Universitat Kiel, apr. 1998.
23 Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *Int. J. Alg. Comput.*, 3:447–489, 1993.