Theory Comput Syst
DOI 10.1007/s00224-014-9595-z

Regular Languages of Thin Trees

Tomasz Idziaszek · Michał Skrzypczak ·
Mikołaj Bojańczyk

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, and definable in weak MSO 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 collapse of the parity index to the level $(1, 3)$ and a collapse of the topological complexity to co-analytic sets. Moreover, a *gap property* is shown: a regular language of thin trees is either weak MSO-definable among all trees or co-analytic-complete.

Keywords Infinite trees · Regular languages · Effective characterizations · Topological complexity

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

T. Idziaszek · M. Skrzypczak (✉) · M. Bojańczyk
Institute of Informatics, University of Warsaw, Warsaw, Poland
e-mail: mskrzypczak@mimuw.edu.pl

T. Idziaszek
e-mail: idziaszek@mimuw.edu.pl

M. Bojańczyk
e-mail: bojan@mimuw.edu.pl

1 Introduction

Since the decidability results by Büchi [7] and Rabin [18], regular languages of infinite words and trees have been studied intensively. These languages can be equivalently described in monadic second-order (MSO) logic, by non-deterministic finite automata, or in terms of homomorphisms to finite algebras. Apart from the emptiness problem, which is known to be decidable, one can 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 monadic second-order (weak MSO) logic; or recognized by a non-deterministic 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 effective 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 [20], two-variable FO [24], or piecewise testable languages [21]. Many of these results carry over to infinite words, see [17, 25], or [13]. 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 algebras for infinite words, or forest algebras for finite trees. Apart 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 [15], for arbitrary regular languages one can decide definability in the temporal logic EF [1] or in the topological class of Boolean combinations of open sets [4]. 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. The algebras proposed so far are not completely satisfactory. The algebras proposed in [1] can recognise non-regular languages, while the algebras proposed in [2] are not closed under homomorphisms.

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.

The developments presented in this paper, in particular introduction of thin forest algebra, lead to some new results on unambiguity and uniformization on infinite trees, see [6].

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,
– open in the standard topology,
– definable in the temporal logic EF,
– definable among all trees in weak MSO 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 the level $(1, 3)$ on thin trees.
– The projective hierarchy of regular languages collapses to the 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 [16]): a regular language of thin trees, treated as a subset of all trees, is either definable in weak MSO logic or $\mathbf{\Pi}^1_1$-complete.
– 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 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 wn for $n \in \mathbb{N}$ in $\mathrm{dom}(t)$. We assume that these nodes are $w0, w1, w2, \ldots, wm$ for some m. For $w = \epsilon$ these 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 the nodes of the form wn ordered by n is called a *list of siblings in* t. The prefix order on nodes of a forest is denoted $x \preceq y$.

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 $t\!\upharpoonright_x$ we denote the subtree of t rooted in x: $\mathrm{dom}(t\!\upharpoonright_x) = \{0w \in \omega^* :\ xw \in \mathrm{dom}(t)\}$, $t\!\upharpoonright_x(0w) = t(xw)$.

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 only finitely many distinct subtrees. A forest is *thin* if it has countably many branches. The set of all thin forests is denoted as $A^{\mathsf{ThinFor}} \subset A^{\mathsf{For}}$.

Some authors use different names to denote tree-like structures with countably many branches. We recall here two of them: a forest that is a tree is thin if and only if it is a *scattered tree* in the meaning of [19]. A forest is thin if and only if it is, up to isomorphism, a *tame tree* in the meaning of [14].

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 this fact by $s \subseteq t$.

Let t be a forest and $s \subseteq t$ be a prefix of t. A node $y \in \mathrm{dom}(t)$ is *off* s if $y \notin \mathrm{dom}(s)$ and either y is a root or the parent of y is in $\mathrm{dom}(s)$. 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 $a \in A$, abusing the notation, we denote by a the single-letter 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 us to build forest from basic elements. Following [8] we introduce the following operations on forests. For a graphical presentation of these operations see Figs. 1, 2, and 3 (compare Figures 1 and 2 in [8]).

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 $(pq)t = p(qt)$.

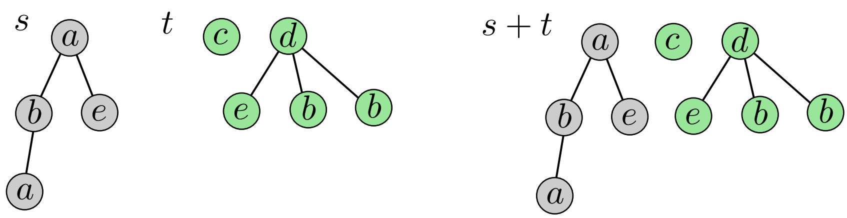


Fig. 1 Forest concatenation

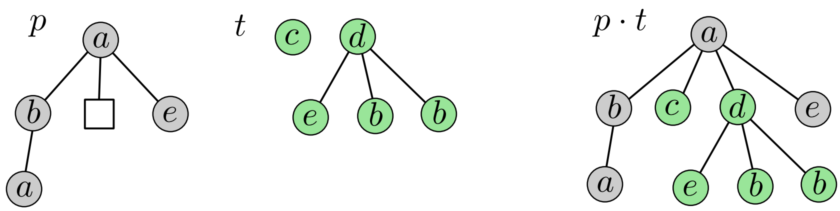


Fig. 2 Context composition

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

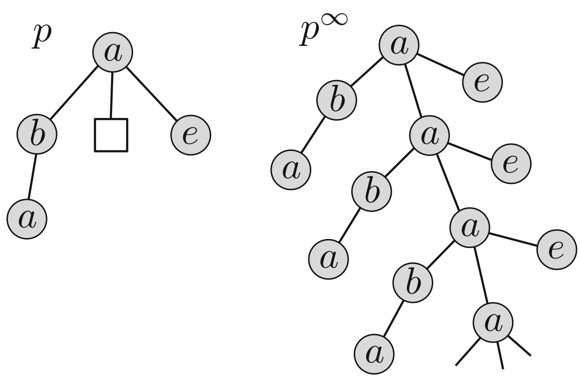
– we can compose a guarded context p with itself infinitely many times, which results in the forest p^∞ that satisfies $p(p^\infty) = p^\infty$. Note that we exclude non-guarded contexts from this definition. (For example the result of $(\square + a0)^\infty$, even if well-defined, is not finitely branching.)

2.2 Automata and Regular Languages

A (non-deterministic 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 an (i, j)-automaton) if i is the minimal and j is the maximal value of Ω on Q.

Fig. 3 Infinite composition



A *run* of an 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 condition means $(0, t(x), \rho(x)) \in \Delta$.

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.

We use MSO logic to describe properties of infinite forests. An infinite forest is treated as a relational structure, where the universe contains 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 weak MSO: the logic with the same syntax as MSO but with the semantic 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, weak MSO can be naturally embedded into MSO. There are examples of languages of infinite forests that are definable in MSO but not in weak MSO.

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

Theorem 1 ([11]) *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.*

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 forests with a fixed prefix r of some depth d: $\{t : t{\restriction}_{\omega^{\leq d}} = r\}$ for a finite forest 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 $\cup_{\beta<\alpha}\mathbf{\Pi}^0_\beta(X)$,
– $\mathbf{\Pi}^0_\alpha(X)$ be the class of countable intersections of sets from $\cup_{\beta<\alpha}\mathbf{\Sigma}^0_\beta(X)$.

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

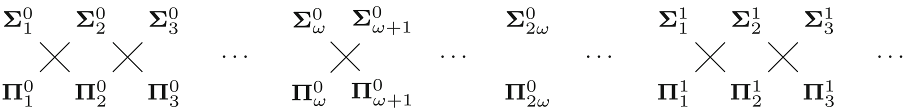


Fig. 4 The boldface hierarchy

[12, 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 $\mathbf{\Pi}^1_i$, for $i < \omega$.

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 its strictness: all the inclusions on Fig. 4 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.*

Proof The first statement follows by automata-theoretic techniques: a forest belongs to a regular language if there exists a run that is accepting on every branch. This statement is $\mathbf{\Sigma}^1_2$.

A forest is thin if and only if it does not contain a full binary subtree as a minor. This definition is a co-analytic definition of A^{ThinFor} among all forests.

For hardness we can use the implication (3) $\Rightarrow$ (4) in Theorem 46 — the language of all thin forests violates condition (15) so it is $\mathbf{\Pi}^1_1$-hard. It can also be proved directly by repeating the construction of the reduction f from Section 7.2. □

3 Ranks

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 definition of ranks we use is based on an appropriate notion of derivative: we inductively remove *simple* parts of a given forest. Depending on which forests are treated as simple, we obtain different ranks.

Definition 3 Let B be a set of thin trees. We say that B is *good as a rank basis* if it satisfies the following conditions for every tree t:

1. if t belongs to B, then all the subtrees of t also belong to B (i.e. B is closed under subtrees),
2. if no subtree of t belongs to B then t contains a branching node.

We use two families B giving rise to two ranks:

- Let B_P contain all trees containing one node and those trees that consists of exactly one infinite branch (i.e. without any finite branch),
- Let B_{CB} contain all trees containing only finitely many finite and infinite branches.

Note that both families B_P, B_{CB} are good as rank basis.

The definition of the derivative and rank on thin forests is an adopted version of the Cantor-Bendixson derivative on closed sets (see e.g. [12, Exercise 6.15 and Chapter IV Section 34.D]). In the case of B_{CB} it is in principle the same operation.

Consider the following operation on forests called *derivative*, parametrized by a set of thin trees B that is good as a rank basis.

Definition 4 For a forest $t \in A^{\mathsf{For}}$ we define the forest $\mathrm{Dv}_B(t) \subseteq t$ that contains only those nodes $x \in \mathrm{dom}(t)$ such that $t{\restriction}_x \notin B$.

We can iterate this derivative transfinitely many times, as expressed by the following definition.

Definition 5 Put $\mathrm{Dv}_B^0(t) = t$. Inductively define $\mathrm{Dv}_B^\eta(t)$ for any countable ordinal $\eta < \omega_1$. Let $\mathrm{Dv}_B^{\eta+1}(t) = \mathrm{Dv}_B(\mathrm{Dv}_B^\eta(t))$ and if η is a limit ordinal let

$$\mathrm{Dv}_B^\eta(t) = \bigcap_{\beta<\eta} \mathrm{Dv}_B^\beta(t),$$

where the intersection is set-theoretical — it restricts the set of nodes of a forest to the common fragment.

Fact 6 *Let $t \in A^{\mathsf{For}}$ be a forest. The sequence $\mathrm{Dv}_B^\eta(t)$ for $\eta < \omega_1$ is a decreasing sequence of forests. There exists $\eta_0 < \omega_1$ such that*

$$\mathrm{Dv}_B^{\eta_0}(t) = \mathrm{Dv}_B^{\eta_0+1}(t) = \mathrm{Dv}_B^{\eta_0+2}(t) = \dots.$$

The following proposition shows a connection of this iterated derivative and thin forests.

Proposition 7 *Let t be a forest and $\eta_0 < \omega_1$ be an ordinal such that $\mathrm{Dv}_B^{\eta_0}(t) = \mathrm{Dv}_B^{\eta_0+1}(t)$. The forest $\mathrm{Dv}_B^{\eta_0}(t)$ is the empty forest if and only if t is a thin forest.*

Proof Assume that $\mathrm{Dv}_B^{\eta_0}(t)$ is the empty forest. Observe that every application of the derivative decreases the number of branches of t by countably many: there are countably many nodes $x \in \mathrm{dom}(t)$ and the subtree under a removed node x belongs to the family B, therefore is thin. Since there are countably many applications of the derivative, the total number of removed branches is also countable.

Assume that $t' = \mathrm{Dv}_B^{\eta_0}(t)$ is not the empty forest. We show that in that case $t' \subseteq t$ has uncountably many branches. We construct an embedding of the complete binary tree into t (also called a Cantor scheme). Such an embedding maps finite sequences $b \in \{L, R\}^*$ into nodes $x_b \in \mathrm{dom}(t')$. We start with any $x_\epsilon \in \mathrm{dom}(t')$. Let $b \in \{L, R\}^*$ be a sequence such that the node $x_b \in \mathrm{dom}(t')$ is defined. Observe that there must be a branching node y of t' under x_b (since all the subtrees of $t'{\upharpoonright}_{x_b}$ do not belong to B and B is good as a rank basis). Put x_{bL}, x_{bR} as two arbitrary distinct children of y in t'.

The above definition gives us distinct infinite branches of t' for every $\pi \in \{L, R\}^\omega$. Therefore, t' has uncountably many infinite branches. So $t \notin A^{\mathsf{ThinFor}}$. □

Definition 8 Let $t \in A^{\mathsf{ThinFor}}$ be a thin forest and B be good as a rank basis. We define the B-rank of the forest t (denoted $r_B(t)$) as the smallest ordinal η_0 such that $\mathrm{Dv}_B^{\eta_0}(t) = 0$. We extend it to $\mathrm{r}_B(x, t)$ (the rank of x in t) for a node $x \in \mathrm{dom}(t)$ in such a way that $\mathrm{r}_B(x, t)$ is the least $\eta_0 < \omega_1$ such that $x \notin \mathrm{dom}\left(\mathrm{Dv}_B^{\eta_0}(t)\right)$.

Fact 9 *For every thin forest* $t \in A^{\mathsf{ThinFor}}$ *and node* $x \in \mathrm{dom}(t)$ *we have* $\mathrm{r}_B(x, t) = \mathrm{r}_B(t{\upharpoonright}_x)$.

If t is a non-empty thin forest and B is good as a rank basis then $\mathrm{r}_B(t)$ *is not a limit ordinal. In particular the ordinal* $\mathrm{r}_B(t) - 1$ *is defined. Also:*

$$\mathrm{Dv}_B^{\mathrm{r}_B(t)-1}(t) \text{ is a concatenation of finitely many trees in } B. \tag{1}$$

If $x \preceq y$ are two nodes of a thin forest t then $\mathrm{r}_B(x, t) \leq \mathrm{r}_B(y, t)$.

The observation that $\mathrm{r}_B(t)$ is not a limit ordinal follows from the fact that each forest has only finitely many roots. By the definition of the limit composition of Dv, for every $x \in \mathrm{dom}(t)$ we have $\mathrm{r}_B(x, t) < \eta$. In particular, if x is a root of t then $x \notin \mathrm{Dv}_B^{\eta_x}(t)$ for some $\eta_x < \eta$. Let η' be the supremum of η_x among the roots x of t. Since there are finitely many roots of t, $\eta' < \eta$. It means that already $\mathrm{Dv}_B^{\eta'}(t) = 0$. For a non-limit η we get (1).

Now we can fix our two derivatives: $\mathrm{Dv}_{CB} = \mathrm{Dv}_{B_{CB}}$, $\mathrm{Dv}_P = \mathrm{Dv}_{B_P}$, and ranks: $\mathrm{rank}^{\mathrm{CB}} = \mathrm{r}_{B_{CB}}$ and $\mathrm{rank} = \mathrm{r}_{B_P}$. $\mathrm{rank}^{\mathrm{CB}}$ is called the *Cantor-Bendixson rank* (shortly CB-rank).

Definition 10 For an ordinal $\eta < \omega_1$ we denote by $A^{\mathsf{ThinFor}\leq\eta}$ the set of thin forests of CB-rank at most η.

The crucial way of using ranks is induction: we can decompose a given forest as its core fragment and a number of trees connected to it. Since all those trees have smaller rank, we can assume the induction hypothesis about them. There are two notions of core fragments for our two ranks.

Definition 11 Let t be a nonempty thin forest. The *spine* of t is $\mathrm{Dv}_P^{\mathrm{rank}(t)-1}(t)$. The *final prefix* of t is $\mathrm{Dv}_{CB}^{\mathrm{rank}^{CB}(t)-1}(t)$.

Using (1) we get the following fact.

Fact 12 *Let t be a nonempty thin forest. The spine of t is of the form $t_1 + t_2 + \ldots + t_n$ for some trees $t_1, \ldots, t_n$ belonging to B_P — each t_i is either an one-node tree or an one-infinite-branch tree.*

The final prefix t' of t is a thin forest of CB-rank 1. Therefore, t' has the form of a finite forest r and finitely many infinite branches $\pi_1, \pi_2, \ldots, \pi_n$ starting from distinct leafs of r. If t is infinite then there are infinite branches in t' (i.e. $n > 0$).

Intuitively, a forest t has rank^{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 CB-rank 1 and for every x that is off s we have $\mathrm{rank}^{\mathrm{CB}}(t{\restriction_x}) \leq M$, then $\mathrm{rank}^{\mathrm{CB}}(t) \leq M + 1$.

Figure 5 presents a sequence of forests of increasing CB-rank. The leftmost branch of each forest is its final prefix. In the case of these forests the final prefix coincides with the spine. However, the two notions diverge in general. For instance, every non-empty finite forest coincides with its final prefix while its spine is of the form $t_1 + \ldots + t_n$ where each t_i is a one-node tree.

3.1 Skeletons

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.

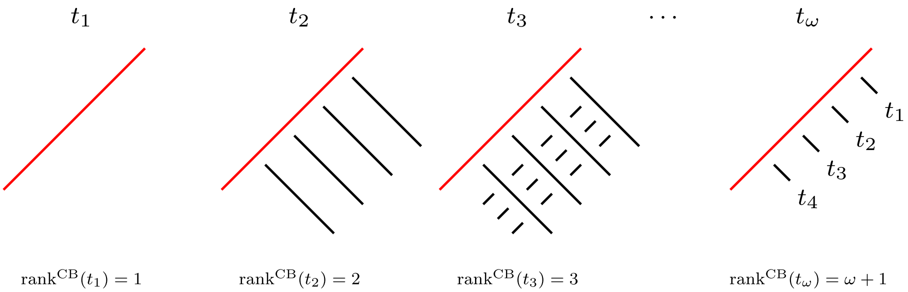


Fig. 5 Examples of trees of increasing CB-rank

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 all but finitely many 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 $(A \times \{0, 1\})^{\mathsf{For}}$.

Definition 13 Assume that (t, σ) is a forest with a skeleton. Take any node $x \in \mathrm{dom}(t)$. The branch π starting in x that *follows* at every point the skeleton σ is called the *main branch of σ from x*. More formally, it can be defined as the unique finite or infinite branch $\pi \in \omega^{\leq\omega}$ such that:

$$x \preceq \pi \wedge \forall_{y \preceq \pi}\ (y \preceq x \vee y \in \sigma).$$

Note that the main branch may be finite if it reaches a leaf of the forest. Otherwise it is infinite. By the assumption that a skeleton contains almost all nodes on every branch, we obtain the following fact.

Fact 14 *Take a forest $t \in A^{\mathsf{For}}$ with a skeleton σ and an infinite branch π of t. There exists a node $x \in t$ such that π is the main branch of σ from x.*

Proposition 15 *A given forest $t \in A^{\mathsf{For}}$ has a skeleton if and only if t is thin.*

Proof If a forest has a skeleton then by the above fact every infinite branch of t is from some point on its main branch (from some node of t). So there are at most countably many branches of t.

Now assume that a forest t is thin. Inductively on the rank of t we construct a skeleton of t. For a technical reason the constructed skeleton will not contain any root of the given forest. After the induction is performed, we can add one of the roots to σ.

If $t = 0$ then the empty set is its skeleton. Assume that $\mathrm{rank}(t) = \eta > 0$ and let $s = s_1 + s_2 + \ldots + s_n$ be the spine of t (see Definition 11). Let σ contain all non-root nodes of $s_1, s_1, \ldots, s_n$. Since all subtrees that are off s have smaller rank, we can inductively define σ on them. Finally, for every s_i that is a single node and not a leaf in t we add to σ the leftmost child of s_i.

First observe that σ defined this way contains exactly one node from each set of siblings. Let us take any infinite branch π of a thin forest t. Note that ranks of nodes along this branch are non-increasing, so from some point on they are all equal some ordinal η. Therefore at the η-th step of our induction one of the trees s_i had the form of one infinite branch containing almost all nodes along π. So, by the definition of σ, almost all nodes along π belong to σ. □

Definition 16 The skeleton σ constructed in the proof of Proposition 15 is called the *canonical skeleton for t* and is denoted by $\sigma(t)$.

4 Algebra

In this section we define two variants of thin-forest algebras. The operations and axioms of the first variant, *regular-thin-forest algebras*, are constructed in such a manner that the free object of this algebras is the set of all regular thin forests and regular thin contexts. Regular-thin-forest algebras form a common generalization of both Wilke algebras [26] and forest algebras [8].

The free object of the second variant, *unrestricted-thin-forest* algebras, is the set of all thin forests and thin contexts. Unrestricted-thin-forest algebras form a common generalization of both ω-semigroups and forest algebras.

A regular-thin-forest algebra is a three-sorted algebra $(H, V_+, V_\square, \mathit{act}, \mathit{in}_l, \mathit{in}_r, \mathit{inf})$. It consists of two monoids H and $V = V_+ \cup V_\square$ (partitioned into a subsemigroup V_+ and a submonoid $V_\square$) along with an operation of an action $\mathit{act} : V \times H \to H$ of V on H, two operations $\mathit{in}_l, \mathit{in}_r : H \to V_\square$, and an infinite loop operation $\mathit{inf} : V_+ \to H$. Instead of writing $\mathit{act}(v, h)$, we write vh. Instead of writing $\mathit{inf}(v)$, we write v^∞. We will call H the *horizontal monoid* and V the *vertical monoid*.

The above construction is based on forest algebras (see [8]). In fact we take a 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 $V_\square$ respectively.

Since the insertion operations are somewhat cumbersome to use, we will use the operation $+$ to concatenate forests with contexts, meaning $h + v = \mathit{in}_l(h)v$, $v + h = \mathit{in}_r(h)v$.

The definition of an unrestricted-thin-forest algebra is the same as that of a regular-thin-forest algebra, except that the infinite loop operation is replaced by an infinite product: $\pi : V_+^\infty \to H$.

4.1 Axioms

A regular-thin-forest algebra must satisfy the following axioms:

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

(A2) $(V, \cdot, \square)$ is a monoid with multiplication $\cdot$ and neutral element $\square$; it contains two disjoint subalgebras: $(V_\square, \cdot, \square)$ 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) $\mathit{in}_l(h)g = h + g$, $\mathit{in}_r(h)g = g + h$ for every $h, g \in H$,

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

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

An unrestricted-thin-forest algebra has its own versions of Axioms (A5) and (A6):

(A5′) for every $v \in V_+$ and for every sequence $\{v_n\}_{n\geq 0} \in V_+^\infty$,

$$v\pi(v_0, v_1, v_2, \ldots) = \pi(v, v_0, v_1, v_2, \ldots),$$

(A6′) for every increasing sequence $\{k_n\}_{n\geq 1}$ and each sequence $\{v_n\}_{n\geq 0} \in V_+^\infty$,

$$\pi(v_0 v_1 \cdots v_{k_1-1},\ v_{k_1} v_{k_1+1} \cdots v_{k_2-1},\ \ldots) = \pi(v_0, v_1, v_2, \ldots).$$

The infinite loop operation can be expressed as $v^\infty = \pi(v, v, v, \ldots)$.

4.2 The Free Objects

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

(a) the horizontal monoid is the set of regular thin forests over A, with the operation of forest concatenation;
(b) the vertical monoid is the set of regular thin contexts over A (respectively guarded and non-guarded), with the operation of context composition;
(c) the action is the operation of composing a context with a forest,
(d) 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);
(e) the infinite loop operation takes a regular thin context and transforms it into a regular thin forest by performing infinite composition.

In the same manner we define the *free unrestricted-thin-forest algebra* over A, which is denoted by $A^{\mathsf{Thin}\triangle}$, by above conditions (a)–(d) without the assumption of regularity and a condition

(e′) the infinite product operation takes an infinite sequence of thin contexts and transforms it into a thin forest by performing infinite composition.

Theorem 17 *The algebra $A^{\mathsf{regThin}\triangle}$ is a regular-thin-forest algebra. Moreover it is the free algebra (in the sense of universal algebra, see* [5]*) in the class of regular-thin-forest algebras over the generator set $A\square = \{a : a \in A\}$ — the set of single-letter tree contexts.*

Similarly, the algebra $A^{\mathsf{Thin}\triangle}$ is an unrestricted-thin-forest algebra and the free algebra in the class of unrestricted-thin-forest algebras over the generator set $A\square$.

Proof The proof is technical, but not surprising. It is divided into two parts: first we show that the free objects are generated by the alphabet $A\square$. Then we prove that if two terms generate the same object then the axioms imply that they are equivalent. See [11] for details. □

4.3 Correspondence Between Two Algebras

Theorem 18 *Every finite regular-thin-forest algebra can be equipped, in a unique way, with a structure of an unrestricted-thin-forest algebra.*

Proof Let $(H, V_+, V_\square)$ be a regular-thin-forest algebra. Consider a set $H_\omega \subseteq H$ which consists of all elements of form vw^∞ for $v, w \in V_+$. It is easy to see that

(V_+, H_ω) is a Wilke algebra. For a Wilke algebra we can, in a unique way, define the operation $\pi : V_+^\infty \to H_\omega$ such that (V_+, H_ω) is an ω-semigroup (see [17, Theorem 5.1]). We can naturally extend the definition of the operation π to $(H, V_+, V_\Box)$. Since the axioms of an ω-semigroup regarding π are the same as the axioms of an unrestricted-thin-forest algebra, we conclude that $(H, V_+, V_\Box)$ with the operation π is an unrestricted-thin-forest algebra.

The uniqueness of this extension follows from the fact that every extension must map elements of V_+ to some element from H_ω (due to the axioms and Ramsey theorem). Therefore, different extensions would differ on (V_+, H_ω) which is impossible, since (V_+, H_ω) is unique [25]. □

5 Recognizability by Thin-Forest Algebras

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

We will consider terms over the signature of thin-forest algebras 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 a 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 algebras of type τ_H with 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 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 $A^{\mathsf{Thin}\triangle}$ with respect to L-equivalence is the *syntactic unrestricted-thin-forest algebra* for L. The *syntactic morphism* of L assigns to every element of $A^{\mathsf{Thin}\triangle}$ its equivalence class in the syntactic unrestricted-thin-forest algebra of L. Similarly, in the case of the regular-thin-forest algebra.

The following notion is technically useful. We say that a thin-forest algebra is *faithful* if there are no two distinct elements $v, w \in V$ such that

- $vh = wh$ for all $h \in H$ and
- $(vu)^\infty = (wu)^\infty$ for all $u \in V$ such that $vu, wu \in V_+$.

Fact 19 *Every syntactic unrestricted-thin-forest algebra (respectively syntactic regular-thin-forest algebra) is faithful.*

Proof There are only two ways a tree-valued term can use a context-valued variable v: either by applying v to a tree-valued term or by using the infinite power. Therefore, if two elements $v, w \in V$ satisfy the conditions of faithfulness, then v and w are L-equivalent and, hence, equal. □

Theorem 20 *A language of thin forests is recognizable by a finite unrestricted-thin-forest algebra if and only if it is regular.*

Every regular language of thin forests is recognizable by its syntactic morphism.

The syntactic unrestricted-thin-algebra and the syntactic morphism can effectively be calculated, given a parity automaton recognising a given language.

The rest of this section is devoted to proving this theorem.

5.1 Automaton to Algebra

In this section we show how to calculate, given a non-deterministic forest automaton $\mathcal{A}$, an unrestricted-thin-forest algebra that recognizes the language recognized by $\mathcal{A}$. This algebra is called the *automaton algebra*.

Let us fix a non-deterministic forest automaton $\mathcal{A}$, with states Q, input alphabet A, priorities $\{0, \ldots, k\}$, and a set of initial states $Q_I \subseteq Q$. Below we describe the automaton algebra (H, V), together with associated morphism $\alpha : A^{\mathsf{Thin}\triangle} \to (H, V)$, which recognizes the language $\mathrm{L}(\mathcal{A})$.

Before describing the algebra itself, we define the morphism α. This morphism should explain the intended meanings of H and V.

(a) The morphism α associates to each thin forest t a subset of Q. A state q belongs to $\alpha(t)$ if some accepting run ρ over t has value q.
(b) The morphism α associates to each thin context p a subset of the product $Q \times \{0, \ldots, k\} \times Q$. A triple (q_1, i, q_2) belongs to $\alpha(p)$ if there exists a thin forest s and an accepting run ρ over ps such that the value of ps in ρ is q_2, the value of s in ρ is q_1, and the highest priority assigned to nodes that are ancestors of the hole in p is i (this priority is equal to 0 if p is non-guarded).

Therefore, the carriers of the horizontal and vertical monoids are subsets

$$H \subseteq P(Q), \qquad V \subseteq P(Q \times \{0, \ldots, k\} \times Q),$$

which are images under α of thin forests and thin contexts, respectively. These might be proper subsets, for instance not every subset of Q needs to be an image $\alpha(t)$. A thin forest belongs to L if and only if its image under α contains a state from Q_I.

We say that two thin forests s, t are *automaton-equivalent* if the subsets associated to these forests by the morphism α are the same. We denote it by $s \sim_{\mathcal{A}} t$. Similarly we define automaton-equivalence of thin contexts.

Lemma 21 *The relation of automaton-equivalence $\sim_{\mathcal{A}}$ is a congruence with respect to the operations of the free thin-forest algebra.*

Proof We show the claim for forest concatenation and for infinite loop operation. The proof for other operations follows along the same lines.

Let t, t', s be thin forests and $t \sim_{\mathcal{A}} t'$. We must show that $t+s \sim_{\mathcal{A}} t'+s$. Suppose that $q \in \alpha(t + s)$. Thus there is an accepting run ρ over $t + s$ such that the value of t is q', the value of s is q'' in ρ, and $q = q' + q''$. The run ρ is accepting over the forest

t, and since $t \sim_{\mathcal{A}} t'$, there is an accepting run ρ' over t' with value q'. Combining the run ρ' over t' with the run ρ over s we get an accepting run over $t' + s$ of value $q = q' + q''$. Thus $q \in \alpha(t' + s)$.

Let p, p' be guarded thin contexts and $p \sim_{\mathcal{A}} p'$. We must show that $p^{\infty} \sim_{\mathcal{A}} p'^{\infty}$. Suppose that $q \in \alpha(p^{\infty})$. Thus there is an accepting run ρ over the forest p^{∞} with value q. For $i \geq 1$ we denote by q_{i-1} the sum of states assigned to the roots of the i-th (counting from the top) instance of the context p (of course $q = q_0$), and by k_i the highest priority assigned to nodes on the path to the i-th hole. Thus for every $i \geq 1$ we have $(q_{i-1}, k_i, q_i) \in \alpha(p)$, and therefore $(q_{i-1}, k_i, q_i) \in \alpha(p')$. That means that for every i there is an accepting run ρ_i of value q_{i-1} over $p's_i$ for some forest s_i evaluates to q_i. Combining these runs we get that $q \in \alpha(p'^{\infty})$. □

The following fact is a direct consequence of Lemma 21 by a standard method of universal algebra.

Fact 22 *The function α induces a structure of an unrestricted-thin-forest algebra on the sets (H, V) in such a way that $\alpha : A^{\mathsf{Thin}\triangle} \to (H, V)$ is a homomorphism.*

Lemma 23 *The morphism α recognizes the language* $\mathrm{L}(\mathcal{A})$.

Proof Let $I = \{h \in H : Q_I \cap h \neq \emptyset\}$. From the definition we have that a forest t is in L if some accepting run over t has a value from Q_I. This is equivalent to saying that $Q_I \cap \alpha(t) \neq \emptyset$, thus $\alpha(t) \in I$, and $t \in \alpha^{-1}(I)$. Therefore $\mathrm{L}(\mathcal{A}) = \alpha^{-1}(I)$. □

Now we show how to effectively calculate the operations of finite arity of the automaton algebra. Defining the operations is straightforward, keeping in mind the intended meaning of the morphism α. We denote by $TC(v)$ the transitive closure of v with respect to the $\cdot$ operation. Formally $(p, \alpha, q) \in TC(v)$ if there exist a sequence of states $p = q_n, q_{n-1}, \ldots, q_0 = q$ and priorities $\alpha_n, \alpha_{n-1}, \ldots, \alpha_1$ such that $\alpha = \max\{\alpha_n, \ldots, \alpha_1\}$ and $(q_i, \alpha_i, q_{i-1}) \in v$ for every $1 \leq i \leq n$.

The operations are as follows (by $(p, \_, q)$ we denote an arbitrary triple of the form (p, j, q) for some j):

$$\begin{aligned}
h + g &= \{p + q \mid p \in h, q \in g\} && \text{for } h, g \in H\\
vw &= \{(p, \max(i, j), q) \mid (p, i, r) \in w, (r, j, q) \in v\} && \text{for } v, w \in V\\
v^{\infty} &= \{q \mid (p, i, p), (p, _, q) \in TC(v), i \text{ is even}\} && \text{for } v \in V_{+}\\
vh &= \{q \mid p \in h, (p, _, q) \in v\} && \text{for } v \in V,\ h \in H\\
in_l(h) &= \{(q, 0, p + q) \mid p \in h\} && \text{for } h \in H\\
in_r(h) &= \{(q, 0, q + p) \mid p \in h\} && \text{for } h \in H
\end{aligned}$$

Finally, we also define the morphism:

$$\begin{aligned}
\alpha(a) &= \{(q, \Omega(p), p) \mid (q, a, p) \in \Delta\} \qquad \text{for } a \in A,\\
\alpha(\square) = \square &= \{(p, 0, p) \mid p \in Q\},\\
\alpha(0) = 0 &= \{0\}
\end{aligned}$$

The proof of correctness of the above operations mimics the reasoning in Lemma 21.

Finally, to calculate the syntactic algebra of L, we can first calculate the automaton algebra for $\mathcal{A}$, and then calculate the L-equivalence relation $\sim_L$ over (H, V) using the idea from Moore's algorithm for minimizing automata. First we put $h \not\sim_L g$ for every $h, g \in H$ such that exactly one of the types h, g belongs to $I \subseteq H$. Then we try to extend the number of non-L-equivalent pairs of elements using every operation. For example for forest concatenation and for infinite loop we do:

– if there are elements $h, h', g \in H$ such that $h + g \not\sim_L h' + g$ or $g + h \not\sim_L g + h'$, then $h \not\sim_L h'$,
– if there are elements $v, v' \in V_+$ such that $v^\infty \not\sim_L v'^\infty$, then $v \not\sim_L v'$.

We terminate the algorithm when there is no new pair we can add. Note that we do not take into account the operation π. Still it is correct due to the fact that minimizing an unrestricted-thin-forest algebra is the same as minimizing a regular-thin-forest-algebra:

Theorem 24 *Let L be a regular language of thin forests with the syntactic unrestricted-thin-forest algebra* synt(L). *Let* synt(L_R) *be the syntactic regular-thin-forest algebra of the language L_R which contains all regular thin forests from L. Then* synt(L) *is isomorphic to the extension of* synt(L_R) *defined in Theorem 18.*

Proof We prove that synt(L) is an extension of synt(L_R). Isomorphism follows from Theorem 18 which states that such an extension is unique.

Therefore we must show that for every two regular thin forests s and t which are equivalent under L-equivalence relation $\sim_L$, they are also equivalent under $\sim_{L_R}$. From the definition $s \sim_L t$ means that for every term σ from the signature of unrestricted-thin-forest algebra we have $\sigma[x \leftarrow s] \in L$ if and only if $\sigma[x \leftarrow t] \in L$. This is equivalent to saying that for every thin forest u over the alphabet $A \cup \{x\}$ we have $u[x \leftarrow s] \in L$ if and only if $u[x \leftarrow t] \in L$. Finally, this is equivalent to checking that two inverse images $(x \leftarrow s)^{-1}(L)$ and $(x \leftarrow t)^{-1}(L)$ are equal.

These images are regular languages, thus checking their equality is equivalent to testing whether they contain the same regular forests. Since regular thin forests are generated by terms of regular-thin-forest algebra, this is equivalent to stating that for every term σ from the signature of regular-thin-forest algebras $\sigma[x \leftarrow s] \in L$ if and only if $\sigma[x \leftarrow t] \in L$, thus $s \sim_{L_R} t$. □

5.2 Algebra to (1, 3)-Automaton

Let L be a regular language of thin forests, (H, V) the syntactic unrestricted-thin-forest algebra of L, and $\alpha : A^{\mathsf{Thin}\triangle} \to (H, V)$ the syntactic morphism of L. We will construct a forest $(1, 3)$-automaton $\mathcal{A}$ recognizing L. Let the set of states of $\mathcal{A}$ be

$$Q = H^3 \cup Q_\sigma \cup q_\perp \qquad \text{where} \qquad Q_\sigma = H^3 \times V \times (V \cup \{\star\}).$$

The main idea is that the automaton $\mathcal{A}$ will (among other things) guess a skeleton σ of the forest t. The nodes in σ are precisely those which will be assigned a state from Q_σ. The state $q_\perp$ is the "error" state. Intuitively, a state of the form $(\overline{h}, e, u)$ indicates that the main branch of the guessed skeleton that starts from the current node can be decomposed into a sequence of contexts of types $u, e, e, \ldots$ In the case $u = \star$ the decomposition has the form $e, e, \ldots$

We will use the notation $\overline{h} = (h', h, h'')$ for $\overline{h} \in H^3$. The idea is that if a node x is assigned a state $\overline{h}$ or a state from $\{\overline{h}\} \times V \times (V \cup \{\star\}) \subseteq Q_\sigma$, then the type of the subtree rooted at the node x is h (i.e. $\alpha(t\restriction_x) = h$), and the type of the subforest rooted in the siblings of x which lie to the left (respectively to the right) of x is h' (respectively h'').

First we define a monoid operation on Q. If $\overline{h}_1, \overline{h}_2 \in H^3$ then the result is from $H^3 \cup \{q_\perp\}$:

$$\overline{h}_1 + \overline{h}_2 = \begin{cases} (h'_1, h_1 + h_2, h''_2) & \text{if } h'_1 + h_1 = h'_2 \text{ and } h''_1 = h_2 + h''_2, \\ q_\perp & \text{otherwise.} \end{cases}$$

If one argument is from H^3 and another from Q_σ then the result is from Q_σ:

$$\overline{h}_1 + (\overline{h}_2, e, u) = (\overline{h}_1 + \overline{h}_2, e, u), \tag{2}$$

$$(\overline{h}_1, e, u) + \overline{h}_2 = (\overline{h}_1 + \overline{h}_2, e, u), \tag{3}$$

if $\overline{h}_1 + \overline{h}_2 \neq q_\perp$, or $q_\perp$ otherwise.

Finally, if both arguments are from Q_σ or at least one is $q_\perp$, then the result is $q_\perp$.

Now we define the transition relation Δ:

$$(\overline{h}, a, \overline{h}_1) \in \Delta \quad \text{iff} \quad h' = h'' = 0 \text{ and } \alpha(a) \cdot h = h_1. \tag{4}$$

$$((\overline{h}, e, \star), a, \overline{h}_1) \in \Delta \quad \text{iff} \quad (\overline{h}, a, \overline{h}_1) \in \Delta \text{ and } h = e^\infty.$$

$$((\overline{h}, e, u), a, (\overline{h}_1, e, u_1)) \in \Delta \quad \text{iff} \quad (\overline{h}, a, \overline{h}_1) \in \Delta$$

$$\text{and} \begin{cases} u(h'_1 + \alpha(a) + h''_1) = u_1 & \text{if } u, u_1 \in V, \\ h'_1 + \alpha(a) + h''_1 = u_1 & \text{if } u = \star, u_1 \in V, \\ u(h'_1 + \alpha(a) + h''_1) = e & \text{if } u \in V, u_1 = \star, \\ h'_1 + \alpha(a) + h''_1 = e & \text{if } u = u_1 = \star. \end{cases}$$

$$(q_\perp, a, q_\perp) \in \Delta.$$

Finally we define priorities Ω:

$$\Omega(q) = \begin{cases} 3 & \text{if } q \in H^3, \\ 2 & \text{if } q \in H^3 \times V \times \{\star\}, \\ 1 & \text{otherwise.} \end{cases}$$

The initial states of $\mathcal{A}$ are precisely those triples $(0, h, 0) \in H^3$ such that $\alpha^{-1}(h) \subseteq L$.

Lemma 25 *The language accepted by the automaton $\mathcal{A}$ equals L.*

Proof First, we show that a forest t has an accepting run ρ with a value different from $q_\perp$ if and only if it is thin.

Suppose that t has an accepting run ρ. First we show that the set $\sigma \subseteq \mathrm{dom}(t)$ defined as the set of nodes assigned a state in Q_σ in ρ is in fact a skeleton of t. If at least one node is assigned $q_\perp$, then the "error" state propagates upwards and the value of ρ is $q_\perp$. From the acceptance condition the maximum priority which appears infinitely often on each path must be 2. Thus priority 3 can appear only finitely often, thus there is only finitely many nodes marked by a state from H^3, thus on every path there is only finitely many nodes outside σ. Since $Q_\sigma + Q_\sigma = \{q_\perp\}$, at most one sibling is in σ. Since there is no transition in Δ of the form $H^3 \times A \times Q_\sigma$, every node in σ has a child from σ. Therefore all conditions for σ are satisfied and t is thin.

Suppose now that t is thin. Denote $Q_h = (H \times \{h\} \times H) \cup (H \times \{h\} \times H \times V \times (V \cup \{\star\}))$. We prove by induction over the rank of the nodes that if a node x is assigned a state from Q_h then $\alpha(t\restriction_x) = h$.

If all successors $x_1, \ldots, x_n$ of x have smaller ranks than rank(x), then from the inductive assumption x_i is assigned a state from Q_{h_i} where $h_i = \alpha(t\restriction_{x_i})$. Then from (2) we get that the sum of states assigned to these successors is from $Q_{h_1+\cdots+h_n}$. Thus by (4) x is assigned a state from $Q_{\alpha(a)(h_1+\cdots+h_n)}$, where a is the label of x.

Assume otherwise that there is an infinite path $\pi = x_0x_1x_2\ldots$ from x of nodes which have the same rank as rank(x). Every successor y of x_i which does not belong to π has smaller rank than rank(x), thus from the induction assumption $\rho(y) \in Q_h$ if and only if $\alpha(t\restriction_y) = h$. Let p_i denote the context which comes after putting a hole instead of x_{i+1} in $t\restriction_{x_i}$. We must ensure that $\alpha(p_0p_1\cdots) = \alpha(t\restriction_{x_0})$.

By Ramsey's theorem there are $u, e \in V_+$ and a partition

$$(p_0p_1\cdots p_{k_0-1})(p_{k_0}\cdots p_{k_1-1})(p_{k_1}\cdots p_{k_2-1})\cdots$$

such that $p_0p_1\cdots p_{k_0-1} = u$ and $p_{k_i}\cdots p_{k_{i+1}-1} = e$ for all $i \geq 0$. The transition relation over Q_σ is devised to guess the values of u and e and the partition. Let x_i be assigned a state $(\overline{h_i}, e, u_i)$. A block $p_{k_i}\cdots p_{k_{i+1}-1}$ of the partition is encoded by $u_{k_i} = \star$ and $u_j = \alpha(p_jp_{j+1}\cdots p_{k_{i+1}-1})$. Since there is an infinite number of encoded blocks, on every path there must be an infinite number of states with priority 2. Finally, the transitions ensure that $\alpha(t\restriction_{x_0}) = ue^\infty$.

Therefore, from the assumption that t is thin we conclude that there is an accepting run on t such that the sum of states assigned to the roots of t is $\alpha(t)$. Thus t is accepted by $\mathcal{A}$ if and only if $t \in L$. □

Therefore, we obtain the following theorem.

Theorem 26 *Every regular language of thin forests can be recognized among all forests by a non-deterministic* $(1, 3)$*-automaton.*

6 Applications of Thin-Forest Algebras

In this section we show how thin-forest algebras can be used to give decidable characterizations of certain classes 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 regular-thin-forest algebras 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. 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 [17] (the number k is not unique but the value v^k is).

Since (thanks to Theorem 24) checking an identity in the syntactic unrestricted-thin-forest algebra of a regular thin-forest language L is equivalent to checking it in the syntactic regular-thin-forest algebra of L, we will shortly write that we check identities in "the syntactic thin-forest algebra" of L.

In the following subsections we show how to decide whether a given regular language of thin forests is commutative, open in the standard topology, and definable by a formula of the temporal logic EF. We start by providing a tool used in the proceeding characterizations.

6.1 Components in a Forest

Our proofs use induction over the number of components in a regular thin forest. In this subsection we give the definition of a component.

Let t be a forest. We say that two nodes x, y of the forest are *in the same component* if the subtree $t{\restriction}_x$ is a subtree of the subtree $t{\restriction}_y$ and vice versa.

To a regular forest we associate a directed graph $G_t = (V_t, E_t)$ (we call it the *component graph* of the forest) in which the set of nodes V_t contains all non-isomorphic subtrees of t and there is an edge $(t_1, t_2) \in E_t$ if the subtree t_2 is an immediate subtree of the subtree t_1 (i.e. $t_2 = t_1{\restriction}_x$ for some child x of the root of t_1). The graph G_t is finite if and only if the forest t is regular. Every component in t corresponds to a strongly connected component in G_t.

There are two kinds of components: *singleton components*, which correspond to strongly connected components in G_t of exactly one node and no edges, and *connected components*, which correspond to other strongly connected components in G_t. Note that a node x in the forest is in a singleton component if and only if $t{\restriction}_x$ is not a proper subtree of $t{\restriction}_x$.

A component is a *root component* if it contains a root of the forest.

In Fig. 6 there is a tree t and the corresponding graph G_t. The tree has five components: two connected ones (which correspond to strongly connected components c_1, c_2 in G_t) and three singleton ones (which correspond to s_1, s_2, s_3). Note that the component which corresponds to a strongly connected component c_1 of one node but with a loop edge is in fact connected. Note that the graph loses some information, so it is not possible to fully reconstruct the forest t from G_t. However, it is only a matter of adding the order and multiplicity to edges of G_t.

Lemma 27 *In a thin regular forest t every connected component corresponds to a strongly connected component in G_t which is a simple cycle, i.e. the graph induced by the nodes of this component is a simple cycle.*

Proof Let c be the strongly connected component in G_t which corresponds to a connected component in t. Let G' be the graph induced by the nodes of c.

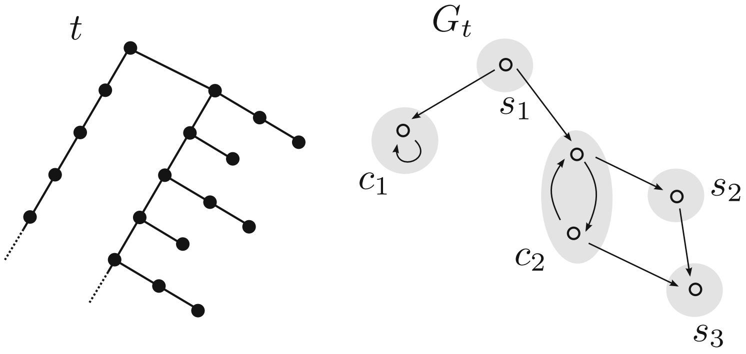


Fig. 6 A tree t with the corresponding graph G_t

We first show that the out-degree of every node in G' is at most 1. Let us assume otherwise — then there is a node u with at least two outgoing edges $u \to v_1, u \to v_2$. Adding a path from v_1 and v_2 back to u we get a full binary tree that is a minor of t, thus the forest is not thin.

Similarly we show that the in-degree of every node in G' is at most 1. Since c does not contain any isolated nodes, the out-degree and in-degree of any node is in fact exactly 1. Since c is connected, it is indeed a simple cycle. □

6.2 Commutative Languages

The notion of a *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

$$h + g = g + h \qquad \text{for } g, h \in H. \tag{5}$$

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 believe that the last (unrestricted) definition is the most appealing. However, it is not captured by the identity (5). Consider the language $L =$ "every node has 0 or 2 children and every branch goes left only a finite number of times". The language L does satisfy (5), but it is not commutative, as witnessed by two thin forests $a(a0 + a)^\infty \in L$, $a(a + a0)^\infty \notin L$.

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

$$h + v = v + h \qquad \textit{for } h \in H \textit{ and } v \in V. \tag{6}$$

Identity (5) corresponds to a weaker notion of commutativity (see [11]), where on every branch we allow only a finite number of rearrangements of siblings.

The rest of this subsection is devoted to proving Theorem 28. We start by formalizing the definition of a commutative language. We say that two forests t_0, t_1 are *commutatively equivalent* (we denote it by $t_0 \sim_C t_1$) if there exists a bijection $f : \mathrm{dom}(t_0) \to \mathrm{dom}(t_1)$ such that for every $x, y \in \mathrm{dom}(t_0)$:

(a) the nodes x and $f(x)$ have the same labels,
(b) the node x is a parent of y if and only if $f(x)$ is a parent of $f(y)$.

Note that condition (b) implies that the node x is a root if and only if $f(x)$ is a root. Observe that for any node $x \in \mathrm{dom}(t_0)$ the trees $t_0\restriction_x$ and $t_1\restriction_{f(x)}$ are commutatively equivalent.

A forest language L is called *commutative* if for every two forests t_0, t_1 which are commutatively equivalent, either both t_0, t_1 belong to L or none of them.

The definition of commutativity could be rephrased also in the language of games. We define a game, called the *commutative game*, which is used to test the commutative equivalence of two forests.

Let t_0, t_1 be two forests. The commutative game over t_0 and t_1, denoted by $\mathrm{G}_{\mathrm{comm}}(t_0, t_1)$, is played by two players: Spoiler and Duplicator. For convenience we add an auxiliary root node at the top of the forest t_i, which results in a tree t'_i.

The game proceeds in rounds. Each state of the game is a pair (x_0, x_1), which means that there is a pebble in a node $x_i \in \mathrm{dom}(t'_i)$. Initially both pebbles are in the roots of the trees t'_0, t'_1. A round is played as follows. If the number of children of node x_0 is different from the number of children of node x_1 then Spoiler wins the whole game. Otherwise Duplicator chooses a bijection f which maps the children of x_0 to the children of x_1.

Now Spoiler moves the pebble x_0 to a child x of x_0 and the pebble x_1 to the child $f(x)$ of x_1. If the labels of nodes x and $f(x)$ are different then Spoiler wins. Otherwise, the round is finished and a new round is played with the state updated to $(x, f(x))$.

It is easy to see that two forests t_0, t_1 are commutatively equivalent if and only if Duplicator can survive for infinitely many rounds in the commutative game $\mathrm{G}_{\mathrm{comm}}(t_0, t_1)$.

Lemma 29 *Let σ be a forest-valued term with one forest-valued variable over the signature of regular-thin-forest algebras and let s, t be thin forests. If Duplicator wins the commutative game $\mathrm{G}_{\mathrm{comm}}(s, t)$ then he also wins the commutative game $\mathrm{G}_{\mathrm{comm}}(\sigma[x \leftarrow s], \sigma[x \leftarrow t])$.*

Proof The strategy of Duplicator is very simple. As long as the children of nodes with pebbles are in σ, Duplicator chooses the identity bijection. Otherwise he uses the strategy from the game $\mathrm{G}_{\mathrm{comm}}(s, t)$. □

First, note the following easy fact.

Fact 30 *Let t_0 and t_1 be two thin trees which are commutatively equivalent. Then* $\operatorname{rank}(t_0) = \operatorname{rank}(t_1)$.

Lemma 31 *Suppose that identity (6) holds. If two thin forests t_0, t_1 are commutatively equivalent, then $\alpha(t_0) = \alpha(t_1)$.*

Proof We prove the lemma for trees, the generalization for forests is straightforward. The proof is by induction on the rank of the trees.

First, observe that from Fact 30, $\operatorname{rank}(t_0) = \operatorname{rank}(t_1)$. From the same argument, the spines of the trees have the same number of nodes (i.e. the same length). Suppose that they are infinite, the remaining case is similar.

Let $x_1^i, x_2^i, x_3^i, \ldots$ be the nodes on the spine of t_i which give us a decomposition $t_i = p_1^i p_2^i p_3^i \ldots$, where p_j^i is a context with a root in x_j^i and a hole in x_{j+1}^i.

Let $f : \operatorname{dom}(t_0) \to \operatorname{dom}(t_1)$ be a bijection which witnesses that $t_0 \sim_C t_1$. Again from Fact 30, $f(x_j^0) = x_j^1$ for all j.

Let T_j^i be the multiset of trees rooted in the children of x_j^i, but not in x_{j+1}^i. Abusing the notation slightly, we see that mapping f gives a natural bijection between T_j^0 and T_j^1, such that for any $s \in T_j^0$, the trees s and $f(s)$ are commutatively equivalent. Since trees from the sets T_j^i have ranks smaller than $\operatorname{rank}(t_0)$, we can use the inductive assumption to get that $\alpha(s) = \alpha(f(s))$ for every $s \in T_j^0$. Thus from (6) we have $\alpha(p_j^0) = \alpha(p_j^1)$ for all j. Therefore we get that $\alpha(t_0) = \alpha(t_1)$. □

Proof of Theorem 28 The “if” part of the theorem follows directly from Lemma 31.

The “only if” part is standard: Suppose for a start that instead of (6) we want to show that the simpler identity (5) is satisfied. By unravelling the definition of the syntactic algebra we need to show that for any forest-valued term σ of one forest-valued variable x and any thin forests t, s we have

$$\sigma[x \leftarrow t + s] \in L \quad \text{iff} \quad \sigma[x \leftarrow s + t] \in L. \tag{7}$$

It is easy to see that Duplicator wins the commutative game on forests $t+s$ and $s+t$, thus from Lemma 29 he wins the commutative game on forests $\sigma[x \leftarrow t + s]$ and $\sigma[x \leftarrow s+t]$. Therefore we get (7) from the fact that the language L is commutative.

To show that (6) is satisfied, we use the faithfulness of the syntactic thin-forest algebra and we show that the algebra satisfies the identities

$$h + vg = vg + h, \qquad \text{for } v \in V_+,\ h, g \in H,$$

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

Again, this boils down to showing that Duplicator wins the commutative game on forests $t + s$ and $s + t$ for any thin forests s, t as well as on forests $(p + t)^\infty$ and $(t + p)^\infty$ for any thin forest t and thin guarded context p. □

6.3 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 32 *A regular language of thin forests L is open if and only if its syntactic morphism* $\alpha\colon A^{\mathsf{Thin}\triangle} \to (H, V)$ *satisfies the following condition for* $v \in V_+$ *and* $h \in H$*:*

$$\textit{if} \quad v^\infty \in \alpha(L) \quad \textit{then} \quad v^\omega h \in \alpha(L). \tag{8}$$

The notion of an open set 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 [17]) to thin-forest algebras. Then the above condition could be simply stated as $v^\infty \geq v^\omega h$.

Let X be an infinite set of variable names. A *thin multicontext* over A is a thin forest over $A \cup X$ in which every variable $x \in X$ appears in a leaf. The number of variables appearing in a thin multicontext is not restricted. An *open thin multicontext* over A is a thin context p such that $p0$ is a thin multicontext. For an (open) thin multicontext p we denote by $\mathrm{vars}(p) \subseteq X$ the set of variables appearing in p.

Let p be a (open) thin multicontext and $\zeta\colon \mathrm{vars}(p) \to A^{\mathsf{ThinFor}}$ be a mapping which assigns thin forests to variables appearing in p. We denote by $p[\zeta]$ the forest (context) which results from replacing every variable x in p by the forest $\zeta(x)$. We say then that p is a *cut-off* of $p[\zeta]$.

By pA^{ThinFor} we denote a language of all thin forests (contexts) such that p is their cut-off.

For any forest s, the composition ps, for p from Fig. 7, is the thin multicontext such that

$$psA^{\mathsf{ThinFor}} = \{a(b0 + t_1 + a(s + t_2)) : t_1, t_2 \in A^{\mathsf{ThinFor}}\}.$$

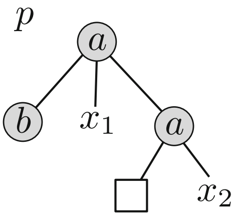


Fig. 7 An open thin multicontext p with the set of variables $\mathrm{vars}(p) = \{x_1, x_2\}$

By the definition of open sets, L is open if there exists a (possibly infinite) set P of finite thin multicontexts such that

$$L = \bigcup_{p \in P} pA^{\mathsf{ThinFor}}.$$

Proof of Theorem 32 It is obvious that if L is open then it must satisfy (8). Indeed, let $v \in V_+$ and let $t \in L$ be a thin forest of the form $t = r^\infty$ for some context $r \in \alpha^{-1}(v)$. In that case $\alpha(t) = v^\infty$. Since L is open, then there exists a cut-off p (of depth n) of the forest t such that $pA^{\mathsf{ThinFor}} \subseteq L$. Thus $r^k A^{\mathsf{ThinFor}} \subseteq L$ for any $k \geq n$. For $k = n \cdot |V|!$, we have $v^k = v^\omega$ for all $v \in V$. Since $r^k s \in L$ for every thin forest s, then for $h = \alpha(s)$, we have $v^k h = v^\omega h \in \alpha(L)$.

The converse implication will follow from Lemma 35, which is formulated at the end of the section. □

Let p, p' be two thin multicontexts. We say that p can be *immediately reduced* to p' if

$$p = qr^\infty \qquad \text{and} \qquad p' = qr^\omega x_\star$$

for an open thin multicontext q, a thin context r, and a variable $x_\star \notin \mathrm{vars}(q)$. We denote this fact by $p \to p'$ (see Fig. 8). We say that p can be *reduced* to p' if there is a sequence $p = p_0, p_1, p_2, \ldots, p_{n-1}, p_n = p'$ of thin multicontexts such that p_i can be immediately reduced to p_{i+1}. We denote this fact by $p \to^* p'$.

Lemma 33 *Let L be a regular language of thin forests which satisfies* (8) *and let p, p' be two thin multicontexts. If $pA^{\mathsf{ThinFor}} \subseteq L$ and $p \to p'$, then $p'A^{\mathsf{ThinFor}} \subseteq L$.*

Proof Let $p = qr^\infty$ and $p' = qr^\omega x_\star$ where q is an open thin multicontext, r is a thin context, and $x_\star$ is a variable not in $\mathrm{vars}(q)$. Observe that all the variables appearing

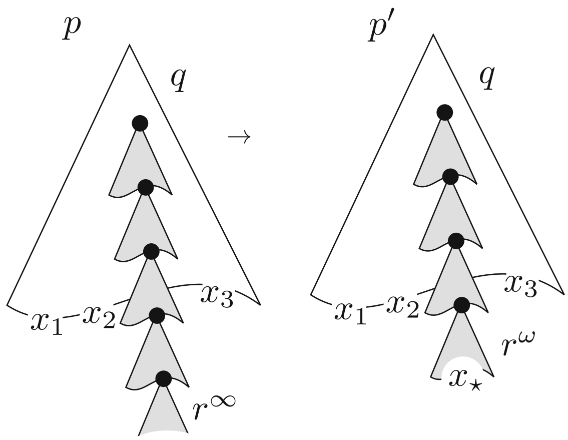


Fig. 8 Thin multicontext $p = qr^\infty$ can be immediately reduced to thin multicontext $p' = qr^\omega x_\star$

in p are from q. Similarly all the variables appearing in p' (except for the additional variable $x_\star$) are also in q.

Let t' be any forest from $p'A^{\mathsf{ThinFor}}$ and $\zeta : \mathrm{vars}(p') \to A^{\mathsf{ThinFor}}$ satisfies $p'[\zeta] = t'$. Applying ζ to thin multicontext p we get a forest $t = p[\zeta] \in pA^{\mathsf{ThinFor}}$. Since $pA^{\mathsf{ThinFor}} \subseteq L$ we get that the forest $t = q[\zeta]r^\infty$ is in L. From (8) the tree $t' = q[\zeta]r^\omega\zeta(x_\star)$ is also in L. Therefore $p'A^{\mathsf{ThinFor}} \subseteq L$. □

Let P' and P'' be the following two sets of cut-offs:

$$P' = \{\text{finite thin multicontext } p \mid pA^{\mathsf{ThinFor}} \subseteq L\},$$
$$P'' = \{\text{finite thin multicontext } p \mid t \to^* p \text{ for some } t \in L\}.$$

Lemma 34 *Let L be a regular language of thin forests. For every regular thin forest $t \in L$ there is a finite thin multicontext $p \in P''$ which is a cut-off of t.*

Proof Let $t \in L$. We prove the lemma by induction on the number of components in the forest t, i.e. we prove the statement: if s is a subforest of t then there is a finite thin multicontext p such that $s \to^* p$.

We can assume that t is a tree, otherwise we just concatenate prefixes for the trees which are rooted in the roots of a forest t.

If the root component of the tree t is a singleton component then $t = as$ for some $a \in A$ and a forest s. From the inductive assumption there is a finite thin multicontext p such that $s \to^* p$. Clearly the thin multicontext ap satisfies $t \to^* ap$.

Let the root component of the tree t be connected. Thus $t = (a_1q_1 \cdots a_nq_n)^\infty$ for some labels $a_1, \ldots, a_n \in A$ and non-guarded contexts $q_1, \ldots, q_n$. It is easy to see that for a variable $x_\star$

$$t \to (a_1q_1 \cdots a_nq_n)^\omega x_\star.$$

Let $q_i = t_i' + \square + t_i''$ for some forests t_i', t_i''. From the inductive assumption there are finite thin multicontexts p_i', p_i'' such that $t_i' \to^* p_i'$ and $t_i'' \to^* p_i''$. Without loss of generality we can assume that these thin multicontexts have different variables appearing in them, i.e. the set $\{x_\star\}$ as well as the sets $\mathrm{vars}(p_i'), \mathrm{vars}(p_i'')$ for $i = 1, \ldots, n$ are pairwise mutually disjoint. Applying these thin multicontexts ω times we get

$$t \to^* (a_1(p_1' + \square + p_1'') \cdots a_n(p_n' + \square + p_n''))^\omega x_\star.$$

□

Lemma 35 *Let L be a regular language of thin forests which satisfies (8). Then $L = P'A^{\mathsf{ThinFor}}$.*

Proof Clearly $P'A^{\mathsf{ThinFor}} \subseteq L$. From Lemma 34 we have $L \subseteq P''A^{\mathsf{ThinFor}}$ when restricted to regular forests. Finally, from Lemma 33, we have $P'' \subseteq P'$, since for every $t \in L$ we have $tA^{\mathsf{ThinFor}} = \{t\} \subseteq L$. Therefore

$$L \subseteq P''A^{\mathsf{ThinFor}} \subseteq P'A^{\mathsf{ThinFor}} \subseteq L$$

when restricted to regular forests. Since both L and $P'A^{\mathsf{ThinFor}}$ are regular languages and they contain the same regular forests, they are equal. □

6.4 The 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).

Two forests t_0 and t_1 are called *EF-bisimilar* if Duplicator wins the following game, denoted by $\mathrm{G}_{\mathrm{bis}}(t_0, t_1)$. Spoiler begins the game by choosing some $i \in \{0, 1\}$ and a node x_i of the forest t_i. Duplicator responds by choosing a 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 Spoiler 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 the next 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. Note that if t_1, t_2 are EF-bisimilar and φ is a forest EF formula then $t_1 \models \varphi$ if and only if $t_2 \models \varphi$.

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

The results of [1] work without any difference if we restrict to the family of thin forests. Also, the results can be formulated in two variants: for all thin forests and only for regular ones (see [11]). The following theorem expresses these two variants.

Theorem 36 (Bojańczyk and Idziaszek [1]) *A regular language of (regular) thin forests L can be defined by a forest EF formula among all regular thin forests if and only if the following conditions hold:*

1. *L is invariant among all (regular) thin forests under EF-bisimulation,*
2. *the syntactic algebra of L satisfies*

$$v^{\omega}h = (v + v^{\omega}h)^{\infty}. \tag{9}$$

In this work we want to reformulate this theorem in the case of thin forests by replacing condition 1 to a set of identities. For this purpose we prove the following proposition.

Proposition 37 *A regular language of regular thin forests L is invariant under EF-bisimulation if and only if its syntactic thin-forest algebra satisfies the following identities*

$$h + v = v + h, \tag{10}$$

$$vh = vh + h, \tag{11}$$

$$(v + (vw)^{\infty})^{\infty} = (vw)^{\infty}, \tag{12}$$

$$(vwu)^{\infty} = (vuw)^{\infty}. \tag{13}$$

Before proving this proposition, we formulate a theorem summarizing our characterization.

Theorem 38 *A regular language of thin forests L can be defined by a forest EF formula if and only if its syntactic thin-forest algebra satisfies identities* (9), (10)–(13).

Proof Assume that a regular language of thin forests L is defined by a forest EF formula. In particular, $L \cap A^{\mathsf{regThin}\triangle}$ is closed under EF-bisimulation so by Proposition 37 the syntactic algebra of L satisfies identities (10)–(13). By Theorem 36 it satisfies also (9).

Now assume that the syntactic thin-forest algebra of L satisfies identities (9), (10)–(13). Let $L' = L \cap A^{\mathsf{regThin}\triangle}$. In that case, by Proposition 37 L' is closed under EF-bisimulation. Therefore, by Theorem 36 we know that L' is definable among regular thin forests by an EF forest formula ψ. Since $\mathrm{L}(\psi)$ is regular and agrees to L on regular thin forests, $\mathrm{L}(\psi) = L$. □

We note that Proposition 37 could also be formulated for regular languages of thin forests if the following conjecture should be true.

Conjecture 1 Let L be a regular language of forests. The language which consists of all forests which are EF-bisimilar to some forest from L is regular.

The rest of this section is devoted to the proof of Proposition 37. It follows the same lines as in [1] but we present it in full for the sake of completeness.

Note that the identity

$$h + g = g + h \tag{14}$$

follows from (10). The identity (13) can be rephrased in a more general way:

Lemma 39 *Let a thin-forest algebra (H, V) satisfy* (13). *Then*

$$(v_1 v_2 \cdots v_n)^{\infty} = (v_{\pi(1)} v_{\pi(2)} \cdots v_{\pi(n)})^{\infty}$$

for every permutation π of $\{1, \ldots, n\}$ and every $v_1, \ldots, v_n \in V$ such that $v_1 v_2 \cdots v_n \in V_+$.

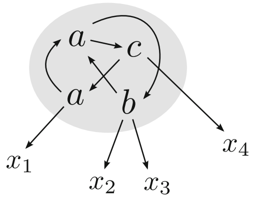


Fig. 9 Example of a 4-ary prime thin multicontexts over the alphabet $\{a, b, c\} \cup \{x_1, x_2, x_3, x_4\}$

Proof Observe that for any $v, w_1, w_2, u \in V$ such that $vw_1w_2u \in V_+$ we have

$$(v\ w_1\ w_2u)^\infty \overset{(13)}{=} (vw_2\ u\ w_1)^\infty \overset{(13)}{=} (vw_2w_1u)^\infty.$$

Now the lemma follows from the fact that every permutation is a product of adjacent transpositions. □

We recall the definition of *thin multicontexts* defined in Section 6.3. An *n-ary thin multicontext* over the variables $x_1, \ldots, x_n$ is a regular thin forest over the alphabet $A \cup \{x_1, \ldots, x_n\}$ where the variables $x_1, \ldots, x_n$ are allowed only in leaves. We allow multiple (possibly infinitely many) occurrences of each variable. Given thin forests $s_1, \ldots, s_n$ and an n-ary thin multicontext p, the thin forest $p(s_1, \ldots, s_n)$ over A is defined in the natural way.

An n-ary thin multicontext is called *prime* if, when treated as a thin forest over the alphabet $A \cup \{x_1, \ldots, x_n\}$, it has one root component, and all of the non-variable nodes are in this component.

An example of a 4-ary prime thin multicontext is given on Fig. 9.

We say that two thin multicontexts are EF-bisimilar if they are EF-bisimilar when treated as forests over the alphabet $A \cup \{x_1, \ldots, x_n\}$.

We say that an element $h \in H$ is *reachable* from $g \in H$ if there is some $v \in V$ with $h = vg$.

Lemma 40 *Let L be a regular language of regular thin forests invariant under EF-bisimulation. The reachability relation in the syntactic thin-forest algebra of L is antisymmetric.*

Proof We recall the proof from [1]. We prove that invariance under EF-bisimulation implies Property (11). Indeed, since α is surjective, there must be some context p with $\alpha(p) = v$ and some forest t with $\alpha(t) = h$. Since the forests $pt + t$ and pt are EF-bisimilar, their types must be equal, and hence (11) holds.

Suppose that g is reachable from h and vice versa. To prove antisymmetry, we need to show that $g = h$. By the assumption there are $v, w \in V$ with $g = wh$ and $h = vg$. Then we have

$$g = wh = wvg \overset{(11)}{=} wvg + vg = g + vg \overset{(11)}{=} vg = h.$$

□

The "only if" part of the proof of Proposition 37 is obvious — all the operations performed in the identities preserve EF-equivalence. The rest of this section is devoted to the "if" part.

We want to show that if two regular thin forests s and t are EF-bisimilar then they have the same types, i.e. $\alpha(s) = \alpha(t)$. The proof is by the induction on the number of components in s plus the number of components in t.

Lemma 41 *Without loss of generality, we can assume that s and t are trees.*

Proof Let $s_1, \ldots, s_n$ be all subtrees in a regular thin forest s and $t_1, \ldots, t_m$ be all subtrees in a regular thin forest t. By using inductively identity (11) and the fact that $s_1, \ldots, s_n$ are all the subtrees of s we have

$$\alpha(s) \stackrel{(11)}{=} \alpha(s) + \alpha(s_1) + \cdots + \alpha(s_n).$$

Now, by using (11) and (14) with $v = \Box$ we obtain

$$\alpha(s) + \alpha(s_1) + \cdots + \alpha(s_n) \stackrel{(11),(14)}{=} \alpha(s_1) + \cdots + \alpha(s_n).$$

Similarly $\alpha(t) = \alpha(t_1) + \cdots + \alpha(t_m)$. Since s and t are EF-bisimilar, every s_i is EF-bisimilar to some $\hat{s}_i \in \{t_1, \ldots, t_m\}$ and every t_i is EF-bisimilar to some $\hat{t}_i \in \{s_1, \ldots, s_n\}$. Suppose we proved the proposition for trees. Then $\alpha(s_i) = \alpha(\hat{s}_i)$ and $\alpha(t_i) = \alpha(\hat{t}_i)$, thus $\{\alpha(s_1), \ldots, \alpha(s_n)\} = \{\alpha(t_1), \ldots, \alpha(t_m)\}$. Therefore $\alpha(s) = \alpha(t)$. $\Box$

The induction basis is when both trees s and t have a single component. If s is finite then it has a single node a. In this case t also has to be a, since this is the only tree that is EF-bisimilar to a. (Note that we cannot check the label in a root of a tree but we can check whether a tree is of height 1 and check the label in a leaf.) Suppose now that s and t are infinite. Let $a_1, \ldots, a_n$ be the labels that appear in s (and therefore also in t). It is easy to see that s and t are EF-bisimilar to the tree $u = (a_1 \cdots a_n)^\infty$. All of the trees s, t, u can be treated as prime thin multicontexts of arity 0.

From Lemma 27, $s = (a_{\pi(1)} \cdots a_{\pi(n)})^\infty$ for some permutation π of the set $\{1, \ldots, n\}$. Applying Lemma 39 we get that $\alpha(s) = \alpha(u)$. Analogously we get that $\alpha(t) = \alpha(u)$.

We now do the induction step. Let $s_1, \ldots, s_n$ be all the subtrees of s that have fewer components than s. In other words, there is a prime n-ary thin multicontext p such that

$$s = p(s_1, \ldots, s_n).$$

Likewise, we distinguish all subtrees $t_1, \ldots, t_k$ inside t that have fewer components than t and find a prime k-ary thin multicontext q with

$$t = q(t_1, \ldots, t_k).$$

Since the trees s and t are EF-bisimilar, each tree s_i must be EF-bisimilar to some subtree $\hat{s}_i$ of t. By the inductive assumption, we know that the trees s_i and $\hat{s}_i$ have the same type (since s_i has fewer components than s and $\hat{s}_i$ has not more components than t). Likewise, each tree t_i has the same type as some subtree $\hat{t}_i$ of s.

By applying (11) in the same manner as in Lemma 41, we conclude that if either p or q is finite then s and t have the same type. We are left with the case when both p and q are infinite prime thin multicontexts. Suppose first that

(1) for some i, the tree $\hat{s}_i$ has the same number of components as t; and
(2) for some j, the tree $\hat{t}_j$ has the same number of components as s.

We use the same notion of reachability on types as was used in Lemma 40. From (1) we conclude that the tree $\hat{s}_i$ is in the root component of t and therefore the type of both s_i and $\hat{s}_i$ is reachable from the type of t. Since s_i is a subtree of s, we conclude that the type of s is reachable from the type of t. Reasoning in the same way from (2) we conclude that type of t is reachable from the type of s. Therefore, by Lemma 40, the types of s and t are equal (note that Lemma 40 used (11)).

Suppose now that one of (1) or (2) does not hold, say (1) does not hold (the other case is symmetric).

Lemma 42 *Without loss of generality, we can assume that $n \leq k$ and*

$$s = p(t_1, \ldots, t_n).$$

Proof Consider the tree $\hat{s} = p(\hat{s}_1, \ldots, \hat{s}_n)$. Since we replaced trees with EF-bisimilar ones, $\hat{s}$ is EF-bisimilar to s. Since we replaced trees with ones of the same type, $\hat{s}$ has the same type as s. So it is enough to prove the result for $\hat{s}$ and t.

Since (1) does not hold, then every $\hat{s}_i$ is equal to some t_j. Rename the subtrees $t_1, \ldots, t_k$ such that $\{\hat{s}_1, \ldots, \hat{s}_n\} = \{t_1, \ldots, t_{n'}\}$ for some $n' \leq \min(n, k)$. After possibly renaming variables in p, the tree $\hat{s}$ has the form $p(t_1, \ldots, t_{n'})$, like in the statement of the lemma. □

What about the trees $t_{n+1}, \ldots, t_k$ that do not appear in s? Each of these is EF-bisimilar to one of $s, t_1, \ldots, t_n$. For those that are EF-bisimilar to some $t_i \in \{t_1, \ldots, t_n\}$, we use the tree instead. Therefore, we can assume without loss of generality that

$$t = q(s, t_1, \ldots, t_n).$$

Lemma 43 *Any label $a \in A$ that appears in q also appears in p.*

Proof Let $a \in A$ be a label in q and consider the following strategy for Spoiler in the game over the trees s and t: he picks t and in that tree, some occurrence of a in the root component. Duplicator, in his response, cannot pick a node inside any of the trees $t_1, \ldots, t_n$, since none of these is EF-bisimilar to a tree in the root component of t, since (1) does not hold. Therefore, he must pick a node inside p. □

Let $a_1, \ldots, a_i$ be the labels that appear in q. Thanks to the above lemma, the labels that appear in p are $a_1, \ldots, a_i$ as well as possibly some other labels, say $a_{i+1}, \ldots, a_j$ for some $j \geq i$. Therefore the trees s, t look like on Fig. 10. Let us define the following two contexts

$$x = a_1 \cdots a_i(\square + t_1 + \cdots + t_n), \qquad y = a_{i+1} \cdots a_j.$$

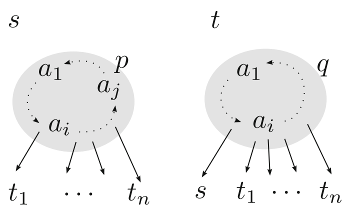


Fig. 10 Two trees illustrating the proof of Lemma 43

Observe that from Lemma 27 every tree with a connected component in the root can be written as

$$u = (b_1(t_1' + \square + t_1'') \cdots b_m(t_m' + \square + t_m''))^\infty$$

for some $m \geq 1$, letters $b_1, \ldots, b_m \in A$, and thin forests $t_1', t_1'', \ldots, t_m', t_m''$. From the identities we conclude that for every $v_1, \ldots, v_m \in V_+$ and $h_1, \ldots, h_m, g_1, \ldots, g_m \in H$ we have

$$\begin{aligned} (v_1(h_1 + \square + g_1) \ \cdots \ v_m(h_m + \square + g_m))^\infty &\overset{(13)}{=} \\ &= (v_1 \cdots v_m(h_1 + \square + g_1) \cdots (h_m + \square + g_m))^\infty = \\ &= (v_1 \cdots v_m(h_1 + \cdots + h_m + \square + g_m + \cdots + g_1))^\infty \overset{(10)}{=} \\ &= (v_1 \cdots v_m(\square + h_1 + \cdots + h_m + g_1 + \cdots + g_m))^\infty. \end{aligned}$$

That shows that the type of u is the same as the type of

$$(b_1 \cdots b_m(\square + t_1' + \cdots + t_m' + t_1'' + \cdots + t_m''))^\infty$$

Using identities (13) and (10) we can further rearrange letters b_i and trees from forests t_i', t_i''. From this it is easy to show that $\alpha(s) = \alpha((xy)^\infty)$ and $\alpha(t) = \alpha((x(\square + s))^\infty)$.

7 Languages Weak MSO-Definable Among All Forests

In this section we consider a non-standard 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 weak MSO-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 44 Let L be a regular language of thin forests and φ be a formula of weak MSO. We say that φ defines L among all forests if

$$L = \left\{ t \in A^{\mathsf{For}} : \ t \models \varphi \right\}.$$

Note that the class of languages definable in weak MSO 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 weak MSO-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 45 *The examples of weak MSO-definable languages lying arbitrarily high on the finite levels of the Borel hierarchy (see* [22]*) can be encoded into thin forests in a way that is weak MSO-definable among all forests.*

To formulate the algebraic characterization we use the notion of a *bottom element*: let L be a regular language of thin forests and $\alpha : A^{\mathsf{Thin}\triangle} \to (H, V)$ 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 $h_1 = (\square + h_2)h_1 = h_1 + h_2 = (h_1 + \square) h_2 = h_2$.

The main result of this section is the following characterization.

Theorem 46 *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 weak MSO-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:*

$$\text{if } h = v(w+h)^\infty \text{ or } h = v(h+w)^\infty \text{ for some } v \in V, w \in V_+, \\ \text{then } h \text{ is the bottom element for } L. \tag{15}$$

The implication (2)⇒(3) is trivial — any language definable in weak MSO is Borel, thus not $\mathbf{\Pi}^1_1$-hard. The remaining implications are proved in the following subsections.

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

Corollary 47 *It is decidable whether a given regular language of thin forests L is weak MSO-definable among all forests.*

One of the applications of our characterization is the following proposition.

Proposition 48 *Assume that L is a regular language of forests that is recognized by a non-deterministic (or equivalently alternating)* $(1, 2)$*-automaton. Assume additionally that L contains only thin forests. Then L can be defined in weak MSO among all forests.*

Proof Since L is recognizable by a $(1, 2)$-automaton, L is an analytic subset of A^{For} (c.f. [9]). Since the space of all forests is an uncountable Polish topological space, $\mathbf{\Pi}^1_1 \neq \mathbf{\Sigma}^1_1$ (see [12, Corollary 26.2]). Therefore, L cannot be $\mathbf{\Pi}^1_1$-hard, thus L satisfies condition 3 in Theorem 46. □

7.1 (1)⇒(2)

Definition 49 Assume that $t \in A^{\mathsf{For}}$ is a forest and $x \preceq y$ are two nodes of t. We say that a node z is *off the path* from x to y if z is not an ancestor of y ($z \not\preceq y$) but there exists x' such that $x \preceq x' \prec y$ and z is a child of x'.

We start by showing the following lemma. The constructed formula φ_m will serve as a basis in the following constructions.

Lemma 50 *For every $m \in \mathbb{N}$ there exists a weak MSO formula φ_m defining among all forests the language of thin forests of CB-rank at most m (denoted $A^{\mathsf{ThinFor}\leq \mathsf{m}}$, see Definition 10).*

Proof Induction on m. For $m = 0$ it is trivial, since the only forest of CB-rank 0 is the empty forest.

Assume that the thesis holds for m — we have defined a formula φ_m. Consider a weak MSO formula φ_{m+1} that for a given forest $t \in A^{\mathsf{For}}$ says that:

- there exists a finite forest $r \subseteq t$,
- and a number of leafs $x_1, x_2, \ldots, x_n$ of r (n may equal 0),
- such that if y is off r in t then y is a child of one of the leafs $x_1, \ldots, x_n$ and
- for every selected leaf x_i of r,
- there are infinitely many nodes y such that $x_i \prec y$ and
- for every node z that is off the path from x_i to y,
- the subtree $t\restriction_z$ has CB-rank at most m (what corresponds to checking the formula φ_m on $t\restriction_z$).

First assume that φ_{m+1} holds on a given forest t. Take $r \subseteq t$ and observe that by König's Lemma, there are infinite branches $\pi_1, \pi_2, \ldots, \pi_n$ starting in leafs $x_1, \ldots, x_n$ of r respectively such that for every node z that is off π_i and below x_i the CB-rank of $t\restriction_z$ is at most m. Therefore $t' = \mathrm{Dv}^m_{CB}(t)$ does not contain any of these nodes z. So the set of branches of t' is contained in $\pi_1, \pi_2, \ldots, \pi_n$ and the branches of r, so $\mathrm{Dv}^{m+1}_{CB}(t) = 0$. So $\mathrm{rank}^{\mathrm{CB}}(t) \leq m + 1$.

Now assume that $\mathrm{rank}^{\mathrm{CB}}(t) \leq m + 1$. Therefore, $t' = \mathrm{Dv}^m(t)$ has only finitely many infinite branches. So t' is of the form $r(\pi_1, \pi_2, \ldots, \pi_n)$ where r is a finite forest and branches π_i go through some leafs $x_1, \ldots, x_n$ of r (see Fact 12, we assume that r contains all finite branches of t' and $\pi_1, \ldots, \pi_n$ are the infinite branches of t'). We satisfy the formula φ_{m+1} by taking $r, \pi_1, \ldots, \pi_n$ as above and using as nodes y all the nodes of the form $x_i \prec y \prec \pi_i$. By the definition of t', every node z that is off one of the branches π_i and below x_i has CB-rank at most m. So the subtree $t\restriction_z$ satisfies φ_m. □

The crucial inductive part of the proof is expressed by the following proposition.

Proposition 51 *Let (H, V) be a finite thin-forest algebra, $\alpha : A^{\mathsf{Thin}\triangle} \to (H, V)$ be a homomorphism, and let m be a number. For every type $h \in H$ there exists a* *weak MSO formula $\varphi_m(h)$ that defines those forests $t \in A^{\mathsf{For}}$ such that $t \in A^{\mathsf{ThinFor}}$,* $\mathrm{rank}^{\mathrm{CB}}(t) = m$*, and the type of t is h with respect to α (i.e. $\alpha(t) = h$).*

For $m = 0$ the only forest of CB-rank 0 is the empty forest. So for $h = 0$ the formula $\varphi_0(h)$ is equivalent to φ_0 from Lemma 50 and for other types h it is false. Assume that the thesis of the proposition holds for all types h and a given number m. We show it for $m + 1$.

First we write a formula $\psi_m(x, y)$ expressing that for a given pair of nodes $x, y \in x \prec y(t)$:

- $x \preceq y$,
- the subtrees $t{\restriction}_x$ and $t{\restriction}_y$ have CB-ranks exactly m (we check it using φ_m and $\neg\varphi_{m-1}$), and
- for every z that is off the path from x to y
- the CB-rank of $t{\restriction}_z$ is at most $m - 1$ (i.e. check φ_{m-1} on $t{\restriction}_z$).

The following lemma summarizes the most important properties of formulas $\psi_m(x, y)$.

Lemma 52 *Assume that for a given forest $t \in A^{\mathsf{For}}$ and a node $x \in$* $\mathrm{dom}(t)$ *there are infinitely many nodes y such that $\psi_m(x, y)$. Then* $\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_x) = m$ *and the set of nodes of CB-rank m below x in t is contained in a single infinite branch π of t.*

Moreover, $\psi_m(x, y)$ holds for some $y \in$ $\mathrm{dom}(t)$ *if and only if $x \preceq y \prec \pi$.*

Proof Take a forest t and a node $x \in \mathrm{dom}(t)$ as in the statement. Observe that $\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_x) = m$. Let $S \subseteq \mathrm{dom}(t)$ be the set of nodes $y \in \mathrm{dom}(t)$ such that $\psi_m(x, y)$ holds. Observe that if $x \preceq y_1 \preceq y_2 \in t$ and $y_2 \in S$ then $y_1 \in S$. Since there are infinitely many nodes y satisfying $\psi_m(x, y)$ so S is infinite. Observe that for every node z that is off S we have $\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_z) \leq m - 1$. Every node $y \in S$ satisfies $\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_y) = m$. So S is the set of nodes of CB-rank m in t below x.

Assume that there is a node $x' \in S$ such that two distinct children y_1, y_2 of x belong to S. Then $\psi_m(x, y_1)$ holds, but y_2 is off the path from x to y_1. So $\mathrm{rank}^{\mathrm{CB}}(y_2, t) \leq m - 1$ by the definition of ψ_m. But $y_2 \in S$ so $\mathrm{rank}^{\mathrm{CB}}(y_2) = m$. A contradiction.

Therefore, S is contained in a single infinite branch of t. □

The above lemma states that the formula $\psi_m(x, y)$ enables us to fix in a weak MSO-definable way a particular branch π in our forest such that almost all nodes that are off this branch have CB-ranks smaller than m. What remains is to compute the type of the subtree rooted in the node x basing on the types of subtrees that are off π. The following formula is an intermediate step in this construction.

Fact 53 *For a type $v \in V$ there exists a weak MSO formula $\gamma_m^v(x, y_1, y_2)$ that for all nodes x, y_1, y_2 expresses the following facts:*

- $x \preceq y_1 \preceq y_2$,
- *$\psi_m(x, y_2)$ holds,*[1]
- *$\alpha(p) = v$, where p is the context rooted in y_1 with the hole in y_2.*

Proof To achieve the last item on the list, the formula computes the types of subtrees rooted in nodes off the path from y_1 to y_2 using inductive formulas $\varphi_{m'}(h)$ for $m' < m$ and $h \in H$. Then it uses the in_l, in_r operations of the finite algebra (H, V) to verify the type of p. □

Now we show how to compute a type of a tree with one main branch.

Definition 54 Let x be a node and $h \in H$ be a type. Let the formula $\delta_m^h(x)$ express the following facts:

- there are infinitely many nodes y such that $\psi_m(x, y)$,
- there exists a pair of context types $u, v \in V$ such that $uv^\infty = h$,
- there exists a node y_0 such that $\gamma_m^u(x, x, y_0)$ holds (the type of the context between x and y_0 is u) and
- for every node y_1 such that $\psi_m(x, y_1)$ there exists a pair of nodes y_2, y_3 such that $y_1 \prec y_2 \prec y_3$, and $\psi_m(x, y_3)$ hold and
- the formulas $\gamma_m^v(x, y_0, y_2)$, $\gamma_m^v(x, y_0, y_3)$, and $\gamma_m^v(x, y_2, y_3)$ hold (the types of the three contexts equal v).

The last two items of this formula are based on a construction from [23] that enables us to verify a type of a given infinite word in first-order logic using predicates of the form "the type of the infix between the positions y_1 and y_2 is v".

Lemma 55 *Let t be a forest and x be a node such that there are infinitely many nodes y satisfying $\psi_m(x, y)$. Then $\alpha(t\restriction_x) = h$ if and only if $\delta_m^h(x)$ holds on t.*

Proof First assume that $t \models \delta_m^h(x)$ for some $x \in \mathrm{dom}(t)$ and $h \in H$. Let π be the branch defined by the predicate $\psi_m(x, y)$ as in Lemma 52.

Let $y_1 \preceq y_2$ be two nodes of the given forest t. In this proof, by $[y_1, y_2)$ we denote the context rooted in y_1 with the hole instead of $t\restriction_{y_2}$.

We show that the formula $\delta_m^h(x)$ gives rise to a sequence of nodes $z_0 \prec z_1 \prec z_2 \ldots$ on π such that for some types u, v satisfying $uv^\infty = h$ we have:

$$\alpha([x, z_0)) = u, \qquad \alpha([z_i, z_{i+1})) = v. \tag{16}$$

Having done so, we conclude that the type of $t\restriction_x$ is h.

Let us fix y_0 as in the definition of δ. We will set y_1 to various nodes along π obtaining appropriate nodes y_2, y_3.

Let us start with y_1 equal y_0 and consider y_2, y_3 given by $\delta_m^h(x)$. Let $z_1 = y_2$ and $u_1 = y_3$. Our inductive invariant is that the types of all three contexts $[y_0, z_i)$, $[y_0, u_i)$, and $[z_i, u_i)$ equal v. For $i = 1$ we get it by the definition of $\delta_m^h(x)$. Assume that $z_i \prec u_i$ are defined and put $y_1 = u_i$. Consider y_2, y_3 as in the definition of

[1] It implies $\psi_m(x, y_1)$.

$\delta_m^h(x)$. Let us put $z_{i+1} = y_2$ and $u_{i+1} = y_3$. Similarly as in the base step we get the invariant by the definition. Now consider the type of the context $[z_i, z_{i+1})$:

$$\begin{aligned}\alpha\left([z_i, z_{i+1})\right) &= \alpha\left([z_i, u_i)\right) \cdot \alpha\left([u_i, z_{i+1})\right)\\ &= v \cdot \alpha\left([u_i, z_{i+1})\right)\\ &= \alpha\left([y_0, u_i)\right) \cdot \alpha\left([u_i, z_{i+1})\right)\\ &= \alpha\left([y_0, z_{i+1})\right)\\ &= v.\end{aligned}$$

Therefore, the constructed sequence $z_1 \prec z_2 \prec \ldots$ satisfies (16).

For the other direction take a forest t with a node x and a branch π as in Lemma 52. Using a Ramsey argument along π we find a pair of types u, v and an infinite sequence of nodes z_i along π satisfying (16). Since $\alpha(t{\restriction_x}) = h$, $uv^\infty = h$. Therefore, we can satisfy the formula $\delta_m^h(x)$ using successive nodes z_i. □

Now we can rewrite the formula φ_m defined in the proof of Lemma 50 so that it additionally verifies the type of the given forest. Take $m > 0$ and define $\varphi_m(h)$ that says:

1. there exists a finite prefix $r \subseteq t$,
2. and a number of leafs $x_1, \ldots, x_n$ of r,
3. and a sequence of types $h_1, \ldots, h_n$ such that
4. the type of $r(h_1, h_2, \ldots, h_n)$ is h and
5. for every leaf x_i,
6. there are infinitely many nodes y such that $\psi_m(x_i, y)$,
7. and $\delta_m^{h_i}(x_i)$ holds for all $i = 1, \ldots, n$.

What remains is to observe that the forest r and leafs x_i correspond to the final prefix of a given forest, formulas $\psi_m(x_i, y)$ fix infinite branches passing through x_i and $\delta_m^{h_i}(x_i)$ verifies the type of the subtree $t{\restriction_{x_i}}$. Therefore, $\varphi_m(h)$ holds on t if and only if $\mathrm{rank}^{\mathrm{CB}}(t) = m$ and $\alpha(t) = h$.

7.2 (3)⇒(4)

Assume contrary that there are types h, v, w, u in the syntactic algebra of a regular language L such that (by symmetry) $h = v(w + h)^\infty$ and $\alpha^{-1}(u \cdot h) \subseteq L$. We show that L is $\mathbf{\Pi}_1^1(A^{\mathsf{For}})$-hard.

Definition 56 An ω-tree is a subset $\alpha \subseteq \mathbb{N}^*$ that is closed under prefixes. The space of ω-trees is denoted as ωTr. The set of all ω-trees that does not contain any infinite branch is denoted as WF $\subseteq \omega$Tr.

Fact 57 (**See** [12, Chapter IV Section 33.A]) *The space of ω-trees is a Polish topological space. The set* WF *is $\mathbf{\Pi}_1^1$-complete.*

First we define a continuous function mapping ω-trees $T \in \omega\mathrm{Tr}$ to forests $t(T) \in A^{\mathsf{For}}$. Let us fix a forest t_h of type h and contexts c_v, c_w, c_u of types v, w, u respectively. If $T = \emptyset$ then let $t(T) = t_h$. If T is non-empty let $T_0, T_1, \ldots$ be the sequence of consecutive subtrees under the root of T. First let us put $c_i = c_v(c_w + t(T_i))$ and define

$$t(T) = c_v(c_w + t_h) \cdot c_0 \cdot c_0 \cdot c_1 \cdot c_1 \cdot c_2 \cdot \ldots . \tag{17}$$

Note that in this definition, for every $i \in \mathbb{N}$ we put the context c_i twice.

Observe that since every context c_i is guarded (because $w \in V_+$), the function $t : \omega\mathrm{Tr} \to A^{\mathsf{For}}$ defined above is continuous — given information about a consecutive child of the root of T it produces further parts of the result $t(T)$.

Now we define $f(T) = c_u \cdot t(T)$.

Lemma 58 *An ω-tree $T \in \omega\mathrm{Tr}$ does not contain an infinite branch (belongs to* WF*) if and only if $f(T) \in L$.*

Proof First assume that $T \in \mathrm{WF}$. We inductively on the structure of T show that $t(T)$ is a thin forest and $\alpha(t(T)) = h$. Having done so we will conclude that $\alpha(f(T)) = u \cdot h$, therefore $f(T) \in L$. Formally speaking the induction on the structure of T is based on the standard rank on well-founded ω-trees, see [12, Chapter I Section 2.E].

If $T = \emptyset$ then it is trivial. Otherwise, without loss of generality we can assume that T contains infinitely many subtrees $(T_i)_{i\in\mathbb{N}}$ of the root and by the inductive assumption we know that $t(T_i)$ is thin and has type h. Therefore, by the definition $t(T)$ is thin and by condition (15) and definition (17) we have

$$\alpha\left(t(T)\right) = v(w + h)^{\infty} = h.$$

Now take $T \notin \mathrm{WF}$. Assume that $d \in \mathbb{N}^{\omega}$ is an infinite branch of T. We show how to embed a full binary tree into $f(T)$ thus showing that $f(T)$ is not thin. Since $L \subseteq A^{\mathsf{ThinFor}}$ so $f(T) \notin L$.

For a node $w \in T$ by $T\restriction_w$ we denote the subtree of T rooted in w. For a number n we denote by $d\restriction_n$ the prefix of d of length n. Thus, $T\restriction_{d\restriction_n}$ is the n-th subtree of T along d. For $n = 0$ we have $T\restriction_{d\restriction_n} = T$.

We take a sequence $e \in \{0, 1\}^{\omega}$ and define an infinite branch b_e of $f(T)$. Intuitively we want to find a sequence $t_0, t_1, \ldots$ of subforests of $f(T)$. During each step t_n is a copy of the $t(T\restriction_{d\restriction_n})$ forest. We start by putting t_0 as the subforest put in the hole of c_u. From that moment on we will traverse infinitely many copies of c_v. In the n-th step we go to one of the two copies of the forest $t(T\restriction_{d\restriction_n})$ in the current subforest t_n — either the first or the second one, depending on the value of $e(n) \in \{0, 1\}$.

To be more precise, we additionally define a sequence of contexts p_n. Our aim is that for every n:

$$\begin{aligned} t_n &= t\left(T\restriction_{d\restriction_n}\right), \\ f(T) &= p_n \cdot t_n, \\ p_{n+1} &= p_n \cdot s_n \text{ for a guarded context } s_n. \end{aligned} \tag{18}$$

Note that if a sequence p_n satisfies these properties then the holes of contexts p_n do indicate an infinite branch b_e of $f(T)$.

We start with $p_0 = c_u$ and note that by the definition of f the invariants (18) are satisfied. Assume that p_n is defined. Let $d(n) = k$ — the branch d goes through k-th child of the current subtree $T{\upharpoonright}_{d{\upharpoonright}_n}$. Let us recall the definition (17) for the subtree $T{\upharpoonright}_{d{\upharpoonright}_n}$ and let

$$\begin{aligned}
r_P &= c_v(c_w + t_h) \cdot c_0 \cdot c_0 \cdot \ldots \cdot c_{k-1} \cdot c_{k-1},\\
r_0 &= \Box,\\
r_1 &= c_k,\\
t_F &= c_{k+1} \cdot c_{k+1} \cdot c_{k+2} \cdot c_{k+2} \cdot \ldots,\\
s_n &= r_P \cdot r_{e(n)} \cdot c_v \cdot \left(c_w \cdot r_{1-e(n)} \cdot t_F + \Box\right).
\end{aligned}$$

Note that by the definition $f(T) = p_n \cdot s_n \cdot t_{n+1}$, and s_n is guarded, so the context p_{n+1} defined as $p_n \cdot s_n$ satisfies the invariant (18).

Observe that the possible two values of $e(n) \in \{0, 1\}$ induce two different contexts p_{n+1} with two incomparable holes (we use either r_0 or r_1 on the path to the hole of s_n). Therefore, for any $e' \neq e$ we have $b_{e'} \neq b_e$. So indeed the forest $f(T)$ is not thin — it contains a full binary subtree. □

7.3 (4)⇒(1)

First we extend the definition of the CB-rank to thin contexts:

$$\mathrm{rank}^{\mathrm{CB}}(p) = \max\{\mathrm{rank}^{\mathrm{CB}}(p{\upharpoonright}_x) : x \text{ is off the path to the hole of } p\}.$$

It is easy to see that the rank can be calculated inductively on the structure of a term describing a given thin forest:

$$\begin{aligned}
\mathrm{rank}^{\mathrm{CB}}(s+t) &= \max(\mathrm{rank}^{\mathrm{CB}}(s), \mathrm{rank}^{\mathrm{CB}}(t)),\\
\mathrm{rank}^{\mathrm{CB}}(p \cdot q) &= \max(\mathrm{rank}^{\mathrm{CB}}(p), \mathrm{rank}^{\mathrm{CB}}(q)),\\
\mathrm{rank}^{\mathrm{CB}}(p \cdot t) &= \begin{cases} \max(\mathrm{rank}^{\mathrm{CB}}(p), \mathrm{rank}^{\mathrm{CB}}(t)) & \text{if } p \text{ is non-guarded},\\ \max(\mathrm{rank}^{\mathrm{CB}}(p), \mathrm{rank}^{\mathrm{CB}}(t), 1) & \text{if } p \text{ is guarded,}\end{cases}\\
\mathrm{rank}^{\mathrm{CB}}(p^\infty) &= 1 + \mathrm{rank}^{\mathrm{CB}}(p),\\
\mathrm{rank}^{\mathrm{CB}}(in_l(t)) &= \mathrm{rank}^{\mathrm{CB}}(in_r(t)) = \mathrm{rank}^{\mathrm{CB}}(t),\\
\mathrm{rank}^{\mathrm{CB}}(a) &= 1\\
\mathrm{rank}^{\mathrm{CB}}(0) &= \mathrm{rank}^{\mathrm{CB}}(\Box) = 0,
\end{aligned}$$

for thin contexts p, q, thin forests s, t, and a letter $a \in A$.

In fact, for regular thin forests and contexts, CB-rank is closely related to the maximal nesting depth of the loop operation. It is stated more formally in the two following lemmas.

Lemma 59 *Every regular thin context p of CB-rank n can be written as either*

$$p_1(p_2 + t) \qquad \text{or} \qquad p_1(t + p_2)$$

where t is a thin forest of CB-rank n.

Proof We do a proof by induction on the structure of a term which generates p.

If $p = \Box$ or $p = a$ we put $t = 0$.

If $p = in_l(t)$ for some forest t then we put $p_1 = p_2 = \Box$.

Otherwise $p = qr$ for some contexts q, r. If r has CB-rank n then by the inductive assumption w.l.o.g. $r = r_1(r_2 + s)$ and $p = qr_1(r_2 + s)$. If q has CB-rank n then $q = q_1(q_2 + s)$ and $p = q_1(q_2 + s)r = q_1(q_2 r + s)$. $\Box$

Lemma 60 *Every regular thin forest t of CB-rank $n > 0$ can be written as either*

$$p(q + t')^{\infty} \qquad \text{or} \qquad p(t' + q)^{\infty}$$

where t' is a thin forest of CB-rank $n-1$.

Proof We do a proof by induction on the structure of a term which generates t.

Assume that $t = s_1 + s_2$ and w.l.o.g. let s_1 be of CB-rank n. Then by the inductive assumption w.l.o.g. $s_1 = p(q + t')^{\infty}$ and t' is of CB-rank $n-1$. Thus

$$t = p(q + t')^{\infty} + s_2 = (p + s_2)(q + t')^{\infty}.$$

Assume then $t = ps$. If s is of CB-rank n we do similarly. If p is of CB-rank n then from Lemma 59 $p = p_1(p_2 + s')$ and s' is of CB-rank n. Thus

$$t = p_1(p_2 + s')s = p_1(p_2 s + \Box)s',$$

and again we use the inductive assumption.

Finally if $t = p^{\infty}$ then p is of CB-rank $n-1$ and from Lemma 59 $p = p_1(p_2 + s')$ and s' is of CB-rank $n-1$. Therefore

$$t = (p_1(p_2 + s'))^{\infty} = p_1((p_2 + s')p_1)^{\infty} = p_1(p_2 p_1 + s')^{\infty}.$$

$\Box$

Lemma 61 *If a regular language L contains a forest of CB-rank n, it contains also a regular forest of CB-rank n.*

Proof From Lemma 50 we get that the language

$$L \cap (A^{\mathsf{ThinFor}\leq n} - A^{\mathsf{ThinFor}\leq n-1})$$

of all thin forests from L of CB-rank n is regular. The lemma follows from the fact that every regular language of thin forests contains a regular thin forest. $\Box$

We introduce a directed graph G, whose set of vertices is a horizontal monoid H of the syntactic thin-forest algebra of L. For $h, g \in H$ a directed edge $h \to g$ belongs to G if there exist elements $v, w \in V$ such that $g = v(w + h)^{\infty}$ or $g = v(h + w)^{\infty}$. The graph G is closed under transitivity:

Lemma 62 *If for $h, g, f \in H$ edges $h \to g$ and $g \to f$ belong to G then the edge $h \to f$ also belongs to G.*

Proof Let $g = v(w + h)^\infty$ and $f = v'(w' + g)^\infty$ for some $v, w, v', w' \in V$. Symmetric cases are done analogously. We have

$$\begin{aligned} f &= v'(w' + g)^\infty = \\ &= v'(w' + g)(w' + g)^\infty = \\ &= v'(w'(w' + g)^\infty + g) = \\ &= v'(w'(w' + g)^\infty + v(w + h)^\infty) = \\ &= v'(w'(w' + g)^\infty + v)(w + h)^\infty. \end{aligned}$$

Thus $f = v''(w + h)^\infty$ for $v'' = v'(w'(w' + g)^\infty + v)$. □

Now we show that if a language L satisfies (15) then there is a bound on CB-ranks of forests in L. We do a proof by contradiction — we assume that the language L satisfies (15) but it has unbounded CB-rank.

From Lemmas 60 and 61 for any sufficiently large n we have a family of forests $t_0, t_1, \ldots, t_n$ such that:

(a) $\mathrm{rank}^{\mathrm{CB}}(t_i) = i$,
(b) there is an edge $\alpha(t_{i-1}) \to \alpha(t_i)$ in G,
(c) $t_n \in L$.

Therefore for $n \geq |H|$ there exist two indices i, j such that $j < i \leq n$ and $\alpha(t_j) = \alpha(t_i) = h_\star$. Condition (b) ensures that there is a path from $h_\star$ to itself in G, thus from Lemma 62 there is a loop-edge $h_\star$ in G, hence $h_\star = v(w + h_\star)^\infty$ or $h_\star = v(h_\star + w)^\infty$ for some $v, w \in V$. Therefore from (15) $h_\star$ is the bottom element for L.

Since there is a path from $h_\star$ to $\alpha(t_n)$ in G, we get that $\alpha(t_n)$ is equal to either $v(w + h_\star)^\infty$ or $v(h_\star + w)^\infty$ for some $v, w \in V$. Observe that

$$v(w + h_\star)^\infty = v(w + h_\star)(w + h_\star)^\infty = v(w(w + h_\star)^\infty + \square)h_\star = h_\star,$$

where the last equality comes from the definition of the bottom element. Therefore we get that $\alpha(t_n) = h_\star$. This contradicts condition 4.

8 Descriptive Properties

In this section we show a number of descriptive properties of regular languages of thin forests.

8.1 Automata

The following theorem expresses that the collapse from Theorem 26 is the best we can get from the point of view of the alternating index hierarchy (also known as the Rabin-Mostowski hierarchy).

Theorem 63 *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.*

Proof We define the language $L \subseteq \{a, b\}^{\mathsf{ThinFor}}$ as containing those thin forests that have a branch with infinitely many letters a. First we observe that this language is $\mathbf{\Pi}^1_1$-hard (see Theorem 46). If it were recognised by an alternating $(1, 2)$-automaton then by Proposition 48 we get a contradiction.

What remains is to show that L cannot be recognized by any alternating $(0, 1)$-automaton. In that case the complement $L^c = \{a, b\}^{\mathsf{For}} \setminus L$ would be recognizable by a non-deterministic $(1, 2)$-automaton $\mathcal{A}$.

Let n be the number of states Q of $\mathcal{A}$. Consider thin forests defined inductively:

$$t_0 = 0, \quad t_{i+1} = (b(\square + at_i))^{\infty}.$$

Let $t = t_{n+1}$. Note that t is thin and $t \in L^c$. Let ρ be an accepting run of $\mathcal{A}$ on t. Observe that ρ is accepting on every b-labelled branch. Therefore, one can find a node with a state of priority 2 on such a branch. Therefore, we can inductively find a sequence of nodes $u_0 \prec u_1 \prec \ldots \prec u_n$ of t such that for every $i = 0, 1, \ldots, n - 1$:

- the run ρ has a state of priority 2 on the path between u_i and u_{i+1},
- there is a node with label a on the path between u_i and u_{i+1}.

Since n is the number of the states of $\mathcal{A}$, so $\rho(u_i) = \rho(u_j)$ for some $i < j$. Therefore, we can decompose (t, ρ) as the context c_1 with a hole in u_i, the context c_2 between u_i and u_j, and the tree t_3 rooted in u_j, in such a way that

$$(t, \rho) = c_1 \cdot c_2 \cdot t_3.$$

Let (t', ρ') be the forest over the alphabet $\{a, b\} \times Q$ equal $c_1 \cdot c_2^{\infty}$. Note that t' has a branch with infinitely many letters a, so $t' \in L$ but ρ' is an accepting run of $\mathcal{A}$ on t' — a contradiction. □

8.2 Embeddings and Quasi Skeletons

The definition of a skeleton σ of a forest t is a co-analytic definition — σ has to contain almost all nodes on every branch of t. Our aim in this section is to define objects less rigid than skeletons but definable in an analytic way. For this purpose, we introduce two relations R_E and R_Q.

Observe that the CB-rank of a forest depends only on the domain of this forest. Therefore, to simplify the notation we will restrict our attention to the case when the alphabet A contains a single letter $A = \{a\}$. In that case we can identify a forest $t \in A^{\mathsf{For}}$ with its domain $\mathrm{dom}(t) \subseteq \omega^+$.

Proposition 64 *There exists an analytic* ($\mathbf{\Sigma}^1_1$) *relation* $R_E \subseteq \{a\}^{\mathsf{For}} \times \{a\}^{\mathsf{For}}$ *such that for every forest t_1 and every thin forest t_2:*

$$(t_1 \textit{ is thin and } \mathrm{rank}^{\mathrm{CB}}(t_1) \leq \mathrm{rank}^{\mathrm{CB}}(t_2)) \quad \textit{if and only if} \quad (t_1, t_2) \in R_E.$$

Intuitively, the relation R_E is defined by the expression of the form: $(t_1, t_2) \in R_E$ if there exists an *embedding* of $\mathrm{dom}(t_1)$ to $\mathrm{dom}(t_2)$. However, to avoid technical difficulties, we do not introduce exact definition of an embedding. Instead, we recall some standard methods from descriptive set theory, see [12, Section 34.D], namely the *Borel derivatives*. It will be shown that the derivative Dv_{CB} from Section 3 is (modulo some technical extension) a *Borel derivative*. We follow here the notions used in [12].

Definition 65 Let X be a countable set and $\mathcal{D} = \mathbf{P}(X)$. A *derivative* on $\mathcal{D}$ is a map $D : \mathcal{D} \to \mathcal{D}$ such that $D(A) \subseteq A$ and $D(A) \subseteq D(B)$ for $A \subseteq B$, $A, B \in \mathcal{D}$. For $A \in \mathcal{D}$ we define $D^0(A) = A$, $D^{\eta+1}(A) = D(D^\eta(A))$ and for a limit ordinal η

$$D^\eta(A) = \bigcap_{\eta' < \eta} D^{\eta'}(A).$$

Now, let $|A|_D$ for $A \in \mathcal{D}$ be the least ordinal η such that $D^\eta(A) = D^{\eta+1}(A)$. Such an ordinal exists by monotonicity of D and since X is countable, $\eta < \omega_1$. We additionally put

$$D^\infty(A) = D^{|A|_D}(A).$$

Now let us state [12, Theorem 34.10] in the case of countable X.

Theorem 66 (**Theorem 34.10 from** [12, Section 34.E]) *Let X be a countable set and $\mathcal{D} = \mathbf{P}(X)$. Let $D : \mathcal{D} \to \mathcal{D}$ be a derivative that is Borel. Put*

$$\Omega_D = \{F \in \mathcal{D} : \; D^\infty(F) = \emptyset\}.$$

Then Ω_D is $\mathbf{\Pi}^1_1$ and the map $F \mapsto |F|_D$ is a $\mathbf{\Pi}^1_1$-rank on Ω_D.

Our aim is to present Dv_{CB} as a Borel derivative in such a way that $\Omega_D = \{a\}^{\mathsf{ThinFor}}$ and the map $F \mapsto |F|_D$ is the CB-rank. The above theorem will then imply that the CB-rank of thin forests is a $\mathbf{\Pi}^1_1$-rank. Then, by the definition of $\preceq^*_{\mathrm{rank}^{\mathrm{CB}}}$ (see [12, Section 34.B]) we obtain that

$$\begin{aligned} R_E(t, t') &\overset{\mathrm{def}}{\Leftrightarrow} t \preceq^*_{\mathrm{rank}^{\mathrm{CB}}} t' \\ &\Leftrightarrow t' \notin \{a\}^{\mathsf{ThinFor}} \vee (t, t' \in \{a\}^{\mathsf{ThinFor}} \wedge \mathrm{rank}^{\mathrm{CB}}(t) \le \mathrm{rank}^{\mathrm{CB}}(t')) \end{aligned}$$

is a $\mathbf{\Sigma}^1_1$-relation. This will conclude the proof of Proposition 64.

Fact 67 *The CB-rank of thin forests comes from a Borel derivative, as in the assumptions of Theorem 66.*

Proof Let $X = \omega^+$ and $\mathcal{D} = \mathbf{P}(X)$. Note that in this case $\{a\}^{\mathsf{For}} \subseteq \mathcal{D}$. Let us extend the derivative Dv_{CB} to a function $D : \mathcal{D} \to \mathcal{D}$ by defining it also on sets $F \subseteq X$ such that $F \notin \{a\}^{\mathsf{For}}$. Let $F \subseteq X$ and let $\bar{F}$ be the prefix-closure of F:

$$\bar{F} = \{u : \exists_{w \in F}\, u \preceq w\}.$$

Now let $D(F) = \mathrm{Dv}_{CB}(\bar{F})$. The function D defined this way is monotone and Borel: the set of forests is Borel in $\mathcal{D}$ and the property that $u \in \mathrm{Dv}_{CB}(t)$ is a Borel property of a forest t: $u \in \mathrm{dom}(t)$ and $t\restriction_u$ does not have a finite number of branches (this property is Borel because our forests are finitely branching). Also, $D^{\infty}(F) = \emptyset$ if and only if $F \in \{a\}^{\mathsf{ThinFor}}$. By applying Theorem 34.10 we obtain that the rank induced by D (that is the CB-rank) is a $\mathbf{\Pi}^1_1$-rank. □

Our second relation R_Q is intended to witness the existence of a particular skeleton $\tilde{\sigma}$ of a given thin forest t. The trick is that $\tilde{\sigma}$ witnesses a skeleton of t given that t is thin. Otherwise, $\tilde{\sigma}$ does not witness anything interesting. Such a (conditional) skeleton is denoted as a *quasi-skeleton*.

We will encode a subset $\tilde{\sigma} \subseteq \mathrm{dom}(t)$ of nodes of a forest t as its characteristic function — a forest (denoted also $\tilde{\sigma}$) over the alphabet $\{0, 1\}$ such that $\mathrm{dom}(t) = \mathrm{dom}(\tilde{\sigma})$. To simplify the notions we will say that $u \in \mathrm{dom}(t)$ *belongs* to $\tilde{\sigma}$ if u belongs to the set encoded by it (i.e. if $\tilde{\sigma}(u) = 1$).

Proposition 68 *There exists a* $\mathbf{\Sigma}^1_1$ *relation* R_Q *on* $\{a\}^{\mathsf{For}} \times \{0, 1\}^{\mathsf{For}}$ *such that:*

1. *for every pair* $(t, \tilde{\sigma}) \in R_Q$ *we have* $\mathrm{dom}(t) = \mathrm{dom}(\tilde{\sigma})$*, and* $\tilde{\sigma}$ *contains (treated as a set of nodes of* t*) exactly one node from each set of siblings in* t*,*
2. *for every thin forest* t *there exists a forest* $\tilde{\sigma}$ *such that* $(t, \tilde{\sigma}) \in R_Q$*,*
3. *if* t *is a thin forest and* $(t, \tilde{\sigma}) \in R_Q$ *then* $\tilde{\sigma}$ *encodes a skeleton of* t*.*

A forest $\tilde{\sigma}$ such that $(t, \tilde{\sigma}) \in R_Q$ is called a *quasi-skeleton of* t.

Note that R_Q may contain some pairs $(t, \tilde{\sigma})$ with a non-thin forest t. In that case $\tilde{\sigma}$ encodes some set of nodes of t but not a skeleton.

We define $R_Q \subseteq \{a\}^{\mathsf{For}} \times \{0, 1\}^{\mathsf{For}}$ as the set of pairs $(t, \tilde{\sigma})$ such that:

- $\mathrm{dom}(\tilde{\sigma}) = \mathrm{dom}(t)$,
- for every set of siblings in t exactly one of them is in $\tilde{\sigma}$,
- for every node u of t such that $u \in \tilde{\sigma}$ and every sibling u' of u we have
$$(t\restriction_{u'}, t\restriction_u) \in R_E, \tag{19}$$
i.e. the subtree under any sibling of u *embeds* into the subtree under u.

Fact 69 *Since* R_E *is analytic and analytic sets are closed under countable intersections, the relation* R_Q *is also analytic.*

The following two lemmas prove Items 2 and 3 of Proposition 68.

Lemma 70 *Let* t *be a thin forest. There exists a quasi-skeleton* $\tilde{\sigma}$ *for* t*.*

Proof Let t be a thin forest. Let $\tilde{\sigma}$ contain, from every set of siblings $u_1, \ldots, u_n$ in t, exactly one of them that has maximal CB-rank: if $u_i \in \tilde{\sigma}$ then for every $j \neq i$ we have $\mathrm{rank}^{\mathrm{CB}}(t\restriction_{u_i}) \geq \mathrm{rank}^{\mathrm{CB}}(t\restriction_{u_j})$. Clearly $\tilde{\sigma}$ defined this way satisfies (19). □

Lemma 71 *If* t *is a thin forest and* $\tilde{\sigma}$ *is a quasi-skeleton of* t *then* $\tilde{\sigma}$ *(treated as a set of nodes of* t*) is a skeleton of* t*.*

Proof Take any infinite branch π of t. We need to show that almost all nodes on π belong to $\tilde{\sigma}$. Assume contrary. Let $u_0 \prec u_1 \prec \ldots \prec u$ be the sequence of nodes on u that do not belong to $\tilde{\sigma}$. By the definition of $\tilde{\sigma}$ for every node u_i there is a sibling u_i' of u_i such that $u_i' \in \tilde{\sigma}$ and $(t{\restriction}_{u_i}, t{\restriction}_{u_i'}) \in R_E$. Since t is thin this property implies that

$$\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_{u_i}) \leq \mathrm{rank}^{\mathrm{CB}}(t{\restriction}_{u_i'}).$$

Since ordinal numbers are well-founded, we can assume without loss of generality that all the ranks $\mathrm{rank}^{\mathrm{CB}}(t{\restriction}_{u_i})$ are equal some ordinal $\eta < \omega_1$. Since $u_i \prec u_{i+1}'$ so we can also assume that for every i we have $\mathrm{rank}^{\mathrm{CB}}(u_i') = \eta$. Let $t_0 = t{\restriction}_{u_0}$ and let t_0' be the spine of t_0. Note that $\mathrm{rank}^{\mathrm{CB}}(t_0) = \eta$ so by the definition t_0' contains all the nodes of CB-rank η in t. In particular t_0' contains all nodes u_i and u_i'. But this is a contradiction, since $u \in B_{CB}$, it cannot contain infinitely many branching nodes. □

Remark 1 Assume that t is a thin forest, $\tilde{\sigma}$ is a quasi-skeleton of t, and $u \in \mathrm{dom}(t)$ is a node of t. The main branch of $\tilde{\sigma}$ from u can be defined in the same way as in the case of skeletons (see Definition 13). The only difference is that if $\tilde{\sigma}$ is not a skeleton then not every infinite branch of t is main.

8.3 Topological Properties

In this section we give several results showing that regular languages of thin forests are topologically simpler than generic regular languages of forests.

Theorem 72 *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.

Proof of Theorem 72 Assume that L is a regular language of thin forests. Let $L^c = A^{\mathsf{For}} \setminus L$ be its complement relatively to all forests. L^c is a regular language of forests. Let $\mathcal{A}$ be a forest automaton recognizing L^c. We will write L^c as a sum

$$L^c = \left(A^{\mathsf{For}} \setminus A^{\mathsf{ThinFor}}\right) \cup K.$$

The language K will be defined this way to be analytic and to satisfy the following condition:

$$K \cap A^{\mathsf{ThinFor}} = L^c \cap A^{\mathsf{ThinFor}}.$$

Let K contain those forests t such that there exists a quasi skeleton $\tilde{\sigma}$ and a run ρ of the automaton $\mathcal{A}$ such that for every node $x \in \mathrm{dom}(t)$ the limes superior of priorities of ρ is even along the main branch of $\tilde{\sigma}$ from x.

Observe that K is defined by an existential quantification over forests $\tilde{\sigma} \in \{0,1\}^{\mathsf{For}}$ and runs ρ. The inner properties:

- $\tilde{\sigma}$ is a quasi skeleton for t,

- ρ is a run of $\mathcal{A}$,
- for every node $x \in \mathrm{dom}(t)$ the main branch from x along $\tilde{\sigma}$ is accepting,

are analytic (the later two are in fact Borel). Therefore, K is analytic. Note that we do not express explicitly that ρ is an accepting run.

Observe that if $t \in L^c \cap A^{\mathsf{ThinFor}}$ then $t \in K$: there is some quasi skeleton $\tilde{\sigma}$ for t and there is an accepting run ρ of $\mathcal{A}$. Since ρ is accepting so it is accepting on all main branches of $\tilde{\sigma}$.

What remains is to show that if $t \in K \cap A^{\mathsf{ThinFor}}$ then $t \in L^c$. Take a thin forest $t \in K$. Assume that $\tilde{\sigma}, \rho$ are a quasi skeleton and a run given by the definition of K. Since t is a thin forest, $\tilde{\sigma}$ actually encodes a skeleton $\sigma \subseteq \mathrm{dom}(t)$. We take any infinite branch π of t and show that ρ is accepting along π. By Lemma 14 we know that there is a node $x \in \mathrm{dom}(t)$ such that π is the main branch of σ from x. Therefore, by the definition of K, the run ρ is accepting on π. We use here the fact that the acceptance condition is prefix independent: it is enough to satisfy it from some point on. □

Theorems 46 and 72 imply the following dichotomy or *gap property* in the spirit of [16].

Remark 2 For every regular language of thin forests L exactly one of the following possibilities holds, it can be effectively decided which one:

- L is weak MSO-definable among all forests and lies on a finite level of the Borel hierarchy of A^{For},
- 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 73 *Let X be a Polish topological space, $f : X \to A^{\mathsf{ThinFor}}$ be continuous, and L be a regular language of thin forests. Then $f^{-1}(L)$ is Borel in X.*

Proof Assume that X, f, and L are given as in the statement of the theorem. Observe that $f(X)$ is an analytic subset of A^{For} and is contained in A^{ThinFor}.

The following lemma can be seen as an instance of Boundedness Principle, see [12, Theorem 35.23].

Lemma 74 *There exists $\eta < \omega_1$ such that $f(X) \subseteq A^{\mathsf{ThinFor}\leq\eta}$.*

Proof Assume contrary. In that case we give an analytic definition of A^{ThinFor} among all forests. It contradicts the fact that A^{ThinFor} is co-analytic-complete. The key tool is the relation R_E defined in Section 8.2.

Let us define a set T of forests by the following property: a forest t_1 belongs to T if there exists a forest t_2 in $f(X)$ such that $(t_1, t_2) \in R_E$. The above definition is obviously analytic. We claim that $T = A^{\mathsf{ThinFor}}$.

Take a thin forest $t_1 \in A^{\mathsf{ThinFor}}$. If there were no forest t_2 in $f(X)$ of CB-rank greater than $\mathrm{rank}^{\mathrm{CB}}(t_1)$ then $\eta = \mathrm{rank}^{\mathrm{CB}}(t_1)$ would be a bound on CB-ranks of forests in $f(X)$. But we assumed that there is no such bound, so there must exist such $t_2 \in f(X)$. Since $\mathrm{rank}^{\mathrm{CB}}(t_1) \leq \mathrm{rank}^{\mathrm{CB}}(t_2)$, $(t_1, t_2) \in R_E$ and $t_1 \in T$.

Now consider any forest $t_1 \in T$. Let t_2 be the witness from the definition of T. Since $t_2 \in f(X)$, t_2 is a thin forest, by applying Proposition 64 we obtain that t_1 is also a thin forest. □

What remains is to show the following lemma.

Lemma 75 *For every $\eta < \omega_1$ the language $L \cap A^{\mathsf{ThinFor}\leq\eta}$ is Borel.*

Proof The construction mimics the formulae defined in Section 7. The difference is that instead of writing weak MSO formulae, we inductively prove that corresponding languages are Borel. First let us note the following fact. It can be derived from Fact 67 but it can also be proved directly, without referring to Borel derivatives.

Fact 76 *For every $\eta < \omega_1$ the set $A^{\mathsf{ThinFor}\leq\eta}$ is Borel.*

Proof The induction goes by ordinal numbers η. The limit step is obtained via a countable union of languages of CB-rank smaller than η. The successor step is done as follows. Assume that $A^{\mathsf{ThinFor}\leq\eta}$ is Borel. Consider $d \in \mathbb{N}$, $t \in A^{\mathsf{For}}$ and define the d-layer of t as the (finite) set of nodes $v \in \mathrm{dom}(t)$ such that $|v| = d$. Now X be the set of forests t such that

$$\begin{aligned}
&\text{there exists } n \in \mathbb{N} \text{ such that}\\
&\quad \text{for every } d \geq 0\\
&\qquad \text{all except at most } n \text{ nodes } v \text{ in the } d-\text{layer of } t \text{ satisfy}\\
&\qquad\quad t{\restriction_v} \in A^{\mathsf{ThinFor}\leq\eta}
\end{aligned}$$

By the definition X is a Borel set. Assume that $t \in X$ what is witnessed by $n \in \mathbb{N}$. In that case the forest $\mathrm{Dv}^{\eta}_{CB}(t)$ has at most n infinite branches, so $\mathrm{rank}^{\mathrm{CB}}(t) \leq \eta + 1$. It means that $X \subseteq A^{\mathsf{ThinFor}\leq\eta+1}$.

Now assume that $t \in A^{\mathsf{ThinFor}\leq\eta+1}$. Let t' be the final prefix of t. By Fact 12 we know that there is a global bound on the cardinality of the d-layers of t'. Let n be such a bound. Then $t \in X$ as witnessed by n. It means that $A^{\mathsf{ThinFor}\leq\eta+1} \subseteq X$ and therefore $X = A^{\mathsf{ThinFor}\leq\eta+1}$. □

It means that for every fixed $\eta < \omega_1$ the condition that a given forest has CB-rank η is Borel. Now we can repeat the construction of formulae ψ_m and $\varphi_M(h)$ with φ_m replaced by $\{t : \mathrm{rank}^{\mathrm{CB}}(t) = \eta\}$. In this way we obtain a Borel definition of the set $\{t : \mathrm{rank}^{\mathrm{CB}}(t) = \eta \wedge \alpha(t) = h\}$ for $\eta < \omega_1$ and $h \in H$, as in Proposition 51.

By the above observations

$$f^{-1}(L) = f^{-1}(f(X) \cap L) = f^{-1}\left(A^{\mathsf{ThinFor}\leq\eta} \cap L\right),$$

and the set $A^{\mathsf{ThinFor}\leq\eta} \cap L$ is Borel, so is its preimage.

The following theorem can be seen as complementing Theorem 73.

Theorem 77 *There exists a regular language of thin forests L_W over some 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 $f : X \to {A_W}^{\mathsf{ThinFor}}$ such that $f^{-1}(L_W) = B$.*

The principal concept of the above language is based on a construction proposed in [10].

Proof Let $A_W = \{\vee, \wedge, \bot, \top\}$. A forest $t \in {A_W}^{\mathsf{For}}$ induces a parity game: a position of the game is a node $x \in \mathrm{dom}(t)$. Nodes labelled by $\top$ (resp. $\bot$) are final positions of the game, winning for Eve (resp. Adam). Nodes labelled by $\vee$ (resp. $\wedge$) belong to Eve (resp. Adam). In each round the player possessing the current node selects one of its children and the token is moved to the selected node. If a player cannot perform a move (the current node is a leaf) then she looses. The priority of nodes labelled $\vee$ (resp. $\wedge$) is 1 (resp. 2).

For technical reasons we restrict ourselves to trees — if t is a nonempty tree then the initial position of the game is the unique root of t.

Definition 78 Let L_W be the language of thin trees over the alphabet A_W such that Eve has a strategy in the induced parity game.

Observe that, since the arena is a tree, every strategy is a positional one. A positional strategy can be encoded as a set of nodes of a given forest, so the language L_W is regular.

Lemma 79 *If X is a Polish topological space and $B \subseteq X$ is Borel then there exists a continuous function $f : X \to {A_W}^{\mathsf{ThinFor}}$ such that $f^{-1}(L_W) = B$.*

Proof The proof is inductive on the level of the Borel hierarchy the given set B occupies. Without loss of generality we can assume that X is Cantor space $\{0, 1\}^{\omega}$. If B is a basic clopen subset of X then the function mapping elements of B to $\top \cdot 0$ and elements of $X \setminus B$ to $\bot \cdot 0$ is continuous.

Let B be a countable union (resp. intersection) of simpler sets $B_1, B_2, \ldots$. Let f_i be a reduction of B_i to L_W. Let $a = \vee$ (resp. $a = \wedge$). Take any element $x \in X$ and define

$$p_i = a(\square + f_i(x)), \text{ and } f(x) = p_1 \cdot p_2 \cdot \ldots$$

Consider the possible two cases:

($a = \vee$) A strategy for Eve in the game on $f(x)$ boils down to selecting one of the subtrees $f_i(x)$ and proceeding in this subtree.

($a = \wedge$) A strategy for Eve in the game on $f(x)$ consists of a sequence of strategies for each subtree $f_i(x)$ selected by Adam.

Therefore, Eve can win the game on $f(x)$ if and only if she can win on some $f_i(x)$ (resp. on every $f_i(x)$). Therefore, $f(x) \in L_W$ if and only if $\exists_i\ f_i(x) \in L_W$ (resp. $\forall_i\ f_i(x) \in L_W$). So $f(x) \in L_W$ if and only if $x \in B$. □

Using the structure of the language L_W one can deduce the following corollary.

Corollary 80 *The language L_W cannot be defined in weak MSO 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 weak MSO formula φ over the alphabet $A_W \times \{0,1\}$ such that

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

Proof Assume that L_W would be weak MSO-definable among thin forests. In that case L_W would be of the form $L \cap {A_W}^{\mathsf{ThinFor}}$ for a weak MSO-definable language of forests L. By a standard estimation $L \in \boldsymbol{\Sigma}^0_n({A_W}^{\mathsf{For}})$ for some n. But, by Lemma 79 we can reduce some $\boldsymbol{\Pi}^0_n$-complete language to L — a contradiction.

Now observe that the trees constructed by reductions f from the proof of Lemma 79 have simple canonical skeletons: $\sigma(f(x))$ contains the leftmost node from each set (pair) of siblings. Therefore, the canonical skeletons are weak MSO-definable on trees generated by f, so it does not change anything if we supply them with the given forests. □

Acknowledgments The authors would like to thank Henryk Michalewski for posing a number of motivating problems and questions on the subject.

References

1. Bojańczyk, M., Idziaszek, T.: Algebra for infinite forests with an application to the temporal logic EF. In: CONCUR, pp. 131–145 (2009)
2. Blumensath, A.: Recognisability for algebras of infinite trees. Theor. Comput. Sci. **412**(29), 3463–3486 (2011)
3. Bojańczyk, M.: Effective characterizations of tree logics. In: PODS, pp. 53–66 (2008)
4. Bojańczyk, M., Place, T.: Regular languages of infinite trees that are Boolean combinations of open sets. In: ICALP, pp. 104–115 (2012)
5. Burris, S., Sankappanavar, H.P.: A Course in Universal Algebra. Number 78 in Graduate Texts in Mathematics. Springer-Verlag (1981)
6. Bilkowski, M., Skrzypczak, M.: Unambiguity and uniformization problems on infinite trees. In: Della Rocca, S.R. (ed.) Computer Science Logic 2013 (CSL 2013) volume 23 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 81–100. Dagstuhl, Germany (2013). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik
7. Büchi, J.R.: On a decision method in restricted second-order arithmetic. In: Proceedings 1960 International Congress for Logic, Methodology and Philosophy of Science, pp. 1–11 (1962)
8. Bojańczyk, M., Walukiewicz, I.: Forest algebras. In: Logic and Automata, pp. 107–132 (2008)
9. Finkel, O., Simonnet, P.: On recognizable tree languages beyond the Borel hierarchy. Fundam. Inform. **95**(2–3), 287–303 (2009)
10. Hummel, S., Michalewski, H., Niwiński, D.: On the Borel inseparability of game tree languages. In: STACS, pp. 565–575 (2009)
11. Idziaszek, T.: Algebraic Methods in the Theory of Infinite Trees. PhD thesis, University of Warsaw, 2012. unpublished
12. Kechris, A.: Classical Descriptive Set Theory. Springer-Verlag, New York (1995)

13. Kufleitner, M., Lauser, A.: Languages of dot-depth one over infinite words. In: LICS, pp. 23–32 (2011)
14. Lifsches, S., Shelah, S.: Uniformization and skolem functions in the class of trees. J. Symb. Log. **63**(1), 103–127 (1998)
15. Murlak, F.: The Wadge hierarchy of deterministic tree languages. Logic. Methods Comput. Sci. **4**(4) (2008)
16. Niwiński, D., Walukiewicz, I.: A gap property of deterministic tree languages. Theor. Comput. Sci. **1**(303), 215–231 (2003)
17. Perrin, D., Pin, J.-É.: Infinite Words: Automata, Semigroups, Logic and Games. Elsevier (2004)
18. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Trans. American Math. Soc. **141**, 1–35 (1969)
19. Rabinovich, A., Rubin, S.: Interpretations in trees with countably many branches. In: LICS, pp. 551–560. IEEE (2012)
20. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Inf. Control. **8**(2), 190–194 (1965)
21. Simon, I.: Piecewise testable events. In: Automata Theory and Formal Languages, pp. 214–222 (1975)
22. Skurczyński, J.: The Borel hierarchy is infinite in the class of regular sets of trees. Theor. Comput. Sci. **112**(2), 413–418 (1993)
23. Thomas, W.: Relationen endlicher valenz über der ordnung der natürlichen zahlen. Habilitationsschrift, Universitat Freiburg, apr. 1980
24. Thérien, D., Wilke, T.: Over words, two variables are as powerful as one quantifier alternation. In: STOC, pp. 234–240 (1998)
25. Wilke, T.: An algebraic theory for regular languages of finite and infinite words. Int. J. Alg. Comput. **3**, 447–489 (1993)
26. Wilke, T.: Classifying discrete temporal properties. Habilitationsschrift, Universitat Kiel, apr. 1998