Regular Tree Algebras

Achim Blumensath[1]

Masaryk University Brno

blumens@fi.muni.cz

Abstract

We introduce a class of algebras that can be used as recognisers for regular tree languages. We show that it is the only such class that forms a pseudo-variety and we prove the existence of syntactic algebras. Finally, we give a more algebraic characterisation of the algebras in our class.

1 Introduction

There are many different formalisms to study regular languages, the most prominent ones being automata and logic. In this paper we are interested in the algebraic approach to formal language theory, in the context of infinite trees. Such algebraic methods are particularly successful in deriving decidable characterisations for various fragments of monadic second-order logic. For instance, a theorem of Schützenberger [15] states that a language of finite words is definable in first-order logic if, and only if, its syntactic monoid is finite and aperiodic. The latter condition is decidable as we can compute the syntactic monoid of a regular language and check it for aperiodicity.

Besides a comprehensive algebraic theory for the usual word languages, there also exist well-developed frameworks for languages of infinite words and – to a lesser degree – finite trees. For languages of infinite trees, the combinatorics involved are much more challenging. As a result, the existing theory is still fragmentary. The first preliminary results were provided in [7, 8], with one article considering languages of *regular* trees only, and one considering languages of *thin* trees. The first framework that could deal with arbitrary infinite trees was provided by [4, 5]. Unfortunately, it turned out to be too complicated and technical to be very useful.

In this article we propose an alternative, much simpler approach, and we develop it to a point where it is suitable for devising decision procedures. Because of space considerations we defer the actual applications to a subsequent article [3]. Our first simplification concerns the notation. It turns out that much of the notational overhead of the old framework can be avoided by adopting the category-theoretical formalism of a monad and an Eilenberg-Moore algebra. Our second contribution is in isolating a suitable class of algebras as recognisers of regular languages. While admittedly its definition is rather naïve and not as concrete as one would like it to be, our key insight – and the main contribution of this paper – is the fact that the resulting class has all the required properties: it forms a pseudo-variety and it has syntactic algebras. Furthermore, we prove that it is the only class that does the trick (cf. Corollary 4.4 below).

The overview of this article is as follows. We start in Section 2 with setting up our algebraic framework. In particular, we explain the notion of an Eilenberg–Moore algebra. In Section 3 we isolate the property ('regularity') we need for a tree algebra to recognise regular languages only, and we give a first characterisation of when an algebra has this property. While both the definition and our characterisation are rather abstract, we show that the resulting class is the only possible one that satisfies all our requirements: we prove in Section 4 that it is the only class with the desired closure properties; and in Section 5 we prove the existence of syntactic algebras, a prerequisite for characterisation results. We conclude in Section 6 with a second, more specific characterisation of regularity for tree algebras.

2   Tree algebras

A convenient algebraic formalism for the various kinds of language theories has turned out to be one based on the category-theoretical notions of a *monad* and an *Eilenberg-Moore algebra* [6]. To make this article accessible to readers without a category-theoretical background we refrain from using category-theoretical terminology where possible and use elementary definitions instead. Readers familiar with category theory should be able to translate our results into their language.

To prepare the reader for our notion of a tree algebra, let us take a look at semigroups first. Instead of using the usual binary product, we can see a semigroup as a set S equipped with a product $\pi : S^+ \to S$ of variable arity that multiplies an arbitrary sequence of semigroup elements in one step. Analogously, we will define a tree algebra as a set A together with a product $\pi : \mathbb{T}A \to A$ that takes an A-labelled tree and returns a single element of A. Let us make this idea precise.

First of all, we will not work with simple sets but with *ranked sets,* that is, sets where each element has an *arity* or *rank.* Formally, we consider such a set as a sequence $A = (A_n)_{n<\omega}$ where A_n is the subset of elements of arity n. A function $f : A \to B$ of such sets is then a family $f = (f_n)_{n<\omega}$ of functions $f_n : A_n \to B_n$. We will frequently identify a ranked set $A = (A_n)_n$ with its disjoint union $A = \biguplus_n A_n$. A function $(f_n)_n : (A_n)_n \to (B_n)_n$ then corresponds to a rank preserving function $f : A \to B$. In the rest of this article all sets will be assumed to be ranked, if not explicitly stated otherwise, and all functions will be rank preserving.

Now let A be a ranked set. An *A-labelled* tree t is a (finite or infinite) tree where every vertex is labelled by an element from A in such a way that the arity of a label matches the number of successors. We set $\mathbb{T}A = (\mathbb{T}_n A)_{n<\omega}$ where $\mathbb{T}_n A$ is the set of all $(A \cup \{x_0, \ldots, x_{n-1}\})$-labelled trees t where the additional labels x_i are called *variables.* These are considered as having arity 0 and we require that

- each variable x_i occurs at most once in the tree t and
- the root is not labelled by any variable.

(We will always assume that $x_i \notin A$.) Note that a tree t containing the variables, say, x_0, x_3, and x_7, can be regarded as an element of $\mathbb{T}_8 A$, of $\mathbb{T}_9 A$, and so on. According to our convention

these elements are considered to be different. For a tree t, we denote its set of vertices by $\mathrm{dom}(t)$, and we write $t(v)$ for the label of a vertex $v \in \mathrm{dom}(t)$. We say that two trees *have the same shape* if there become equal when we remove all non-variable labels.

▶ Remark. There is some freedom in choosing how to define $\mathbb{T}A$. Instead of requiring that every variable occurs at most once, we could allow each occurring several, even infinitely many times. We also could allow the use of infinitely many different variables by adding elements of arity ω. Finally, we could require that every variable appears at least once. For most of our results, these details do not matter. Hence, the precise definition is more of a matter of taste. But let us mention that some results in Section 6 fail if we allow multiple occurrences of the same variable. In particular, this is the case for Proposition 6.5.

To write down trees concisely we use the usual term notation. For instance, $a(x_3, c)$ denotes the tree where the root is labelled by a and its two successors by x_3 and c, respectively. Another useful piece of notation is the following one. Given a (rank-preserving) function $f : A \to B$ we denote by $\mathbb{T}f : \mathbb{T}A \to \mathbb{T}B$ the function that applies f to every (non-variable) label of the input tree.

Now we can define a *tree algebra* $\mathfrak{A} = \langle A, \pi \rangle$ as a ranked set A together with a product $\pi : \mathbb{T}A \to A$ that satisfies certain *associativity laws.* Before stating these laws formally let us again take a look at semigroups. For a function $\pi : S^+ \to S$ to be the product associated with a semigroup it has to satisfy two conditions. First of all, we require that the product is the identity on singletons, that is,

$$\pi(\langle a \rangle) = a, \quad \text{for every } a \in S.$$

Let us call this the *unit law.* Secondly, if we factorise a product in different ways, we always get the same result. That means, for a sequence of sequences $w = \langle w_0, \ldots, w_{m-1} \rangle \in (S^+)^+$, we require that

$$\pi\big(\pi(w_0), \ldots, \pi(w_{m-1})\big) = \pi(w_0 \ldots w_{m-1}).$$

Writing $\pi^+ : (S^+)^+ \to S^+$ for the function that multiplies each component of the given sequence and $\mathrm{flat} : (S^+)^+ \to S^+$ for the concatenation function, we can write this equation in the compact form

$$\pi \circ \pi^+ = \pi \circ \mathrm{flat}.$$

This is the *associative law.*

$$
\begin{array}{ccc}
(S^+)^+ & \xrightarrow{\ \mathrm{flat}\ } & S^+ \\
\ \downarrow{\scriptstyle \pi^+} & & \downarrow{\scriptstyle \pi} \\
S^+ & \xrightarrow[\ \pi\ ]{} & S
\end{array}
$$

Introducing the corresponding auxiliary functions for trees, we can write similar laws for a tree algebra $\mathfrak{A} = \langle A, \pi \rangle$:

$$\pi \circ \mathrm{sing} = \mathrm{id}$$

and $\pi \circ \mathbb{T}\pi = \pi \circ \mathrm{flat}$.

$$
\begin{array}{ccc}
\mathbb{T}\mathbb{T}A & \xrightarrow{\ \mathrm{flat}\ } & \mathbb{T}A \\
\ \downarrow{\scriptstyle \mathbb{T}\pi} & & \downarrow{\scriptstyle \pi} \\
\mathbb{T}A & \xrightarrow[\ \pi\ ]{} & A
\end{array}
$$

Here, the *singleton function* $\mathrm{sing} : A \to \mathbb{T}A$ maps a label $a \in A_n$ to the *singleton tree* $a(x_0, \ldots, x_{n-1}) \in \mathbb{T}_n A$, and the *flattening function* $\mathrm{flat} : \mathbb{T}\mathbb{T}A \to \mathbb{T}A$ takes a tree t whose vertices v are labelled by trees $t(v)$ from $\mathbb{T}A$ and returns the tree obtained by simultaneously
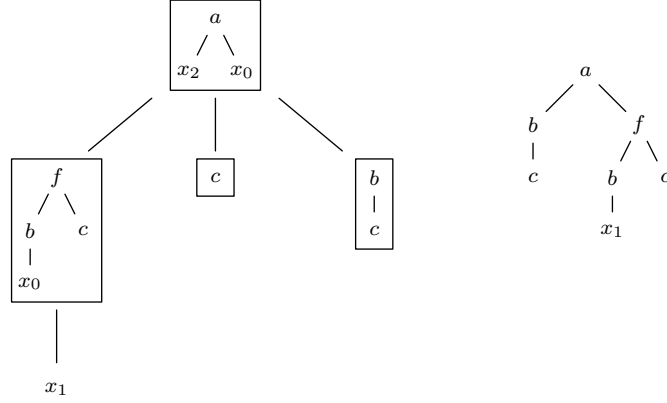
■ **Figure 1** The flattening operation

substituting in $t(v)$ each variable x_i by the tree associated with the corresponding successor of v. In more detail, we compute $\mathrm{flat}(t)$ as follows. We start with the disjoint union of all trees $t(v)$, for $v \in \mathrm{dom}(t)$. We then remove every leaf of (the copy of) $t(v)$ that is labelled by a variable x_i, and replace it with an edge to the root of the corresponding copy of $t(u_i)$, where u_i is the $(i+1)$-th successor of v. Of the resulting forest, we take the connected component containing the root $t(\langle\rangle)$. This is the value of the flattening $\mathrm{flat}(t)$. For instance, in Figure 1 the tree on the left evaluates to the tree on the right.

To summarise let us give the formal definition.

▶ **Definition 2.1.** (a) A *tree algebra* is a pair $\mathfrak{A} = \langle A, \pi \rangle$ consisting of a ranked set A and a *product* $\pi : \mathbb{T}A \to A$ which satisfy

$$\pi \circ \mathrm{sing} = \mathrm{id} \quad \text{and} \quad \pi \circ \mathbb{T}\pi = \pi \circ \mathrm{flat}\,.$$

(b) A *morphism* of tree algebras is a function $f : A \to B$ between their domains commuting with the respective products:

$$f \circ \pi = \pi \circ \mathbb{T}f\,.$$

An important example of a tree algebra is a free one. Given a ranked set Σ, the *free tree algebra* over Σ is $\langle \mathbb{T}\Sigma, \mathrm{flat}\rangle$. (The facts that this is indeed a tree algebra and that it has the desired universal property follow from a general category-theoretical result on monads; see e.g. Proposition 4.1.4 of [10].)

Let us next explain how to use a tree algebra to recognise tree languages. For the purpose of this article, a *tree language* is a subset $L \subseteq \mathbb{T}_m\Sigma$ where Σ is a finite ranked set and the arity $m < \omega$ is fixed. A tree language $L \subseteq \mathbb{T}_m\Sigma$ is called *regular* if it is recognised by a nondeterministic tree automaton with the parity condition, or it is definable by a formula of monadic second-order logic, see [17]. (For $m > 0$, the automaton or formula treats the variables as distinguished letters in the leaves.) Such a language is *recognised* by a morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ if there is a subset $P \subseteq A_m$ such that $L = \varphi^{-1}[P]$. In this case, we also say that the algebra $\mathfrak{A}$ recognises L.

As an example, let us construct a tree algebra recognising the set of all trees $t \in \mathbb{T}_0\{a, b\}$ that contain the label a at least once. For every (part of an) input tree, we have to remember one bit of information: whether or not it contains the label a. This suggests to have two elements, say 0_m and 1_m, for each arity m. When constructing arbitrary products of such elements, we obtain additional elements that are of the form $c(x_i, x_j, \ldots, x_k)$ where $c \in \{0_m, 1_m\}$ and $i < j < \cdots < k < m$. (As it is important to know which variables appear in a term, we cannot simply identify these with 0_m and 1_m.) Thus, the domain A_m for arity m of our algebra $\mathfrak{A}$ will consist of all these elements and the recognising morphism maps a term $t \in \mathbb{T}_m\{a, b\}$ to the element $c(x_i, x_j, \ldots, x_k)$ where c specifies whether or not t contains the label a and $x_i, x_j, \ldots, x_k$ are the variables that actually appear in t.

3 Regular tree algebras

The goal of this paper is to find a class $\mathcal{C}$ of tree algebras that *characterises* the class of regular tree languages in the sense that a tree language is regular if, and only if, it is recognised by some algebra from $\mathcal{C}$. One obvious condition we have to impose on such a class is that all algebras $\mathfrak{A} \in \mathcal{C}$ are *finitary,* which means that

- $\mathfrak{A}$ is finitely generated (i.e., there is a finite set $C \subseteq A$ such that every element $a \in A$ can be written as a product of some tree in $\mathbb{T}C$) and
- for every $n < \omega$, there are only finitely many elements of arity n.

Unfortunately, this in itself is not enough. There are examples of finitary tree algebras that recognise non-regular languages [9].

A naïve way to obtain the desired class of algebras is to take the class of all tree algebras that only recognise regular languages. This is obviously the largest class that will do (if any exists at all). The problem with this definition is that it is not very enlightening: we have no idea of what these algebras look like. We will nevertheless adopt this naïve approach for its simplicity. A more satisfying, but also much more complicated, alternative definition will be provided in Section 6 below.

By looking at what it means to only recognise regular languages, we arrive at the following definition.

▶ **Definition 3.1.** A tree algebra $\mathfrak{A} = \langle A, \pi \rangle$ is *regular* if it is finitary and there exists a finite set $C \subseteq A$ of generators such that, for every element $a \in A$, the preimage

$\pi^{-1}(a) \cap \mathbb{T}C$ is a regular language.

Note that an alternative way to write the set in the above definition is $(\pi \restriction \mathbb{T}C)^{-1}(a)$, where $\pi \restriction X$ denotes the restriction of π to the set X. This will come in handy in several of the proofs below.

Before showing that our definition has the desired effect, let us mention that it does not depend on the choice of the set C of generators.

▶ **Lemma 3.2.** *Let $\mathfrak{A}$ be a regular tree algebra and $D \subseteq A$ a finite set. Then*

$\pi^{-1}(a) \cap \mathbb{T}D$ *is regular,*   *for all $a \in A$.*

Proof. For each $d \in D$, we fix some term $t \in \mathbb{T}C$ with $\pi(t) = d$. This defines a function $s_0 : D \to \mathbb{T}C$ such that $\pi \circ s_0 = \mathrm{id}$. We can extend s_0 to a morphism $s : \mathbb{T}D \to \mathbb{T}C$ by setting

$s(t) = \mathrm{flat}(\mathbb{T}s_0(t))\,.$

For $t \in \mathbb{T}D$, it follows that

$$(\pi \circ s)(t) = (\pi \circ \mathrm{flat} \circ \mathbb{T}s_0)(t) = (\pi \circ \mathbb{T}\pi \circ \mathbb{T}s_0)(t)$$
$$= (\pi \circ \mathbb{T}(\pi \circ s_0))(t) = (\pi \circ \mathbb{T}\mathrm{id})(t) = \pi(t),$$

which implies that $\pi \upharpoonright \mathbb{T}D = (\pi \upharpoonright \mathbb{T}C) \circ s$. For $a \in A$, we therefore have

$$(\pi \upharpoonright \mathbb{T}D)^{-1}(a) = s^{-1}[(\pi \upharpoonright \mathbb{T}C)^{-1}(a)].$$

By assumption the set $(\pi \upharpoonright \mathbb{T}C)^{-1}(a) = \pi^{-1}(a) \cap \mathbb{T}C$ is regular. As regular languages are closed under inverse morphisms (see Lemma A.1), so is $(\pi \upharpoonright \mathbb{T}D)^{-1}(a)$. ◄

Let us now establish the rather obvious fact that our definition does what it is supposed to.

▶ **Theorem 3.3.** *A finitary tree algebra is regular if, and only if, all languages recognised by it are regular.*

Proof. ($\Leftarrow$) Suppose that $\mathfrak{A}$ is not regular. Then there exists a finite set $C \subseteq A$ and an element $a \in A$ such that the preimage $L := \pi^{-1}(a) \cap \mathbb{T}C$ is not regular. Consequently, the restriction $\pi \upharpoonright \mathbb{T}C : \mathbb{T}C \to \mathfrak{A}$ of the product is a morphism that recognises a non-regular language L.

($\Rightarrow$) Suppose that $\mathfrak{A}$ is regular and let $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ be a morphism recognising the language $L := \varphi^{-1}[P]$ with $P \subseteq A_m$. Set $C := (\varphi \circ \mathrm{sing})[\Sigma]$. By Lemma 3.2, each preimage

$$K_a := \pi^{-1}(a) \cap \mathbb{T}C, \quad \text{for } a \in A,$$

is regular. Hence, so is the (finite) union $K := \bigcup_{a \in P} K_a$. Let $i : C \to A$ be the inclusion map, $\pi_0 := \pi \upharpoonright \mathbb{T}C : \mathbb{T}C \to A$ the restriction of the product, and set $\varphi_0 := \varphi \circ \mathrm{sing} : \Sigma \to C$. It follows that

$$\varphi \circ \mathrm{sing} = i \circ \varphi_0 = \pi_0 \circ \mathrm{sing} \circ \varphi_0 = \pi_0 \circ \mathbb{T}\varphi_0 \circ \mathrm{sing}.$$

Since $\mathbb{T}\Sigma$ is generated by the range of sing, this implies that $\varphi = \pi_0 \circ \mathbb{T}\varphi_0$. Hence,

$$L = \varphi^{-1}[P] = (\pi_0 \circ \mathbb{T}\varphi_0)^{-1}[P] = (\mathbb{T}\varphi_0)^{-1}[\pi_0^{-1}[P]] = (\mathbb{T}\varphi_0)^{-1}[K].$$

As regular languages are closed under inverse projections (see Lemma A.1), it therefore follows that L is regular. ◄

Conversely one can prove that every regular language is recognised by some regular tree algebra.

▶ **Theorem 3.4.** *A tree language is regular if, and only if, it is recognised by a regular tree algebra.*

One direction follows immediately from Theorem 3.3. For the other one, we have to construct a regular tree algebra recognising a given regular language L.

We start by fixing notation and collecting a few basic definitions (more details can be found in, e.g., [12]). We work with non-deterministic parity automata of the form $\mathcal{A} = \langle Q, \Sigma, \Delta, q_0, \Omega \rangle$, where Q is the (unranked) set of states, Σ the (ranked) input alphabet, q_0 the initial state, $\Omega : Q \to \omega$ a priority function, and $\Delta \subseteq Q \times \Sigma \times Q^*$ the transition relation. Each transition $\langle q, a, p_0, \dots, p_{n-1} \rangle$ consists of the current state q, the current letter a, and states $p_0, \dots, p_{n-1}$ for the successors. For leaves, the letter a has arity 0 and the transition simply takes the form $\langle q, a \rangle$.

A *partial run* of $\mathcal{A}$ on some input tree $t \in \mathbb{T}\Sigma$ is a labelling $\varrho : \mathrm{dom}(t) \to Q$ of the tree such that

- there are arbitrary states at vertices carrying a variable x_i,
- the labelling respects the transition relation Δ at all other vertices, and
- every infinite branch satisfies the parity condition.

The *profile* of a partial run ϱ is the tuple

$$\langle q, k_0, p_0 \dots, k_{m-1}, p_{m-1} \rangle \,,$$

where q is the state at the root of t, p_i the state at the vertex carrying the variable x_i, and k_i the minimal priority seen along the path from the root to this vertex. If there is no vertex labelled x_i, we set $k_i := \bot$ and $p_i := \bot$, for some special bottom symbol $\bot$.

We aim to construct a tree algebra where the elements encode sets of possible profiles, i.e., sets of possible behaviours of $\mathcal{A}$ on a given input tree. To simplify the definition and accommodate the material in Section 6 below, we will construct an algebra that is slightly larger than necessary: instead of using only the usual profiles of $\mathcal{A}$, we will work with *partial* ones, i.e., profiles where we only specify data for some of the variables. Formally, this can be done by labelling the paths to the variables by elements of a suitable ω-semigroup (for the definition of an ω-semigroup, see, e.g., [13, Section 4.1]). The construction is performed in three steps. (We keep the presentation rather informal. More details can be found in [2].)

(i) We denote by $\mathfrak{S}_{\mathcal{A}} = \langle S, S_\omega \rangle$ the (partial) ω-semigroup where

- $S := Q \times D \times Q$ contains all triples of the form $\langle p, k, q \rangle$ for states $p, q \in Q$ and a priority k (D is the set of priorities used by $\mathcal{A}$) and
- $S_\omega := Q$ contains the states of $\mathcal{A}$.

A triple $\langle p, k, q \rangle \in S$ encodes a finite path of a run that starts in state p, ends in state q, and has minimal priority k. A state $p \in S_\omega$ encodes an infinite branch that starts in state p and satisfies the parity condition.

The product is defined naturally: if we multiply two triples $\langle p, k, q \rangle$ and $\langle p', k', q' \rangle$ with matching states $p' = q$, the result is $\langle p, \min(k, k'), q' \rangle$. If $p' \neq q$, the product remains undefined. Similarly, the product of $\langle p, k, q \rangle$ and $p' \in S_\omega$ evaluates to $p \in S_\omega$, provided that $q = p'$. Otherwise, it is again undefined. Finally, an infinite product of a sequence $\langle p_i, k_i, q_i \rangle_{i < \omega}$ produces the state p_0, provided that $q_i = p_{i+1}$ for all i and the parity condition

$$\liminf_{i \to \infty} \Omega(p_i) \text{ is even}$$

is satisfied.

(ii) Next we turn $\mathfrak{S}_{\mathcal{A}}$ into a tree algebra where the elements of arity m are of the form $a(x_i)$ or b with $a \in S$, $b \in S_\omega$, and $i < m$. The product is induced by the ω-semigroup

product: given a tree t labelled by elements of this form, we construct a branch by starting at the root and proceeding downwards as follows. If the label of the current vertex is $a(x_i)$, we continue with the $(i + 1)$-th successor. If it is of the form $b \in S_\omega$, we stop. This process yields a sequence of elements of the ω-semigroup, which we can multiply to a new element c. If the chosen branch ends in a variable x_k, we return $c(x_k)$, otherwise we simply return c. We leave the product of t undefined, if the product of the ω-semigroup elements is not defined.

(iii) Finally, we formally close the tree algebra constructed in (ii) first under conjunctions, and then under disjunctions, that is, the new elements are formal expressions of the from $\bigvee_i \bigwedge_k a_{ik}$ where the elements a_{ik} all have the same arity. We define a product of such elements by requiring that disjunction and conjunctions commute with the product operation. If a product of basic elements is undefined, we treat it as an empty conjunction.

Let $\mathfrak{A}$ be the tree algebra constructed in (iii). (It is straightforward, but rather tedious, to check that $\mathfrak{A}$ is indeed a tree algebra, i.e., that the product is associative. The interested reader can find a full proof in [2].)

To show that $\mathfrak{A}$ recognises $L(\mathcal{A})$, note that conjunctions of ω-semigroup elements can be used to encode partial profiles of $\mathcal{A}$ and disjunctions of such conjunctions sets of partial profiles. Hence, we consider the morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ that maps a tree t to the disjunction $\bigvee_\varrho \tilde{\varrho}$ where ϱ ranges over all partial runs of $\mathcal{A}$ on t and $\tilde{\varrho}$ is an element encoding the run ϱ defined as follows. Let q be the state at the root, $(v_i)_i$ an enumeration of all vertices of t with a variable, $(p_i)_i$ the corresponding sequence of states, $x_{m(i)}$ the variable at v_i, and let k_i be the minimal priority on the path from the root to v_i. We set

$$\tilde{\varrho} := q \wedge \bigwedge_i \langle q, k_i, p_i \rangle (x_{m(i)}) \,.$$

It follows that $\mathcal{A}$ accepts a tree t if, and only if,

$$\varphi(t) \geq q_0 \wedge \bigwedge_i \langle q_0, k_i, p_i \rangle (x_{m(i)}) \,,$$

for some $p_i, k_i, m(i)$ such that, when starting in state p_i, the automaton $\mathcal{A}$ accepts the singleton tree with label $x_{m(i)}$. (The ordering $\geq$ here is the one induced by the conjunctions and disjunctions. In case of the above formula it simply means that $\varphi(t)$ is a disjunction where one of the terms is a conjunction that contains the right-hand side as a subconjunction.) Consequently, we can find a set $P \subseteq A$ such that

$$L(\mathcal{A}) = \varphi^{-1}[P] \,,$$

as desired.

Finally, let $\mathfrak{A}_0 \subseteq \mathfrak{A}$ be the image of φ. We claim that $\mathfrak{A}_0$ is the desired regular tree algebra. We already have a morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}_0$ recognising L. Hence, it remains to show that $\mathfrak{A}_0$ is regular. Clearly, $\mathfrak{A}_0$ is generated by the finite set $C := (\varphi \circ \text{sing})[\Sigma]$. Consider an element $\bigvee_i \bigwedge_k a_{ik}$. To check that a product $\pi(t)$ with $t \in \mathbb{T}C$ evaluates to this value we have to select, for every conjunction $\bigwedge_k a_{ik}$ and every vertex v of t, some term of the disjunction $t(v)$. Then we have to multiply the corresponding ω-semigroup elements along every branch of t and check that the result is equal to the corresponding element a_{ik}. This process can clearly be performed by a tree automaton.

4 Closure properties

So far, we have done nothing deep. The interesting realisation is that our naïve definition is actually sufficient for applications: the class of regular algebras has all the desired closure

properties and it allows the computation of syntactic algebras. We start by taking a look at the closure properties. Syntactic algebras are the topic of Section 5.

Recall that a *variety* is a class $\mathcal{C}$ of algebras that is closed under the operations of taking: (i) H homomorphic images (i.e., quotients), (ii) S subalgebras, and (iii) P arbitrary products. Equivalently, this can be written as the equation $\mathcal{C} = \mathsf{HSP}(\mathcal{C})$. It follows from the axioms that every variety is also closed under directed colimits (see, e.g., Remark 3.6 (6) of [1]). Furthermore, the famous Variety Theorem of Birkhoff states that varieties are exactly those classes of algebras that can be defined by systems of equations (see, e.g., Theorem 3.9 of [1]).

If we are interested in classes of finite algebras only, one has to adapt these definitions slightly. Since the product operation P can produce infinite algebras, we replace it by the operation P_ω of taking finite products only. This leads to the definition of a *pseudo-variety,* which is a class $\mathcal{C}$ satisfying $\mathcal{C} = \mathsf{HSP}_\omega(\mathcal{C})$. For classes of finite algebras, closure under directed colimits is trivial. There is also a variant of the Birkhoff Variety Theorem by Reiterman [14] that characterises pseudo-varieties as exactly those classes that can be defined by a system of *profinite* equations.

In our setting with infinitely many sorts, we are interested in classes of *finitary* algebras, and we are again forced to slightly modify the definitions. The problem is that subalgebras and finite products of finitary algebras are not necessarily finitely generated (we will provide counterexamples below). Therefore we replace S by the operation S_ω of taking finitely-generated subalgebras only and we require closure under $\mathsf{HS}_\omega\mathsf{P}_\omega$. As closure under directed colimits is not automatic anymore we also have to add it as an extra requirement. In fact a slightly weaker condition suffices: closure under *rank-limits.* We say that a tree algebra $\mathfrak{A}$ is the rank-limit of a sequence $(\mathfrak{B}_n)_{n<\omega}$ of tree algebras if, for every $m < \omega$, the algebras $\mathfrak{A}$ and $\mathfrak{B}_n$, for $n \geq m$, are isomorphic if we restrict them to elements of arity at most m. Note that closure under rank-limits is a rather natural condition. For instance, it is satisfied by every class axiomatised by a set of equations. One can show that, for classes of finitary algebras that are closed under quotients, closure under rank-limits and under directed colimits are equivalent.

▶ **Definition 4.1.** A *pseudo-variety* of tree algebras is a class $\mathcal{C}$ of finitary tree algebras that is closed under (i) quotients, (ii) finitely generated subalgebras of finite products, and (iii) rank-limits.

We start by showing that the regular tree algebras form a pseudo-variety. At the moment it is open whether there is an analogue to the Theorem of Reiterman in our setting. There is a general result by Milius and Urbat [11] which provides variety theorems for many classes, including the class of regular tree algebras. But it uses a rather abstract notion of an equation and it remains to be worked out how exactly such equations look like in our case.

Before continuing, let us introduce a bit of notation concerning rank-limits. First, for a ranked set A and an arity k, we set $A_{<k} := A_0 \cup \cdots \cup A_{k-1}$. We consider $A_{<k}$ as a ranked set that has no elements of arity k or higher. For our functor $\mathbb{T}$ we similarly set $\mathbb{T}_{<k}A := (\mathbb{T}A_{<k})_{<k}$. Finally, for a tree algebra $\mathfrak{A} = \langle A, \pi \rangle$ we denote by $\mathfrak{A}|_{<k}$ the algebra with domain $A_{<k}$ and product $\pi \upharpoonright \mathbb{T}_{<k}A : \mathbb{T}_{<k}A \to A_{<k}$. Note that $\mathfrak{A}|_{<k}$ is not a tree algebra (an algebra for the functor $\mathbb{T}$) as the product is not of the right form. Instead it is an algebra for the functor $\mathbb{T}_{<k}$ (a $\mathbb{T}_{<k}$-algebra is defined by the same two laws as a tree algebra, except that we replace the functor $\mathbb{T}$ by $\mathbb{T}_{<k}$ throughout). With this notation we can say that $\mathfrak{A}$ is a rank-limit of $(\mathfrak{B}^n)_n$ if

$$\mathfrak{A}|_{<k} \cong \mathfrak{B}^n_{<k}, \quad \text{for } k \leq n \leq \omega,$$

where the isomorphism is understood as a $\mathbb{T}_{<k}$-algebra isomorphism.

▶ **Theorem 4.2.** *The class of regular tree algebras forms a pseudo-variety.*

Proof. The proof is straightforward. We have to show that the class of regular tree algebras is closed under

(a) finitely generated subalgebras of finite products,
(b) rank-limits,
(c) quotients.

(a) The empty product has exactly one element 1_m for each arity m. Given a subalgebra $\mathfrak{A}$ generated by some finite set C and some element $1_m \in A$, we have

$$\pi^{-1}(1_m) \cap \mathbb{T}C = \mathbb{T}_m C \,,$$

which is regular.

Hence, it remains to consider a finitely generated subalgebra $\mathfrak{A}$ of a non-empty, finite product $\prod_{i<n} \mathfrak{B}^i$. Let $C \subseteq A$ and $D^i \subseteq B^i$ be finite sets of generators. Increasing the D^i if necessary, we may assume that $C \subseteq \prod_i D^i$. Let $p_i : \prod_i B^i \to B^i$ be the projections. For $t \in \mathbb{T}\prod_i D^i$ and $\bar{a} = (a_i)_i \in A \subseteq \prod_i B^i$, we have

$$\pi(t) = \bar{a} \quad \text{iff} \quad \pi(\mathbb{T}p_i(t)) = a_i \quad \text{for all } i \,.$$

As the $\mathfrak{B}^i$ are regular, it follows that

$$\pi^{-1}(\bar{a}) \cap \prod_i \mathbb{T}D^i = \bigcap_i (\mathbb{T}p_i)^{-1}\big(\pi^{-1}(a_i) \cap \mathbb{T}D^i\big)$$

is regular. Since regular languages are closed under intersection, the preimage

$$\pi^{-1}(\bar{a}) \cap \mathbb{T}C = \pi^{-1}(\bar{a}) \cap \prod_i \mathbb{T}D^i \cap \mathbb{T}C$$

is also regular.

(b) Let $(\mathfrak{B}^n)_n$ be a sequence of regular tree algebras with rank-limit $\mathfrak{A}$. To show that $\mathfrak{A}$ is regular, let $C \subseteq A$ be a finite set of generators and $a \in A$. Fix a number $k < \omega$, such that

$$C \cup \{a\} \subseteq A_{<k} = B^n_{<k} \,, \quad \text{for } n \geq k \,.$$

As $\mathfrak{B}^k$ is regular, the preimage $\pi^{-1}(a) \cap \mathbb{T}C$ is regular.

(c) Let $\varphi : \mathfrak{A} \to \mathfrak{B}$ be a surjective morphism of tree algebras and suppose that $\mathfrak{A}$ is regular. We have to show that $\mathfrak{B}$ is also regular. Fix a finite set $C \subseteq A$ of generators and set $D := \varphi[C]$. Increasing C if necessary we may assume that $C = \varphi^{-1}[D]$.

First, note that $\mathfrak{B}$ is finitely generated by D. Furthermore,

$$t = \mathbb{T}\varphi(s) \quad \text{implies} \quad \pi(t) = \pi(\mathbb{T}\varphi(s)) = \varphi(\pi(s)) \,, \quad \text{for } s \in \mathbb{T}C \text{ and } t \in \mathbb{T}D \,.$$

Hence, for $b \in B_m$,

$$\begin{aligned}
\pi^{-1}(b) \cap \mathbb{T}D &= \{\, t \in \mathbb{T}D \mid \pi(t) = b \,\} \\
&= \{\, \mathbb{T}\varphi(s) \mid s \in \mathbb{T}C \,, \ \varphi(\pi(s)) = b \,\} \\
&= \mathbb{T}\varphi\big[\bigcup \{\, \pi^{-1}(a) \cap \mathbb{T}C \mid a \in \varphi^{-1}(b) \,\}\big]
\end{aligned}$$

Since $\varphi^{-1}(b) \subseteq A_m$ is a finite set, the above union is finite and, therefore, regular. As regular languages are closed under projections, so is its image under $\mathbb{T}\varphi$. ◀

As mentioned above, the definition of a regular tree algebra does not tell us what these algebras look like. The next theorem sheds a bit more light on this question. A less abstract characterisation will be given in Section 6. To state the theorem, we need the notion of a *finitary sub-quotient* of a tree algebra $\mathfrak{A}$. By definition this is an algebra which can be obtained from a finitary subalgebra of $\mathfrak{A}$ by taking a quotient. Recall that we say that a class $\mathcal{C}$ *characterises the regular languages* if a language is regular if, and only if, it is recognised by some algebra from $\mathcal{C}$.

▶ **Theorem 4.3.** *Let $\mathcal{C}$ be an arbitrary class of finitary tree algebras that characterises the regular languages and that is closed under finite products. A finitary tree algebra $\mathfrak{A}$ is regular if, and only if, it is the rank-limit of a sequence of finitary sub-quotients of algebras in $\mathcal{C}$.*

Proof. ($\Leftarrow$) Let $(\mathfrak{B}^n)_{n<\omega}$ be a sequence of algebras in $\mathcal{C}$ and let $\mathfrak{D}^n$ be a sub-quotient of $\mathfrak{B}^n$ such that $(\mathfrak{D}^n)_{n<\omega}$ converges to $\mathfrak{A}$. As $\mathcal{C}$ characterises the regular tree languages, every algebra in $\mathcal{C}$ is regular. Since the regular tree algebras are closed under finitely generated subalgebras and quotients, it follows that each $\mathfrak{D}^n$ is regular. Finally, so is the limit $\mathfrak{A}$ since the class of regular algebras is closed under rank-limits.

($\Rightarrow$) Suppose that $\mathfrak{A}$ is regular. Let $C \subseteq A$ be a finite set of generators and choose a number $k < \omega$ such that $C \subseteq A_{<k}$. We construct a sequence $(\mathfrak{B}^n)_{n<\omega}$ of algebras in $\mathcal{C}$ and sub-quotients $\mathfrak{D}^n$ of $\mathfrak{B}^n$ such that $(\mathfrak{D}^n)_{n<\omega}$ converges to $\mathfrak{A}$.

Let $n \le \omega$. For each $a \in A_{<n}$, we choose an algebra $\mathfrak{B}_a \in \mathcal{C}$ and a morphism $\varphi_a : \mathbb{T}C \to \mathfrak{B}_a$ recognising $\pi^{-1}(a) \cap \mathbb{T}C$. Set

$$\mathfrak{B}^n := \prod_{a \in A_{<n}} \mathfrak{B}_a \quad \text{and} \quad \varphi := \langle \varphi_a \rangle_{a \in A_{<n}} : \mathbb{T}C \to \mathfrak{B}^n \,.$$

Let $\mathfrak{D}' \subseteq \mathfrak{B}^n$ be the subalgebra induced by the set $D' := \operatorname{rng} \varphi$. Note that $\mathfrak{D}'$ is finitely generated by $\varphi[C]$. We will show that

$$\varphi(s) = \varphi(t) \quad \text{implies} \quad \pi(s) = \pi(t) \,, \quad \text{for } s, t \in \mathbb{T}_{<n}C \,.$$

Then it follows by standard arguments that there exists a function $\psi : D'_{<n} \to A_{<n}$ satisfying $\psi \circ \varphi \upharpoonright \mathbb{T}_{<n}C = \pi \upharpoonright \mathbb{T}_{<n}C$. As φ and π are morphisms of $\mathbb{T}_{<n}$-algebras, so is ψ. And since $\pi \upharpoonright \mathbb{T}_{<n}C$ is surjective, so is ψ. Consequently, $\psi : \mathfrak{D}|_{<n} \to \mathfrak{A}|_{<n}$ is a morphism. Let $\mathfrak{D}^n := \mathfrak{D}'/\ker \psi$ where $\ker \psi$ denotes the equivalence relation of 'having the same image under ψ'. Then $\mathfrak{D}^n$ is a sub-quotient of $\mathfrak{B}^n$ and $\mathfrak{D}^n|_{<n} \cong \mathfrak{A}|_{<n}$. Consequently, $(\mathfrak{D}^n)_{n<\omega}$ is a sequence of finitary sub-quotients that converges (up to isomorphisms) to $\mathfrak{A}$.

It remains to prove the claim. Let $s, t \in \mathbb{T}_{<n}C$ be trees with $\varphi(s) = \varphi(t)$. By construction there exist sets $P_a \subseteq D'$, for $a \in A_{<n}$, such that

$$\pi^{-1}(a) \cap \mathbb{T}C = \varphi^{-1}[P_a] \,.$$

It follows that

$$\pi(s) = a \quad \text{iff} \quad \varphi(s) \in P_a \quad \text{iff} \quad \varphi(t) \in P_a \quad \text{iff} \quad \pi(t) = a \,.$$

Hence, $\pi(s) = \pi(t)$, as desired. ◀

It follows by Theorems 3.3 and 3.4 that the class of regular tree algebras is the largest class that characterises the regular languages. From the preceding theorem we can now conclude that it is in fact the only pseudo-variety with this property. This means that the notion of a regular tree algebra is quite canonical, although we still would like to have a more concrete definition.

▶ **Corollary 4.4.** *The class of regular tree algebras is the only pseudo-variety characterising the class of regular tree languages.*

Proof. We have already shown that the class of regular tree algebras forms a pseudo-variety. For uniqueness, let $\mathcal{C}$ be any pseudo-variety characterising the regular tree languages. Then every algebra in $\mathcal{C}$ is regular. Conversely, let $\mathfrak{A}$ be a regular tree algebra. By Theorem 4.3, there exist algebras $\mathfrak{B}_n \in \mathcal{C}$ and finitary sub-quotients $\mathfrak{D}_n$ of $\mathfrak{B}_n$, for $n < \omega$, such that $\mathfrak{A}$ is the rank-limit of $(\mathfrak{D}_n)_n$. As $\mathcal{C}$ is a pseudo-variety, it follows that every $\mathfrak{D}_n$ belongs to $\mathcal{C}$ and, therefore, also the limit $\mathfrak{A}$. ◀

Our definition of a pseudo-variety was complicated by the fact that the class of finitary tree algebras is not closed under subalgebras and finite products. Here we present two examples showing that a subalgebra or a finite product of regular tree algebras need not be finitely generated.

(a) Let us start with subalgebras. We use a result by Yanov and Muchnik [18] about so-called *clones*. A clone $\mathfrak{C}$ is a set of functions (of various arities) over some fixed set X that contains all projections and that is closed under composition, i.e., if $\mathfrak{C}$ contains $f : X^n \to X$ and $g_0, \ldots, g_{n-1} : X^m \to X$, it also contains the m-ary function

$$\bar{x} \mapsto f(g_0(\bar{x}), \ldots, g_{n-1}(\bar{x})).$$

Note that this composition also makes sense if the functions $g_0, \ldots, g_{n-1}$ have different arities since we can make their arities equal by composing them by suitable projections (which are in $\mathfrak{C}$ by assumption).

▶ **Theorem 4.5** (Yanov–Muchnik). *There are uncountably many clones on a three element set.*

As there are only countably many finitely generated clones, it follows in particular that there exists some clone $\mathfrak{C}$ that is not finitely generated. We will use it to construct the desired tree algebra.

Let $[3] = \{0, 1, 2\}$ be a three element set and let A_n be the set of all functions $[3]^n \to [3]$ together with a special error value $\bot$. We turn $A = (A_n)_n$ into a tree algebra by defining the following multiplication $\pi : \mathbb{T}A \to A$. For a finite tree $t \in \mathbb{T}A$ that does not contain the symbol $\bot$, we compute the product $\pi(t)$ by composing all the functions that label the vertices of t. For all other trees, we set $\pi(t) := \bot$. The resulting structure $\mathfrak{A} = \langle A, \pi \rangle$ forms a tree algebra which is finitely generated. (To see the latter, one can, e.g., represent every 3-valued function in a similar way as boolean functions can be written in disjunctive normal form.) Furthermore, $\mathfrak{A}$ is even regular since, when evaluating a tree t an automaton is able to first check that t is finite and does not contain $\bot$, and then evaluate t bottom up by remembering where each (of the bounded number) of the input arguments is mapped to.

To conclude the construction recall that we have seen above that there exists a clone on $[3]$ that is not finitely generated. Let $\mathfrak{C} \subseteq \mathfrak{A}$ be the subalgebra of $\mathfrak{A}$ consisting of the elements of that clone. Then $\mathfrak{C}$ is not finitely generated.

(b) Our counterexample for products looks as follows. We start with a tree algebra $\mathfrak{B}$ where the elements of arity n are all finite sequences in $\{x_0, \ldots, x_{n-1}\}^*$ that contain every variable at most once. We define the product as follows. Suppose we have sequences $\alpha \in B_m$ and $\beta_0, \ldots, \beta_{m-1} \in B_n$ where the β_i are disjoint. If $\alpha = \langle x_{i_0}, \ldots, x_{i_{k-1}} \rangle$, we set

$$\alpha(\beta_0, \ldots, \beta_{m-1}) := \beta_{i_0} \ldots \beta_{i_{k-1}},$$

i.e., we substitute β_i for x_i in α. For a finite tree $t \in \mathbb{T}B_n$, we can now inductively define

$$\pi(t) = \alpha(\pi(s_0), \dots, \pi(s_{m-1})),$$

where $\alpha := t(\langle\rangle)$ is the label at the root and $s_0, \dots, s_{m-1}$ are the attached subtrees. (With the convention that $\pi(s_i) = \langle x_k \rangle$ in case that $s_i = x_k$ is a single variable.)

We can extend this definition to infinite trees as follows. If t does not contain variables, we set $\pi(t) = \langle\rangle$. Otherwise, we choose a finite prefix s of t that contains all the variables, separately compute the products of s and of the attached subtrees, and then multiply the results as above. Note that this definition ensures that $\pi(t)$ is the sequence of all variables appearing in t, but not necessarily in the order they appear in t.

Again it is straightforward to check that $\mathfrak{B}$ is a tree algebra. Furthermore, note that we can write every sequence $\alpha \in B_m$ as the product of a tree t where all internal vertices are labelled by $\langle x_0 \rangle$ or $\langle x_0, x_1 \rangle$ by suitably choosing the ordering of the variables of t. Hence, $\mathfrak{B}$ is finitely generated by three elements $\langle\rangle, \langle x_0 \rangle, \langle x_0, x_1 \rangle$.

Furthermore, $\mathfrak{B}$ is regular since, given an element $b \in B_n$ and a finite set of generators, an automaton can determine whether an input tree evaluates to b since all intermediate results are sequences of length at most n.

We claim that the product $\mathfrak{B} \times \mathfrak{B}$ is not finitely generated. For a contradiction suppose otherwise and fix a finite set C of generators. Choose a number m that is greater than the arity of all elements in C. We consider the element $\langle \alpha, \beta \rangle \in B_{2m} \times B_{2m}$ where

$$\alpha := \langle x_0, \dots, x_{2m-1} \rangle$$
$$\beta := \langle x_m, x_0, x_{m+1}, x_1, \dots, x_{m+i}, x_i, \dots, x_{2m-1}, x_{m-1} \rangle.$$

By assumption, there is a tree t with product $\langle \alpha, \beta \rangle$. Let $\langle \gamma, \delta \rangle$ be the label at the root of t and let $s_0, \dots, s_{n-1}$ be the subtrees attached to it. (For simplicity, we assume that $n > 1$. Otherwise our proof needs to be slightly modified.) By choice of m, there is some subtree s_i that contains at least two variables. Let $\sigma, \tau : [n] \to [n]$ be the permutations such that

$$\gamma = \langle x_{\sigma(0)}, \dots, x_{\sigma(n-1)} \rangle \quad \text{and} \quad \delta = \langle x_{\tau(0)}, \dots, x_{\tau(n-1)} \rangle,$$

and let $p : \mathfrak{B} \times \mathfrak{B} \to \mathfrak{B}$ be the projection to the first component. By looking at the first components, we see that

$$\pi\big(\mathbb{T}p(s_{\sigma(0)})\big) \dots \pi\big(\mathbb{T}p(s_{\sigma(n-1)})\big) = \gamma\big(\pi(\mathbb{T}p(s_0)), \dots, \pi(\mathbb{T}p(s_{n-1}))\big)$$
$$= \alpha = \langle x_0, \dots, x_{2m-1} \rangle.$$

Consequently, there exist numbers $k < l$ such that the term s_i contains the variables $x_k, x_{k+1}, \dots, x_{l-1}$. By choice of i, we have $l \geq k + 2$.

Looking at the second components, we see that β must have some segment of length $l - k \geq 2$ which contains the variables $x_k, x_{k+1}, \dots, x_{l-1}$ (in any order). But the only segments of β of this form are those of length 1 and the one of length $2m$. A contradiction.

5 Syntactic algebras

Besides being a pseudo-variety we also need our class of recognisers to have what is called *syntactic algebras.* These are algebras recognising a given language that are minimal in a certain sense. Usually we can obtain such an algebra by taking a suitable quotient of the free algebra. In this section we will show that for tree algebras the situation is exactly the same. Let us start with some basic definitions.

A *congruence* for a tree algebra $\mathfrak{A}$ is an equivalence relation $\approx$ on its universe A that is compatible with the product in the sense that, if $s, t \in \mathbb{T}A$ are two trees of the same shape such that $s(v) \approx t(v)$, for all v, then $\pi(s) \approx \pi(t)$. If $\approx$ is a congruence, we can define a tree algebra structure on the quotient $A/\approx$ in the natural way. We denote it by $\mathfrak{A}/\approx$.

A *tree with a hole,* or a *context,* is a tree $t \in \mathbb{T}(A \cup \square)$ where the new symbol $\square$ is called the *hole.* It works as a kind of variable, but with the difference that it can have an arbitrary (but fixed) arity and that it can appear several times in t. Note that we allow $\square$ to have positive arity, which means that it can occur in a non-leaf position in the tree. Given such a context t and an element a of the right arity, we denote by $t[a]$ the product $\pi(t')$ where t' is the tree obtained from t by replacing all labels $\square$ by a.

▶ **Definition 5.1.** Let $\mathfrak{A}$ be a tree algebra and $P \subseteq A_n$ a set of elements of arity n. The *syntactic congruence* for P is defined by

$$a \approx_P b \quad :\text{iff} \quad t[a] \in P \Leftrightarrow t[b] \in P, \quad \text{for all contexts } t \in \mathbb{T}_n(A \cup \square).$$

The non-obvious part of this definition is the fact that the resulting equivalence relation is indeed a congruence. In fact, the proof of the next result crucially relies on the fact that the tree algebra in question is regular. For arbitrary tree algebras the statement is simply false.

▶ **Theorem 5.2.** *The syntactic congruence on a regular tree algebra is a congruence.*

For the proof, we need to set up a bit of technical machinery. Fix a finite ranked set Σ, let $\sim$ be an equivalence relation on $\mathbb{T}\Sigma$, and let $\mathcal{A}$ and $\mathcal{B}$ be two non-deterministic parity automata. We will define a game $\mathcal{G}_\sim(\mathcal{A}, \mathcal{B})$ where the first player wins if, and only if, there exist two trees $S, T \in \mathbb{T}\mathbb{T}\Sigma$ of the same shape such that

- $S(v) \sim T(v)$, for all vertices v,
- $\mathcal{A}$ accepts $\mathrm{flat}(S)$,
- $\mathcal{B}$ accepts $\mathrm{flat}(T)$.

The game is a variant of the well-known Automaton–Pathfinder Game. The only difference is that we simulate two automata at the same time and that, instead of playing single letters, we play larger trees in each step. The game has two players *Automaton* and *Pathfinder.* Each round starts in a position of the form $\langle p, q \rangle$, where p is a state of $\mathcal{A}$ and q one of $\mathcal{B}$. We allow p and/or q to be undefined, which we denote by the special symbol $\bot$. In the first round of the game, p and q are the initial states of the respective automata. Given such a position $\langle p, q \rangle$,

- Automaton chooses two trees $s, t \in \mathbb{T}_m \Sigma$ with $s \sim t$ and $m < \omega$,
- if $p \neq \bot$, he also chooses a profile δ for some partial run of $\mathcal{A}$ on s that starts in state p, and
- if $q \neq \bot$, he chooses a profile ε for some partial run of $\mathcal{B}$ on t that starts in state q.

Pathfinder responds by selecting a number $i < m$. The *outcome* of this round is the pair $\langle \delta|_i, \varepsilon|_i \rangle$ where $|_i$ denotes the restriction to the $(i + 1)$-th successor, that is,

$$\langle r, k_0, p_0, \ldots, k_{m-1}, p_{m-1} \rangle\big|_i := \langle r, k_i, p_i \rangle.$$

If there is no δ, we set $\delta|_i := \langle \bot, \bot, \bot \rangle$ and similarly for ε. If this outcome is $\langle p, k, p' \rangle, \langle q, l, q' \rangle$, the next round of the game will start in the position $\langle p', q' \rangle$.

If at some point in the game one of the players cannot make his choice, that player loses the game. Otherwise, the players produce an infinite sequence $\langle \delta_0, \varepsilon_0 \rangle, \langle \delta_1, \varepsilon_1 \rangle, \ldots$ of

outcomes. Let k_i be the priority in δ_i and l_i the priority in ε_i. Player Automaton wins the game if each of the sequences $k_0, k_1, \ldots$ and $l_0, l_1, \ldots$ either satisfies the parity condition or if it contains the symbol $\bot$. Otherwise, Pathfinder wins.

Clearly, if there are two trees $S, T \in \mathbb{TT}\Sigma$ of the same shape such that

- $S(v) \sim T(v)$, for all vertices v,
- $\mathcal{A}$ accepts $\mathrm{flat}(S)$, and
- $\mathcal{B}$ accepts $\mathrm{flat}(T)$,

then Automaton has the following winning strategy in $\mathcal{G}_\sim(\mathcal{A}, \mathcal{B})$. He fixes two accepting runs ϱ and σ on, respectively, $\mathrm{flat}(S)$ and $\mathrm{flat}(T)$. During the game he descends through the trees S and T. When the game reaches a vertex v, Automaton chooses the trees $S(v)$ and $T(v)$ and the profiles of the subruns of ϱ and σ that correspond to the trees $S(v)$ and $T(v)$, respectively.

Conversely, if Automaton has a winning strategy in the game, we can use it to construct

- two trees $S, T \in \mathbb{TT}\Sigma$ such that $S(v) \sim T(v)$ for all v and
- accepting runs of $\mathcal{A}$ and $\mathcal{B}$ on, respectively, $\mathrm{flat}(S)$ and $\mathrm{flat}(T)$.

Proof of Theorem 5.2. Let $\mathfrak{A}$ be a regular tree algebra and let $C \subseteq A$ be a finite set of generators. For a contradiction, suppose that there exists a subset $P \subseteq A_m$ such that $\approx_P$ is not a congruence. Then we can find two trees $s, t \in \mathbb{T}A$ (of the same shape) such that

$$s(v) \approx_P t(v), \quad \text{for all vertices } v, \quad \text{but} \quad \pi(s) \not\approx_P \pi(t).$$

For every vertex v, we can choose trees $S(v), T(v) \in \mathbb{T}C$ such that $s(v) = \pi(S(v))$ and $t(v) = \pi(T(v))$. This defines two trees $S, T \in \mathbb{TT}C$ with $s = \mathbb{T}\pi(S)$ and $t = \mathbb{T}\pi(T)$.

As the algebra $\mathfrak{A}$ is regular and every $\approx_P$-class $[a]$ is finite (by definition, $\approx_P$ only relates elements of the same arity), we can construct automata $\mathcal{A}_a$, for $a \in A$, such that

$$L(\mathcal{A}_a) = \left\{ t \in \mathbb{T}C \mid \pi(t) \approx_P a \right\} = \bigcup_{b \approx_P a} (\pi^{-1}(b) \cap \mathbb{T}C).$$

Let $a := \pi(s)$ and $b := \pi(t)$. We consider the game $\mathcal{G}_{\approx_P}(\mathcal{A}_a, \mathcal{A}_b)$. The trees S and T show that Automaton has a winning strategy in this game. As the winning condition of the game is regular, we can apply the Büchi–Landweber Theorem, which tells us that Automaton even has a finite-memory winning strategy. Since the choice of $S(v)$ and $T(v)$ by Automaton in the game only depends on the current position $\langle p, q \rangle$ and on the contents of the memory, there are only finitely many different values for $S(v)$ and $T(v)$. This implies that there are only finitely many different labels used by the trees s and t. Consequently, we can get from s to t by a finite number of steps in each of which we replace several occurrences of a single label of s by the corresponding label of t. Thus, there exists a sequence $u_0, \ldots, u_n \in \mathbb{T}A$ such that $s = u_0$, $t = u_n$, and each u_{i+1} is obtained from u_i by replacing (several occurrences of) a single label by an $\approx_P$-equivalent one, i.e., $u_i = r_i[a_i]$ and $u_{i+1} = r_i[b_i]$, for a suitable context $r_i \in \mathbb{T}(A + \square)$ and elements $a_i \approx_P b_i$ in A. By induction on i, it now follows that $\pi(u_i) \approx_P \pi(s)$. For $i = 0$, this is trivial; and for $i > 0$ it is sufficient to note that $a_{i-1} \approx_P b_{i-1}$ implies

$$p[\pi(u_{i-1})] = p[r_{i-1}[a_{i-1}]] \in L \iff p[\pi(u_i)] = p[r_{i-1}[b_{i-1}]] \in L, \quad \text{for all contexts } p.$$

Consequently, we have $\pi(s) \approx_P \pi(t)$. A contradiction. ◀

As a consequence we obtain the same statement for free algebras, provided that the given subset is a regular language.

▶ **Corollary 5.3.** *Let $L \subseteq \mathbb{T}_n \Sigma$ be a regular language. Then the syntactic congruence for L is a congruence on $\mathbb{T}\Sigma$.*

Proof. Let $L \subseteq \mathbb{T}_n \Sigma$ be regular. Then there exists a regular tree algebra $\mathfrak{A}$ and a morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ such that $L = \varphi^{-1}[P]$ for some set $P \subseteq A_n$. By Theorem 5.2, the syntactic congruence $\approx_P$ of P is a congruence. The claim now follows from two facts that are both straightforward to prove:

(1) $L = \varphi^{-1}[P]$ implies that

$$s \approx_L t \quad \text{iff} \quad \varphi(s) \approx_P \varphi(t) \,.$$

(2) If $\sim$ is a congruence of $\mathfrak{A}$ then

$$a \sim_\varphi b \quad :\text{iff} \quad \varphi(a) \sim \varphi(b)$$

is a congruence of $\mathbb{T}\Sigma$.   ◀

For a regular language $L \subseteq \mathbb{T}_n \Sigma$, we call the quotient $\mathbb{T}\Sigma/\approx_L$ the *syntactic algebra* of L. An immediate consequence of the way we have defined $\approx_L$ is that the syntactic algebra is minimal in the sense that the projection $\mathbb{T}\Sigma \to \mathbb{T}\Sigma/\approx_L$ factorises through every morphism $\mathbb{T}\Sigma \to \mathfrak{A}$ that recognises L.

▶ **Theorem 5.4.** *The syntactic algebra of a regular tree language L is regular and it is the smallest tree algebra recognising L.*

Proof. Let $L \subseteq \mathbb{T}_n \Sigma$ be regular and let $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ be a morphism recognising it. Replacing $\mathfrak{A}$ by the image of φ we may assume that φ is surjective. We start by constructing a morphism $\psi : \mathfrak{A} \to \mathbb{T}\Sigma/\approx_L$ such that $\psi \circ \varphi = q$, where $q : \mathbb{T}\Sigma \to \mathbb{T}\Sigma/\approx_L$ is the quotient map.

$$
\begin{array}{ccc}
\mathbb{T}\Sigma & \xrightarrow{\ \varphi\ } & \mathfrak{A} \\
 & \searrow^{q} & \downarrow^{\psi} \\
 & & \mathbb{T}\Sigma/\approx_L
\end{array}
$$

To do so it is sufficient to prove that

$$\varphi(s) = \varphi(t) \quad \text{implies} \quad s \approx_L t \,.$$

Then we can define $\psi(a) := q(t)$, for some $t \in \varphi^{-1}(a)$. By the above implication, ψ is well-defined. Furthermore, it is straightforward to check that this function is in fact a morphism of tree algebras.

Hence, it remains to prove the claim. Suppose that $\varphi(s) = \varphi(t)$. To show that $s \approx_L t$ consider a context r with $r[s] \in L$. Then $\varphi(r[s]) \in P := \varphi[L]$. Let $r' := \mathbb{T}\varphi(r)$. Then

$$\varphi(r[t]) = r'[\varphi(t)] = r'[\varphi(s)] = \varphi(r[s]) \in P \,,$$

which implies that $r[t] \in L$, as desired.

To conclude the proof, we have to show that the syntactic algebra $\mathbb{T}\Sigma/\approx_L$ is regular. Fix a morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ into a regular algebra recognising L. We have just shown that there exists a morphism $\psi : \mathfrak{A} \to \mathbb{T}\Sigma/\approx_L$ with $\psi \circ \varphi = q$. As the quotient map q is surjective, so is ψ. Hence, $\mathbb{T}\Sigma/\approx_L$ is a quotient of $\mathfrak{A}$ and, therefore, regular by Theorem 4.2.   ◀

We have just proved the existence of syntactic algebras in our framework. If we want to use our theory to develop decidable characterisations of logical fragments, we further require an algorithm to actually compute these algebras. Before presenting one we need to explain how to represent a regular tree algebra to an algorithm. The problem is that, while finitary, a regular tree algebra still has infinitely many elements. So we cannot simply write down its multiplication table. What we do instead is to use an algorithm that, given an arity $n < \omega$, produces a (finite) list of automata, one for each language of the form $\pi^{-1}(a) \cap \mathbb{T}C$ for $a \in A_n$. Using this representation, we can then algorithmically construct and process regular tree algebras.

▶ **Theorem 5.5.** *Given a regular language $L \subseteq \mathbb{T}_n\Sigma$, we can compute the syntactic algebra $\mathbb{T}\Sigma/\approx_L$.*

Proof. Let $L \subseteq \mathbb{T}_n\Sigma$ be regular and $\mathcal{A}$ an automaton for L. Using the construction from the proof of Theorem 3.4, we can compute a regular tree algebra $\mathfrak{A}$, a morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$, and a set $P \subseteq A_n$ such that $L = \varphi^{-1}[P]$. By Theorem 5.4, it follows that

$$\mathbb{T}\Sigma/\approx_L \;\cong\; \mathfrak{A}/\approx_P .$$

Hence, it is sufficient to prove that the relation $\approx_P$ is decidable. Note that

$$a \not\approx_L b \quad \text{iff} \quad \text{there exists some context } t \text{ with } (t[a] \in L \Leftrightarrow t[b] \notin L) .$$

We will prove the decidability of the latter condition.

Let $C \subseteq A$ be a finite set of generators of $\mathfrak{A}$. W.l.o.g. we may assume that all labels of the term t we are looking for (except for the hole $\square$) are in C. As $\pi^{-1}(a) \cap \mathbb{T}C$ is regular, there exists a regular tree $u \in \mathbb{T}C$ with $\pi(u) = a$. Similarly, we can find a regular tree $v \in \mathbb{T}C$ with $\pi(v) = b$. Let m be the arity of a and b and fix finite graphs G and H whose unravellings are, respectively, u and v. Given G we can compute the set U of all tuples $\langle p, q_0, \ldots, q_{m-1} \rangle$ such that there exists a partial run ϱ of $\mathcal{A}$ on the tree u such that

- ϱ starts in state p,
- the leaf with the variable x_i has state q_i, and
- every infinite branch satisfies the parity condition.

Similarly, we can compute an analogous set V for the tree v. Given these two sets we can then construct an automaton $\mathcal{B}$ that reads a context s and checks whether the original automaton $\mathcal{A}$ accepts the tree $s[u]$, but does not accept $s[v]$, or vice versa. It follows that

$$a \not\approx_P b \quad \text{iff} \quad L(\mathcal{B}) \neq \emptyset ,$$

a condition that is decidable.   ◀

As an example of how to use syntactic algebras let us derive a characterisation of the class of *commutative* tree languages. For space reasons, we defer more substantial applications to a forthcoming article [3]. We say that a tree s is a *permutation* of the tree t if s is obtained from t by rearranging the successors of every vertex. Formally, we call a function $\sigma : \mathrm{dom}(s) \to \mathrm{dom}(t)$ a *permutation* if it is bijective and it preserves the successor and sibling relations. Then s is a *permutation* of t if there exists some permutation $\mathrm{dom}(s) \to \mathrm{dom}(t)$. A language $L \subseteq \mathbb{T}\Sigma$ is *commutative* if it is closed under permutations. Note that this is not the same as saying that L is closed under rearranging the successors of a single vertex (or finitely many of them).

▶ **Theorem 5.6.** *A regular tree language $L \subseteq \mathbb{T}_n\Sigma$ is commutative if, and only if, its syntactic algebra $\mathfrak{A}$ satisfies the equations*

$$a(x_0, \ldots, x_{m-1}) = a(x_{\sigma(0)}, \ldots, x_{\sigma(m-1)}),$$

for all $a \in A_m$, $m < \omega$, and all permutations $\sigma : [m] \to [m]$.

Proof. ($\Leftarrow$) Note that the quotient morphism $\varphi : \mathbb{T}\Sigma \to \mathfrak{A}$ recognises L. If s is a permutation of t, we have $\varphi(s) = \varphi(t)$. Hence, $s \in L \Leftrightarrow t \in L$ and L is commutative.

($\Rightarrow$) Fix an element $a \in A_m$ and a permutation $\sigma : [m] \to [m]$. We have to show that

$$a(x_0, \ldots, x_{m-1}) \approx_L a(x_{\sigma(0)}, \ldots, x_{\sigma(m-1)}).$$

Hence, let r be a context. Note that the two trees obtained from r by replacing the hole $\square$ by, respectively, $a(x_0, \ldots, x_{m-1})$ and $a(x_{\sigma(0)}, \ldots, x_{\sigma(m-1)})$ are permutations of each other. As L is commutative we therefore have

$$r[a(x_0, \ldots, x_{m-1})] \in L \Leftrightarrow r[a(x_{\sigma(0)}, \ldots, x_{\sigma(m-1)})] \in L. \qquad \blacktriangleleft$$

Note that it follows in particular that commutativity is decidable. Given a regular language L, we can compute its syntactic algebra and check whether it satisfies the above equations. (We only need to check them for elements a in a finite set of generators.)

6 Deterministic tree algebras

In Theorem 4.3 we have provided a characterisation of regular tree algebras in terms of an unspecified second class $\mathcal{C}$ of algebras that characterises the regular languages. We can obtain a more informative result by making a concrete choice for $\mathcal{C}$. In this section we will consider one such class. In order to make this second class as simple as possible, we allow the relationship between the two classes to be more complicated that what we had in Theorem 4.3.

In addition, the characterisation we obtain in Theorem 6.11 below can also serve as an alternative definition of the notion of a regular tree algebra. It has the advantage that it is purely algebraic and does not refer to automata, avoiding the apparent circularity of the original definition. Its main disadvantage is that it is more complicated and abstract, which is why we did not adopt it as the official definition.

For the definition, we need to work with *ordered* algebras. An ordered tree algebra $\mathfrak{A} = \langle A, \pi, \leq \rangle$ consists of a tree algebra $\langle A, \pi \rangle$ that is expanded by a partial order $\leq$ on A such that the product π is monotone. (We order $\mathbb{T}A$ componentwise: $s \leq t$ if the trees s and t have the same shape and each label of s is less than or equal to the corresponding label of t.) Such an ordering is *complete* if it has arbitrary joins and meets (i.e., suprema and infima). Morphisms of ordered tree algebras are assumed to preserve the ordering and morphisms of completely ordered algebras are assumed to also preserve joins and meets. The class of algebras we are considering in this section is the following one.

▶ **Definition 6.1.** Let $\mathfrak{A}$ be a completely ordered tree algebra.

(a) $\mathfrak{A}$ is *meet-continuous* if products commute with meets, that is, given a tree $T \in \mathbb{T}\mathcal{P}(A)$ labelled by subsets of A, we have

$$\pi(\mathbb{T}\inf(T)) = \inf \{ \pi(t) \mid t(v) \in T(v) \text{ for all vertices } v \}.$$

(b) An element $a \in A_n$ is *rectangular* if it can be written as a meet of elements of arity 0 and elements of the form $b(x_i)$, for $b \in A_1$ and $i < n$.

(c) $\mathfrak{A}$ is *deterministic* if it is meet-continuous and all elements are rectangular.

The motivating example for a deterministic algebra is one arising from an automaton in the following way.

▶ **Definition 6.2.** Let $\mathcal{A}$ be a tree automaton and let $\mathfrak{A}$ be the tree algebra constructed at the end of Section 3. The *transition algebra* $\mathfrak{T}(\mathcal{A})$ of $\mathcal{A}$ is the subalgebra of $\mathfrak{A}$ whose elements are conjunctions of semigroup elements plus the empty disjunction $\bot$, i.e., we omit all disjunctions with more than one term. We consider $\mathfrak{T}(\mathcal{A})$ an ordered algebra where the ordering is the one induced by the conjunctions and disjunctions.

▶ **Lemma 6.3.** *The transition algebra $\mathfrak{T}(\mathcal{A})$ is deterministic.*

Proof. By definition, every element is a meet (conjunction) of elements of arity 0 or elements of the form $a(x_i)$ where a has arity 1. Thus, all elements are rectangular. For meet-continuity it is sufficient to note that, in every tree algebra constructed from an ω-semigroup as in the proof of Theorem 3.4, the subalgebra consisting of the one-element disjunctions is meet-continuous (see Proposition 4.12 (a) of [2]). ◀

Deterministic algebras are a very special case of regular tree algebras. One can show that their expressive power corresponds to a certain form of deterministic tree automata. For our purposes, it is sufficient to know that they are regular.

▶ **Proposition 6.4.** *Every finitary subalgebra of a deterministic tree algebra is regular.*

Proof. Let t be a tree we want to multiply. As every label of t is rectangular, we can use meet-continuity to transform the product of t into a meet of products where every label has arity at most one. Such products correspond to ω-semigroup products along a single branch of t (see Lemma 4.23 (b) of [2]). This is something an automaton can evaluate. Consequently, in order to check whether t evaluates to a given element a an automaton can compute all the products along the branches of t, take their infimum, and compare it to a. ◀

Let us use deterministic algebras to give a second characterisation of the regular algebras. We start with an observation that simplifies proofs of regularity: we only need to check elements of arity at most one.

▶ **Proposition 6.5.** *A finitary tree algebra $\mathfrak{A}$ is regular if, and only if, it has a finite set $C \subseteq A$ of generators such that*

$$\pi^{-1}(a) \cap \mathbb{T}C \text{ is regular}, \quad \text{for every } a \in A \text{ of arity at most } 1.$$

Before giving the proof, we need to collect a few results about factorisations. A *factorisation* of a tree $t \in \mathbb{T}A$ is a tree $T \in \mathbb{T}\mathbb{T}A$ such that $\mathrm{flat}(T) = t$. We denote by $\mathcal{F}(t)$ the set of all factorisations T of t such that the trees $T(v)$ are singletons for all vertices v of T with more than one successor. The *height* of a factorisation T is the height of the tree T.

We call a tree $t \in \mathbb{T}A$ *reduced* if it has no non-trivial factor of arity at most one, that is, for every factorisation T of t and every vertex $v \in \mathrm{dom}(T)$ of arity at most one, we have $T(v) = \mathrm{sing}(a)$, for some $a \in A$. The important fact about reduced trees is that they are small.

▶ **Lemma 6.6.** *Let $\mathfrak{A}$ be a tree algebra and $m < \omega$. Every reduced tree $t \in \mathbb{T}_m A$ has height at most $2m$.*

Proof. We prove the claim by induction on m. For $m = 0$, note that every reduced tree of arity m is of the form $\mathrm{sing}(a)$, for some $a \in A$. Hence, the height is 0. For the inductive step, suppose that $m > 0$ and consider a reduced tree $t \in \mathbb{T}_m A$. We distinguish two cases.

First, suppose that the root has an arity greater than 1. As t is reduced, every subtree attached to the root must have fewer variables than t. By inductive hypothesis, their height is at most $2(m-1)$. Hence, the height of t is at most $2(m-1)+1$.

It remains to consider the case where the root has arity 1. As t is reduced, the successor must then have arity greater than 1. Hence, the attached subtree satisfies the above case, which means that its height is bounded by $2(m-1)+1$. Consequently, the height of t is at most $2(m-1)+2 = 2m$. ◀

Next we will show that the set $\mathcal{F}(t)$ of factorisations of t contains reduced trees. For the proof we will employ the following ordering on $\mathcal{F}(t)$. For $S, T \in \mathcal{F}(t)$, we set

$$S \sqsubseteq T \quad :\text{iff} \quad \text{there is some } U \in \mathbb{TTT}A \text{ such that } S = \mathrm{flat}(U) \text{ and}$$
$$\text{every } U(v) \text{ is a factorisation of } T(v)\,, \text{ for } v \in \mathrm{dom}(U)\,.$$

▶ **Lemma 6.7.** *The set $\mathcal{F}(t)$ is inductively ordered by $\sqsubseteq$, i.e., every chain as an upper bound.*

Proof. Let $(T_i)_{i \in I}$ be an increasing sequence in $\mathcal{F}(t)$. We have to find an upper bound. Note that every factorisation T of t induces an equivalence relation $\approx_T$ on $\mathrm{dom}(t)$ by

$$u \approx_T v \quad :\text{iff} \quad u \text{ and } v \text{ are vertices belonging to the same factor } T(w)\,.$$

Hence, the sequence $T_0 \sqsubseteq T_1 \sqsubseteq T_2 \sqsubseteq \ldots$ induces a corresponding sequence $\approx_{T_0} \subseteq \approx_{T_1} \subseteq \approx_{T_2} \subseteq \ldots$ of equivalence relations. The limit

$$\approx := \bigcup_{i \in I} \approx_i$$

is an equivalence relation on $\mathrm{dom}(t)$ that corresponds to some factorisation T of t. We will show that $T \in \mathcal{F}(t)$. Then T is the desired upper bound for $(T_i)_{i \in I}$.

To prove the claim, note that every $\approx$-class E is the union of an increasing sequence $(E_i)_{i \in I}$ of $\approx_{T_i}$-classes. Since each T_i belongs to $\mathcal{F}(t)$, every E_i is of one of the following two types.

(I) The class is a singleton.
(II) The class corresponds to a factor of arity at most one.

If there are arbitrarily large i such that E_i is of type (I), the sequence is constant and the limit E is also of type (I). Otherwise, the limit E is a union of classes of type (II) and, hence, is also of type (II). As this holds for all classes of $\approx$, it follows that $T \in \mathcal{F}(t)$. ◀

▶ **Lemma 6.8.** *Let $\mathfrak{A}$ be a tree algebra and $C \subseteq A$ a set with $A_0 \cup A_1 \subseteq C$. Every $t \in \mathbb{T}_m C$ has a factorisation $T \in \mathcal{F}(t)$ such that*

(1) *T is reduced,*
(2) *the height of T is at most $2m$, and*
(3) *$\mathbb{T}\pi(T) \in \mathbb{T}C$.*

Proof. By Lemma 6.7, we can use Zorn's Lemma to find a maximal element $T \in \mathcal{F}(t)$. We claim that T is the desired factorisation.

(1) For a contradiction, suppose otherwise. Then there exists a factorisation U of T and a vertex $u \in \mathrm{dom}(U)$ of arity at most one such that $U(u)$ is not a singleton. Let T' be the tree obtained from T by replacing the factor $U(u)$ by its product. Then, $T \sqsubset T'$ and T is not maximal.

(2) follows from (1) by Lemma 6.6.

(3) Note that every factor $T(v)$ is either a singleton or of arity at most one. Since $A_0 \cup A_1 \subseteq C$, it follows that $\pi(T(v)) \in C$. Hence, $\mathbb{T}\pi(T) \in \mathbb{T}C$. ◀

Proof of Proposition 6.5. For the nontrivial direction, suppose that $\mathfrak{A}$ is an algebra as in the proposition and let $C \subseteq A$ be the corresponding set of generators. To prove that $\mathfrak{A}$ is regular, we fix an element $a \in A_m$. We have to show that $\pi^{-1}(a) \cap \mathbb{T}C$ is regular. Set $C' := C \cup A_0 \cup A_1$ and let $t \in \mathbb{T}C$. By Lemma 6.8, t has a factorisation $T \in \mathcal{F}(t)$ such that T is reduced, its height is at most $2m$, and $\mathbb{T}\pi(T) \in \mathbb{T}C'$. It follows that $\mathbb{T}\pi(T) \in H(a)$ where

$$H(a) := \{\, s \in \mathbb{T}C' \mid s \text{ has height at most } 2m \text{ and } \pi(s) = a \,\}\,.$$

Consequently, we have

$$\pi(t) = a \quad \text{iff} \quad \pi(\mathrm{flat}(T)) = a \quad \text{iff} \quad \pi(\mathbb{T}\pi(T)) = a \quad \text{iff} \quad \mathbb{T}\pi(T) \in H(a)\,.$$

For every finite tree s, we will construct an MSO-formula ϑ_s such that

$$t \models \vartheta_s \quad \text{iff} \quad t \text{ has a factorisation } T \in \mathcal{F}(t) \text{ such that } \mathbb{T}\pi(T) = s\,.$$

Then it follows that

$$\pi(t) = a \quad \text{iff} \quad \mathbb{T}\pi(T) \in H(a) \quad \text{iff} \quad t \models \bigvee_{s \in H(a)} \vartheta_s\,,$$

as desired. Hence, it remains to construct the formulae ϑ_s.

First, note that we can encode a factorisation T of t by a set Z that contains the root of each factor $T(v)$. Using this encoding, we can set

$$\vartheta_s := \exists Z \Big[\text{`}Z \text{ encodes a factorisation } T \text{ in } \mathcal{F}(t)\text{'}$$
$$\wedge \bigwedge_{v \in \mathrm{dom}(s)} \text{`the factor } T(v) \text{ evaluates to } s(v)\text{'} \Big]\,.$$

The first part of this formula is clearly expressible in MSO. For the second part, note that s is finite and each factor $T(v)$ is either a singleton or a term of arity at most one. In the first case it is trivial to compute the product. In the second case, we can use the formulae defining the sets $\pi^{-1}(a) \cap \mathbb{T}C$, for $a \in A_0 \cup A_1$. ◀

The price we pay for using deterministic algebras in our characterisation theorem below is that we need a slightly more general notion of recognition. A *span* $\langle \varphi, \psi \rangle : \mathfrak{A} \to \mathfrak{B}$ from a tree algebra $\mathfrak{A}$ to another tree algebra $\mathfrak{B}$ consists of two morphisms $\varphi : \mathfrak{C} \to \mathfrak{A}$ and $\psi : \mathfrak{C} \to \mathfrak{B}$ where $\mathfrak{C}$ is a third tree algebra. A subset $L \subseteq A_n$ is *recognised* by a span $\langle \varphi, \psi \rangle$ if there exists a set $P \subseteq B_n$ such that

$$L = \varphi[\psi^{-1}[P]]\,.$$

Below we will use a span $\langle p, q \rangle : \mathfrak{A} \to \mathfrak{T}(\mathcal{A})$ where the middle algebra is a subalgebra of the product $\mathfrak{A} \times \mathfrak{T}(\mathcal{A})$ and the morphisms p and q are the corresponding projections.

▶ **Definition 6.9.** Let $\mathfrak{A}$ be a tree algebra and $\mathcal{A}$ an automaton. We denote by $\widehat{\mathfrak{A}}$ the subalgebra of the product $\mathfrak{A} \times \mathfrak{T}(\mathcal{A})$ with domains

$$\widehat{A}_n := \{\, \langle \pi(t), \delta \rangle \mid t \in \mathbb{T}_n A \text{ and } \delta \text{ the profile of some partial run of } \mathcal{A} \text{ on } t \,\}\,.$$

Let $p : \widehat{\mathfrak{A}} \to \mathfrak{A}$ and $q : \widehat{\mathfrak{A}} \to \mathfrak{T}(\mathcal{A})$ be the corresponding projections.

(Note that $\widehat{\mathfrak{A}}$ is well defined as its domains are closed under products.) We start with a technical result showing that the projection $\widehat{\mathfrak{A}} \to \mathfrak{A}$ is surjective. At least this is the case if the algebra $\mathfrak{A}$ is regular and $\mathcal{A}$ the corresponding automaton, i.e., a tree automaton such that, for every element $a \in A_0 \cup A_1$, we can choose a starting state for $\mathcal{A}$ from which it recognises the set $\pi^{-1}(a) \cap \mathbb{T}C$.

▶ **Lemma 6.10.** *Let $\mathfrak{A}$ be a regular tree algebra and $\mathcal{A}$ an automaton for $\mathfrak{A}$. The projection $p : \widehat{\mathfrak{A}} \to \mathfrak{A}$ is surjective and every fibre $p^{-1}(a)$ is finite.*
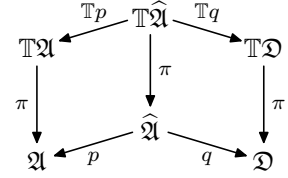
Proof. Consider an element $a \in A_m$. Let ϱ be the run of $\mathcal{A}$ on the tree $\mathrm{sing}(a)$ and δ the (profile corresponding to the) transition at the root of ϱ. Then $\langle a, \delta \rangle \in \widehat{A}$ and $p(\langle a, \delta \rangle) = a$. Hence, $a \in \mathrm{rng}\, p$. For the second statement, note that every domain $\widehat{A}_m \subseteq A_m \times T_m(\mathcal{A})$ is finite. Hence, so is $p^{-1}(a) \subseteq \widehat{A}_m$, for $a \in A_m$. ◀

Combining the notions and results of this section, we obtain the following characterisation of when a tree algebra is regular.

▶ **Theorem 6.11.** *Let $\mathfrak{A}$ be a finitary tree algebra and $C \subseteq A$ a finite set of generators. $\mathfrak{A}$ is regular if, and only if, there exists a deterministic algebra $\mathfrak{D}$ and a subalgebra $\widehat{\mathfrak{A}} \subseteq \mathfrak{A} \times \mathfrak{D}$ such that*

- *the first projection $p : \widehat{\mathfrak{A}} \to \mathfrak{A}$ is surjective,*
- *every fibre $p^{-1}(a)$ is finite, and*
- *the span $\langle \mathbb{T}p, \pi \circ \mathbb{T}q \rangle : \mathbb{T}\mathfrak{A} \to \mathfrak{D}$ recognises every preimage*

$$\pi^{-1}(a) \cap \mathbb{T}, \quad \text{for } a \in A_0 \cup A_1.$$

Proof. ($\Rightarrow$) Fix an automaton $\mathcal{A}$ for $\mathfrak{A}$, set $\mathfrak{D} := \mathfrak{T}(\mathcal{A})$, and let $\widehat{\mathfrak{A}}$ be the algebra from Definition 6.9. We have seen above that $\mathfrak{D}$ is deterministic, the projection $p : \widehat{\mathfrak{A}} \to \mathfrak{A}$ is surjective, and all fibres $p^{-1}(a)$ are finite. To conclude the proof, consider an element $a \in A_0 \cup A_1$. Let q_a be the starting state that $\mathcal{A}$ uses to recognise the preimgae $\pi^{-1}(a) \cap \mathbb{T}C$ and let $P \subseteq D$ be the set of all profiles $\bigwedge_{i<m} \langle q, k_i, p_i \rangle$ such that $q = q_a$ and, from the state p_i, $\mathcal{A}$ accepts the singleton tree with label x_i, for $i < m$. For $t \in \mathbb{T}C$, it follows that

$$
\begin{aligned}
\pi(t) = a \quad &\text{iff} \quad \text{there exists an accepting run on } t \text{ starting in the state } q_a \\
&\text{iff} \quad \text{there exists } s \in (\mathbb{T}p)^{-1}(t) \text{ such that } \mathbb{T}q(s) \text{ is such a run} \\
&\text{iff} \quad \text{there exists } s \in (\mathbb{T}p)^{-1}(t) \text{ such that } \pi(\mathbb{T}q(s)) \in P.
\end{aligned}
$$

($\Leftarrow$) By Proposition 6.5, it is sufficient to show that the preimages $\pi^{-1}(a) \cap \mathbb{T}C$ are regular for elements a of arity at most 1. Hence, let $a \in A_0 \cup A_1$ and set $C' := q[p^{-1}[C]]$. Note that C' is a finite set since, by assumption, all fibres of p are finite. Furthermore, we know that there exists a (finite) set $P \subseteq D$ such that

$$
\begin{aligned}
\pi^{-1}(a) \cap \mathbb{T}C &= (\mathbb{T}p)[(\pi \circ \mathbb{T}q)^{-1}[P]] \\
&= \bigcup_{d \in P} (\mathbb{T}p)[(\mathbb{T}q)^{-1}[\pi^{-1}(d)]] \\
&= \bigcup_{d \in P} (\mathbb{T}p)[(\mathbb{T}q)^{-1}[\pi^{-1}(d) \cap C']],
\end{aligned}
$$

where the last equality holds as every tree in $\mathbb{T}C$ is mapped by $\mathbb{T}q \circ (\mathbb{T}p)^{-1}$ to a tree in $\mathbb{T}C'$. As finitary subalgebras of deterministic algebras are regular, each preimage $\pi^{-1}(d) \cap C'$ forms a regular language. Furthermore, regular tree languages are closed under projections and inverse projections. Hence, each term in the above union is regular and, therefore, so is the union itself. ◀

7 Conclusion

In this article we have developed a framework for recognisability of tree languages. We have isolated a class of algebras that recognise exactly the regular tree languages and we have shown that this class meets our main requirements: it forms a pseudo-variety and it has syntactic algebras. Furthermore, we have proved that it is the only class with these properties. Finally, we have included a simple example of how to use our framework to obtain characterisation results. More substantial applications are deferred to a forthcoming article [3].

The basic concept our framework is built around is the notion of a regular tree algebra. We have given two different definitions of these algebras: the first one is simple and easy to use, but it requires automata theory; the second one is more complicated and abstract, but it has the advantage that it is purely algebraic and does not require automata. It is currently open whether one can also define regularity of a tree algebra in terms of a set of equations. The work of Milius and Urbat [11] suggests that this might be possible, but no explicit description of the equations involved is known at this point. It is also unclear how large and complicated such a set of equations would be. Our current conjecture is that, for every finite set X of variables and every tree $t \in \mathbb{T}X$, we need to have an equation of the form $t = t'$, where $t' \in \mathbb{T}X$ is some *regular* tree depending on t. But it is not obvious why this should be equivalent to the tree algebra being regular. In fact, both directions of this equivalence seem to require non-trivial arguments.

A Closure properties of regular tree languages

The closure properties of the class of all regular languages of infinite trees is well-understood. In particular, the class is closed under boolean operations and projections (see, e.g., [16, 17, 12]). Another well-known closure property is that under inverse morphisms. As I have not been able to find a published proof of this fact, I include one here.

▶ **Lemma A.1.** *Let* $\varphi : \mathbb{T}\Sigma \to \mathbb{T}\Gamma$ *be a morphism of tree algebras. If* $L \subseteq \mathbb{T}_m\Gamma$ *is regular, so is* $\varphi^{-1}[L] \subseteq \mathbb{T}_m\Sigma$.

Proof. Set $\widehat{\Sigma} := \Sigma \cup \{x_0, \dots, x_{m-1}\}$ and $\widehat{\Gamma} := \Gamma \cup \{x_0, \dots, x_{m-1}\}$. Fix an automaton $\mathcal{A} = \langle Q, \widehat{\Gamma}, \Delta, q_0, \Omega \rangle$ recognising L. For $c \in \Sigma$, let $\Pi(c)$ be the set of all profiles of partial runs of $\mathcal{A}$ on $\varphi(\mathrm{sing}(c))$ and, for $i < m$, let $\Pi(x_i)$ be the set of all states from which $\mathcal{A}$ accepts the singleton tree with label x_i. We construct an automaton $\mathcal{B}$ for $\varphi^{-1}[L]$ as follows. The set of states is $Q \times D \cup \{\langle \bot, \bot \rangle\}$, where D is the set of priorities used by $\mathcal{A}$. The initial state is $\langle q_0, k \rangle$, for an arbitrary $k \in D$, and the priority function is given by $\Omega(\langle p, k \rangle) := k$ and $\Omega(\langle \bot, \bot \rangle) = 0$. The transitions of $\mathcal{B}$ are as follows.

$$\langle \langle p, l \rangle, \ c, \ \langle q_0, k_0 \rangle, \dots, \langle q_{n-1}, k_{n-1} \rangle \rangle \qquad \text{for } c \in \widehat{\Sigma}, \ \langle p, k_0, q_0, \dots, k_{n-1}, q_{n-1} \rangle \in \Pi(c),$$

$$\langle \langle \bot, \bot \rangle, \ c, \ \langle \bot, \bot \rangle, \dots, \langle \bot, \bot \rangle \rangle \qquad \text{for } c \in \widehat{\Sigma}.$$

It is straightforward to check that $\mathcal{B}$ accepts a tree $t \in \mathbb{T}_m\Sigma$ if, and only if, $\mathcal{A}$ accepts $\varphi(t)$. ◀

----- **References** -----

1   J. ADÁMEK AND J. ROSICKÝ, *Locally Presentable and Accessible Categories*, Cambridge University Press, 1994.

2   A. BLUMENSATH, *Branch-Continuous Tree Algebras.* arXiv:1807.04568.

3   ——, *Wreath Products of ω-Forest Algebras.* in preparation.

4   ——, *Recognisability for algebras of infinite trees*, Theoretical Computer Science, 412 (2011), pp. 3463–3486.

5   ——, *An Algebraic Proof of Rabin's Tree Theorem*, Theoretical Computer Science, 478 (2013), pp. 1–21.

6   M. Bojańczyk, *Recognisable languages over monads.* unpublished note, arXiv:1502.04898v1.

7   M. Bojańczyk and T. Idziaszek, *Algebra for Infinite Forests with an Application to the Temporal Logic EF*, in Proc. 20th International Conference on Concurrency Theory, CONCUR, LNCS 5710, 2009, pp. 131–145.

8   M. Bojańczyk, T. Idziaszek, and M. Skrzypczak, *Regular languages of thin trees*, in Proc. 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, 2013, pp. 562–573.

9   M. Bojańczyk and B. Klin, *A non-regular language of infinite trees that is recognizable by a sort-wise finite algebra.* arXiv:1804.06667.

10 F. Borceux, *Handbook of Categorical Algebra*, vol. 2, Cambridge University Press, 1994.

11 S. Milius and H. Urbat, *Equational Axiomatization of Algebras with Structure*, in Proc. 22nd International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2019, 2019, pp. 400–417.

12 F. Niesser, *Non-deterministic Tree Automata*, in Automata, Logic, and Infinite Games, E. Grädel, W. Thomas, and T. Wilke, eds., LNCS 2500, Springer Verlag, 2002, pp. 149–156.

13 D. Perrin and J.-E. Pin, *Infinite Words – Automata, Semigroups, Logic and Games*, Elsevier, 2004.

14 J. Reiterman, *The Birkhoff theorem for finite algebras*, Algebra Universalis, 14 (1982), pp. 1–10.

15 M. P. Schützenberger, *On Finite Monoids Having Only Trivial Subgroups*, Information and Control, 8 (1965), pp. 190–194.

16 W. Thomas, *Automata on Infinite Objects*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B, Elsevier, Amsterdam, 1990, pp. 135–191.

17 ——, *Languages, Automata, and Logic*, in Handbook of Formal Languages, G. Rozenberg and A. Salomaa, eds., vol. 3, Springer, New York, 1997, pp. 389–455.

18 Y. I. Yanov and A. A. Muchnik, *Existence of k-valued closed classes without a finite basis*, Dokl. Akad. Nauk., 127 (1959), pp. 44–46.