Masaryk University
Faculty of Informatics

Algebras for unranked trees

Master's Thesis

Jakub Lédl

Brno, Fall 2018

This is where a copy of the official signed thesis assignment and a copy of the Statement of an Author is located in the printed version of the document.

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Jakub Lédl

Advisor: Dr. rer. nat. Achim Blumensath

Acknowledgement

I would like to express my sincerest gratitude to my advisor, Dr. rer. nat. Achim Blumensath, for his patient guidance and invaluable advice with which he counseled me while I was writing this manuscript. I would also like to heartfeltly thank my dear family and friends for all their help and support they so considerately provided me with.

Abstract

We describe an algebraic framework for languages of infinite forests using Eilenberg-Moore algebras of a monad. We describe the class of *regular algebras*, which as recognisers correspond to regular forest languages, prove the existence of regular syntactic algebras for regular languages and show that this class forms a pseudovariety, suggesting existence of an equational characterisation. We also prove a special case of a characterisation of regular languages definable in the logic EF, and give a counterexample to the full characterisation.

Keywords

infinite forests, monads, eilenberg-moore algebras, logic, definability

Contents

1 Introduction

Historically, algebraic methods have had valuable applications in the theory of formal languages. First defined in the setting of languages of finite words, one of the central notions in this field is that of the *syntactic monoid* of a language L, originally given by Schützenberger. In the original setting, the algebraic notion of recognisability was proven to correspond with recognisability by automata in the sense that a language of finite words is regular (recognised by a finite automaton) if and only if its syntactic monoid is finite. Subsequently, applying algebraic methods, such as study of pseudovarieties and their equational characterisations, yielded useful connections between classes of regular languages and equational classes of finite monoids, a well-known example being Schützenberger's decidable characterisation of first-order definable languages as those with aperiodic syntactic monoids.

The success of this algebraic approach to language theory naturally led to considerations of objects other than finite words. For more complicated structures, however, especially for those which are in some way infinite, even the correct notion of what the corresponding algebraic structure should be becomes less clear. Specifically, in case of infinite trees, previous work include that of Blumensath (2011) for infinite trees proper, Idziaszek et al. (2016) for thin trees, Bojańczyk and Place (2012) using a topological approach and Bojańczyk and Idziaszek (2009) for infinite forests, where, however, only so-called *regular* forests, i.e. forests with only finitely many nonisomorphic subtrees, were considered.

A possible solution to the problem of organizing together all the operations and identities of the desired algebraic structures has been found in category theory, specifically in the notion of an *Eilenberg-Moore algebra*. These algebras form a general abstract framework for description of "well-behaved" algebraic structures and posses a well-developed basic theory. This is the approach proposed by Bojańczyk (2015), used by Blumensath (2018b) for infinite trees and the one we will adapt for infinite forests.

The goal of this thesis therefore is to develop an algebraic framework for recognisability of languages of infinite forests using this approach. This entails specifying the correct notion of an infinite forest, so that the monad operations can be defined. Once the monad and the corresponding algebraic structures are obtained, the next natural step is to give an algebraic characterisation of regular forest languages. These are the two primary goals accomplished in this work.

This manuscript is organised as follows: in Chapter 2, we define the basic concepts such as infinite forests themselves and the category-theoretic setting in which the rest of the work will be discussed. We describe the monadic structure of the infinite forest functor and define the algebraic structures, the ω-forest algebras.

In Chapter 3, we describe the fundamentals of recognisability of regular forest languages; we introduce the class of ω-forest algebras, show that they recognise exactly the regular languages, paralleling Blumensath (2018b), and show that for regular languages, it is possible to construct its syntactic recognising algebra.

Finally, in Chapter 4, we characterise definability of regular forest languages using the logic EF, a fragment of CTL, and its characterisation in terms of the syntactic algebra. We adapt the work of Bojańczyk and Idziaszek (2009) to prove a special case of this characterisation and give a counterexample showing that in general, the characterisation does not hold in our setting. We then discuss the implications for our current algebraic framework.

In Appendix A, we collect some technical results which would unnecessarily distract from the main text.

2 ω-forests and algebras

This chapter introduces the basic notions and structures used in the following ones. The algebraic framework for languages of forests with infinite branches will be developed using the theory of Eilenberg–Moore algebras for an appropriate monad as in (Blumensath, 2018b); the category-theoretical notions used can be found in (Borceux, 1994, Chapter 4). Unlike in (Blumensath, 2018b), we wish to describe forests that are unranked, i.e. do not distinguish the arity of labels. On the other hand, the approach we wish to adapt distinguishes between usual trees, forests etc. and so-called *contexts*, which are forests into which other forests can be substituted. This suggest that the correct setting is that of two-sorted sets. The following definition captures this.

Definition 2.1. A *biset* is simply a pair of sets (A, B). Given a biset S, we will denote its first component by S_0 and the second one by S_1. Given bisets S and T, a *biset map* $f \colon S \to T$ is a pair of maps (f_0, f_1), where f_0 is a map $S_0 \to T_0$ and f_1 a map $S_1 \to T_1$. Given such a map, we will likewise write f_0 for the first component and f_1 for the second one. Finally, a *relation* R between bisets S and T is a pair of relations between their corresponding components. Similarly, the relations comprising R will be denoted R_0 and R_1. Given a biset S, we will write $S_\cup$ for the set $S_0 \cup S_1$.

The category of bisets will be denoted by $\mathbf{BiSet} := \mathbf{Set} \times \mathbf{Set}$.

We follow the approach of Blumensath (2018a) for trees in defining forests.

Definition 2.2. Let $\mathbb{N}^*$ denote the free monoid on the set of natural numbers (including zero) and $\mathbb{N}^+ := \mathbb{N}^* - \{\varepsilon\}$ the free semigroup on the same set. We will usually write w for a general element of either $\mathbb{N}^*$ or $\mathbb{N}^+$ and $\langle n_1 \dots n_k \rangle$ for the word composed of letters n_1 to n_k. Concatenation of words will be usually denoted either by simple juxtaposition or as $w ^\frown v$ where wv would not be sufficiently clear.

A *forest domain* is a subset D of $\mathbb{N}^+$ which satisfies the following properties:

1. D is nonempty – a forest may not be empty.

2. For every $w \in D \cup \{\varepsilon\}$, only finitely many words of the form $w\langle n\rangle, n \in \mathbb{N}$ are in D. For $w \in D$, this condition ensures finite branching, while for $w = \varepsilon$, it guarantees that forests only have finitely many trees.

3. For every $v, w \in \mathbb{N}^+$, if $vw \in D$, then also $v \in D$.

4. For every $w \in D \cup \{\varepsilon\}$ and every n such that $w\langle n\rangle \in D$, it is the case that $w\langle k\rangle \in D$ for every $k < n$.

Given a forest domain D, we define the subsets of *leaves*

$$\mathrm{Leaf}(D) := \{w \in D \,|\, wv \notin D \text{ for any } v \neq \varepsilon\},$$

the subset of *internal vertices* $\mathrm{Int}(D) := D - \mathrm{Leaf}(D)$ and that of *roots* $\mathrm{Root}(D) := \{w \in D \,|\, |w| = 1\}$.

Definition 2.3. Given a biset S, a *forest* over S is a triple (D, ℓ_0, ℓ_1), where D is a forest domain and ℓ_0, ℓ_1 are maps of types $\mathrm{Leaf}(D) \to S_0$ and $\mathrm{Int}(D) \to S_1$ respectively, called the *labellings*.

We will write $\mathrm{Dom}(t)$ for the domain of a forest t and we will define, for $w \in \mathrm{Dom}(t)$, $t(w)$ to be either $\ell_0(w)$ or $\ell_1(w)$ as appropriate.

Finally, a *tree* over S is a forest t such that $|\mathrm{Root}(t)| = 1$, in which case we will refer to the single $w \in \mathrm{Root}(t)$ simply as *the* root of t.

Given a vertex $w \in \mathrm{Dom}(t)$, we will write $t|w$ for the subtree of t rooted at w; formally, the domain of $t|w$ is obtained by taking those vertices of t of which w is a prefix and replacing this prefix by $\langle 0\rangle$, while the labels are transferred in the obvious manner.

The set of all forests over S will be denoted $\mathbb{F}_0 S$ and the subset of all trees by $\mathbb{T}_0 S$.

Figure 2.1 shows examples of trees – non-nullary labels will be usually drawn grey, nullary ones white. A general plain tree will be denoted simply as a triangle, as the tree t in the figure.

Definition 2.4. Given an alphabet S, let $S^\square$ be a biset obtained from S by adding a new nullary label [1], which we will denote $\square$ and call the

1. Formally, $S_1^\square := S_1$ and $S_0^\square$ is the disjoint union $S_0 + \mathbf{1}$, where $\mathbf{1}$ is an arbitrary but fixed singleton. We will freely identify the elements of S_0 with the corresponding ones of $S_0^\square$.

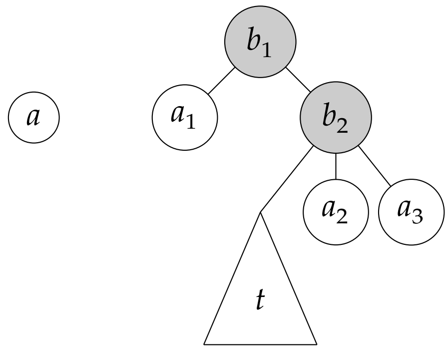


Figure 2.1: Examples of trees

hole. A *context* s over S, also called a *forest with a hole*, is a forest over $S^\square$ such that there is exactly one leaf v labelled by $\square$ which furthermore is not a root of s, i.e. the path to v has nonzero length.

Given a context s, we will write $\mathrm{Hole}(s) \in \mathrm{Dom}(s)$ for the hole vertex of $\mathrm{Dom}(s)$.

A *tree context* is simply a context which is also a tree.

We will denote by $\mathbb{F}_1 S$ and $\mathbb{T}_1 S$ the sets of all contexts and tree contexts over S.

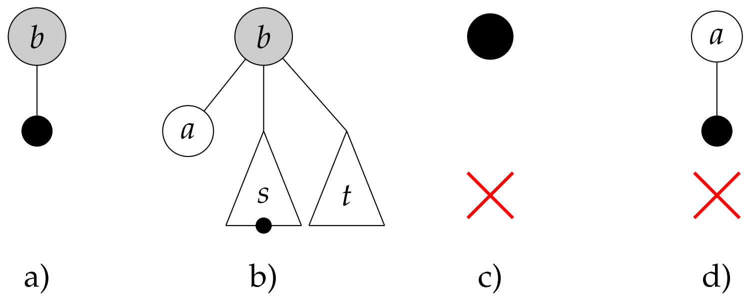


Figure 2.2: Tree contexts – examples and nonexamples

The first two pictures depict simple tree contexts (the hole is drawn black). A pictorial representation of a typical tree context as a triangle "with the hole at the bottom" is also shown, marked s. Regarding the nonexamples, the first one illustrates the condition that the path from root to the hole has length at least one; this is required to make the general composition operation, which we will introduce later, well-defined. The second one illustrates that since the hole can be understood as standing for an arbitrary forest, the label of its parent must be always nonnullary, as forests may be (in fact, are required to be) nonempty.

Having defined forests and contexts, we can describe the functorial action on morphisms.

Definition 2.5. For a biset map $f\colon S \to T$, the map $\mathbb{F}_0(f)\colon \mathbb{F}_0 S \to \mathbb{F}_0 T$ acts on forests by the obvious relabelling, i.e. we have

$$\mathbb{F}_0(f)(D, \ell_0, \ell_1) := (D, f_0 \circ \ell_0, f_1 \circ \ell_1)\,.$$

Similarly, we define $\mathbb{F}_1(f)\colon \mathbb{F}_1 S \to \mathbb{F}_1 T$ by setting

$$\mathbb{F}_1(f)(s) := \mathbb{F}_0(f^{\Box})(s)\,,$$

where $f^{\Box}\colon S^{\Box} \to T^{\Box}$ acts as f does on ordinary labels and as identity on $\Box$.

We also define $\mathbb{T}_0(f)\colon \mathbb{T}_1 S \to \mathbb{T}_1 T$ and $\mathbb{T}_1(f)\colon \mathbb{T}_1 S \to \mathbb{T}_1 T$ as the appropriate restrictions of the previous maps.

We also extend relations in a similar way.

Definition 2.6. Given a relation R between bisets S and T, we define the relation $R_0^{\mathbb{F}}$ from $\mathbb{F}_0 S$ to $\mathbb{F}_0 T$ by setting $R_0^{\mathbb{F}}(t_1, t_2)$ if and only if $\mathrm{Dom}(t_1) = \mathrm{Dom}(t_2)$, $R_0(t_1(v), t_2(v))$ for every leaf $v \in \mathrm{Leaf}(t_1)$ and $R_1(t_1(w), t_2(w))$ for every $w \in \mathrm{Int}(t_1)$, i.e. the forests are related if they have the same shape and all the correspondings labels are related. The relation $R_1^{\mathbb{F}}$ from $\mathbb{F}_1 S$ to $\mathbb{F}_1 T$ is defined similarly, requiring both the domains and the holes to be the same.

The following observation is immediate. We omit the proof.

Observation 2.7. *With these actions on maps, the assignments $\mathbb{T}_0$, $\mathbb{T}_1$, $\mathbb{F}_0$ and $\mathbb{F}_1$ all form functors of type* **BiSet** $\to$ **Set**.

Definition 2.8. The endofunctor $\mathbb{F}\colon$ **BiSet** $\to$ **BiSet** is simply the pairing $(\mathbb{F}_0, \mathbb{F}_1)$. Given a relation R from S to T, the relation $R^{\mathbb{F}}$ between $\mathbb{F}S$ and $\mathbb{F}T$ is defined as the pair $(R_0^{\mathbb{F}}, R_1^{\mathbb{F}})$.

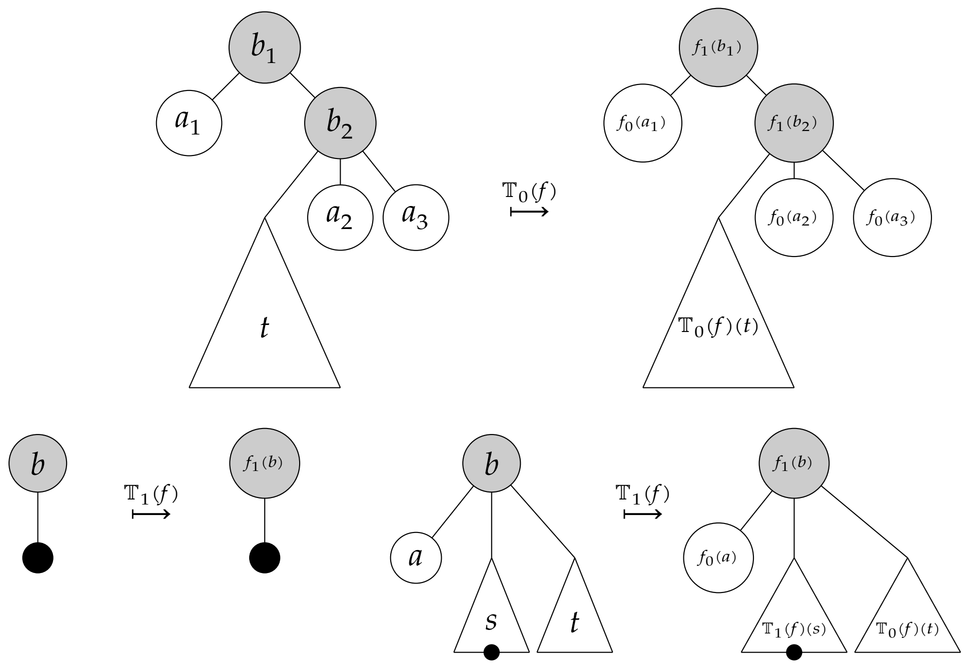


Figure 2.3: Action of the maps $\mathbb{T}_i(f)$

Before defining the natural transformations required to make $\mathbb{F}$ into a monad on **BiSet**, we will describe some useful elementary operations on forests. However, since fully formal definitions of the following concepts would be unnecessarily involved, we only describe them in sufficient detail, leaving the full formal treatment to the appendix.

Definition 2.9. Given forests t_1 and t_2 over S, we will write $t_1 \oplus t_2$ for their *horizontal composition*, obtained by putting all trees of t_2 to the right of all trees of t_1. Note that this definition makes sense in two cases: either both t_1 and t_2 are plain forests, in which case $t_1 \oplus t_2$ is plain as well, or exactly one of t_1 and t_2 is a context, whence $t_1 \oplus t_2$ is also a context. Therefore, we actually obtain three operations with types $\mathbb{F}_0 S \times \mathbb{F}_0 S \to \mathbb{F}_0 S$, $\mathbb{F}_1 S \times \mathbb{F}_0 S \to \mathbb{F}_1 S$ and $\mathbb{F}_0 S \times \mathbb{F}_1 S \to \mathbb{F}_1 S$. When there is no danger of confusion, however, we will denote all three by $\oplus$. Note that this "operation" is associative; for all appropriate t_1, t_2, t_3, we have $(t_1 \oplus t_2) \oplus t_3 = t_1 \oplus (t_2 \oplus t_3)$ for the unique consistent choice of the actual operations.

Definition 2.10. Let s be a context and t either a forest or a context. By their *vertical composition* $s \cdot t$ we mean the forest which results from substituting t for the hole in s, i.e. removing the hole and attaching every tree in t to the vertex the hole was attached to.

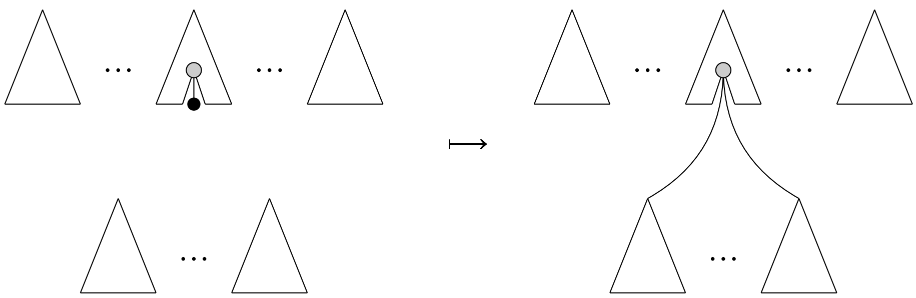


Figure 2.4: Vertical composition of forests

Since $s \cdot t$ has a hole if and only if t does, we again get two operations of types $\mathbb{F}_1 S \times \mathbb{F}_0 S \to \mathbb{F}_0 S$ and $\mathbb{F}_1 S \times \mathbb{F}_1 S \to \mathbb{F}_1 S$. As before, when no confusion can arise, we will denote both operations by $s \cdot t$.

Also, given $b \in S_1$ and a forest or context t, we will write bt for the tree with the root labelled by b and the forest of successors of the root given by t.

Finally, given a context $s \in \mathbb{F}S_1$, we will write $s^\omega \in \mathbb{F}_0 S$ for the forest obtained as the "infinite product" $s \cdot s \cdot \ldots$ satisfying $s^\omega = s \cdot s^\omega$.

2.1 The natural transformations sing and flat

As discussed in the beginning of this chapter, we want to equip the functor $\mathbb{F}$ with the structure of a monad on the category **BiSet**. This amounts to defining two natural transformations $\mathrm{sing}\colon \mathrm{Id}_{\mathbf{BiSet}} \to \mathbb{F}$ and $\mathrm{flat}\colon \mathbb{F}^2 \to \mathbb{F}$ satisfying certain conditions. For Eilenberg-Moore algebras for a general monad T, the objects TX correspond to the (carriers of) free algebras; the transformation sing therefore can be understood as the inclusion of generators into the term algebra, while flat represents a general kind of term-flattening operation that corresponds e.g. in the case of free semigroups to word concatenation.

Definition 2.11. Given a biset S, the map $\mathrm{sing}_{S,0}\colon S_0 \to \mathbb{F}_0 S$ takes a label $a \in S_0$ to the tree with a single vertex labelled by a. When no confusion can arise, the tree $\mathrm{sing}_{S,0}(a)$ may be simply written as a.

Similarly, the map $\mathrm{sing}_{S,1}\colon S_1 \to \mathbb{F}_1 S$ takes $b \in S_1$ to the tree $b\square$.

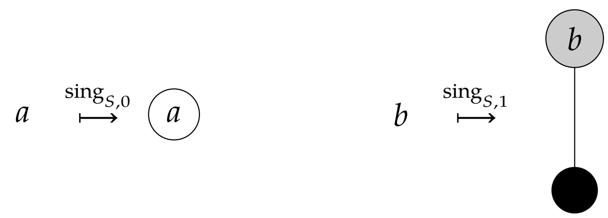


Figure 2.5: The maps $\mathrm{sing}_{S,i}$

Given these definitions, the squares

$$\begin{array}{ccc} S_0 & \xrightarrow{f_0} & T_0 \\ \downarrow{\scriptstyle \mathrm{sing}_{S,0}} & & \downarrow{\scriptstyle \mathrm{sing}_{T,0}} \\ \mathbb{F}_0 S & \xrightarrow{\mathbb{F}_0(f)} & \mathbb{F}_0 T \end{array} \qquad \begin{array}{ccc} S_1 & \xrightarrow{f_1} & T_1 \\ \downarrow{\scriptstyle \mathrm{sing}_{S,1}} & & \downarrow{\scriptstyle \mathrm{sing}_{T,1}} \\ \mathbb{F}_1 S & \xrightarrow{\mathbb{F}_1(f)} & \mathbb{F}_1 T \end{array}$$

commute for all S, T and $f\colon S \to T$, so the family $(\mathrm{sing}_{S,0}, \mathrm{sing}_{S,1})_S$ forms a natural transformation $\mathrm{sing}\colon \mathrm{Id}_{\mathbf{BiSet}} \to \mathbb{F}$.

The following definition is rather informal. A more precise version can be found in the appendix.

Definition 2.12. For a biset S, the *flattening map* $\mathrm{flat}_{S,0}\colon \mathbb{F}_0\mathbb{F}S \to \mathbb{F}_0 S$ acts as follows: to flatten a forest labelled by forests, it flattens the individual trees and composes the resulting forests horizontally. To flatten an individual tree, there are two cases.

1. The tree consists of a single vertex, in which case it is labelled by a forest $t \in \mathbb{F}_0 S$; this forest is then the result of the flattening.

2. The tree has a forest of successors. In this case, its root is labelled by a context $s \in \mathbb{F}_1 S$. The forest of successors is flattened "recursively", obtaining a forest $t \in \mathbb{F}_0 S$, and $s \cdot t$ is returned as the result.

In symbols, the operation is specified by $\mathrm{flat}_{S,0}(\mathrm{sing}_{\mathbb{F}S,0}(t)) := t$ and $\mathrm{flat}_{S,0}(t\tau) := t \cdot \mathrm{flat}_{S,0}(\tau)$.

Note that this "definition" is not properly recursive, since there may be infinite branches. However, as will be argued shortly, it can be realized as a *corecursive* definition and it does actually define a proper operation thanks to the restrictions placed on forests.

The map $\mathrm{flat}_{S,1}\colon \mathbb{F}_1\mathbb{F}S \to \mathbb{F}_1 S$ acts similarly to $\mathrm{flat}_{S,0}$, the only additional case being the case of a tree τ whose root is directly succeeded by a hole. In that case, the hole of τ becomes the hole of the forest that labels τ, as in Figure 2.7.

Some of the reasons for the restrictions placed on forests can now be explained. First, the equation in Definition 2.12 suggests that the flattened forest is built in a corecursive manner. The usual requirement for (positively presented) corecursive definitions to be valid is that they are productive (Atkey & McBride, 2013), i.e. each iteration actually proceeds in constructing the resulting structure further; this is exactly what the requirement that paths from roots to holes have nonzero length ensures. Furthermore, without this requirement, the flattening could produce forests with infinite branching, as Figure 2.8 demonstrates.

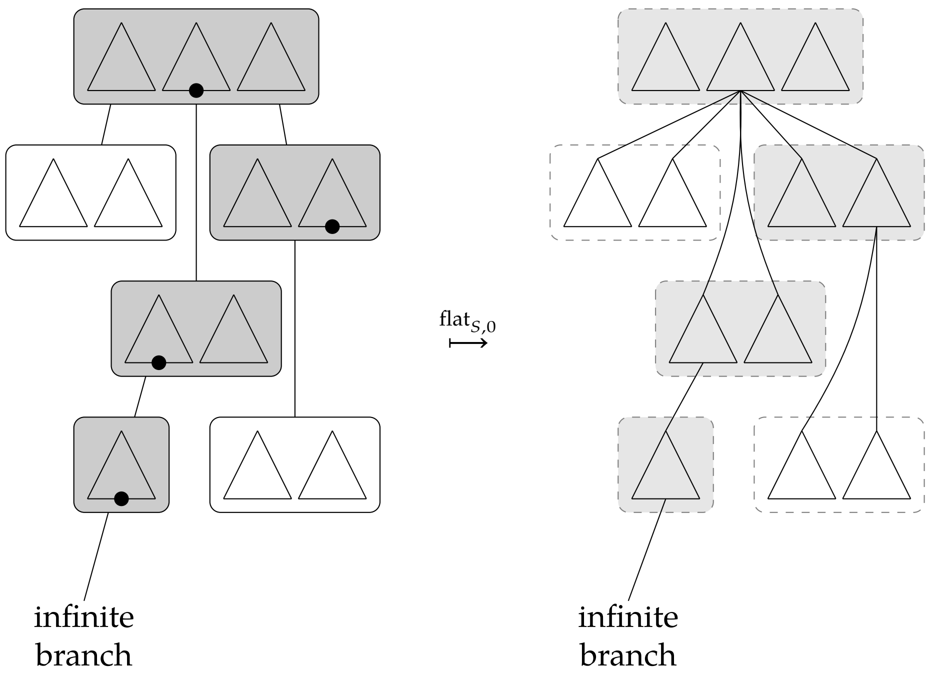


Figure 2.6: The map $\text{flat}_{S,0}$

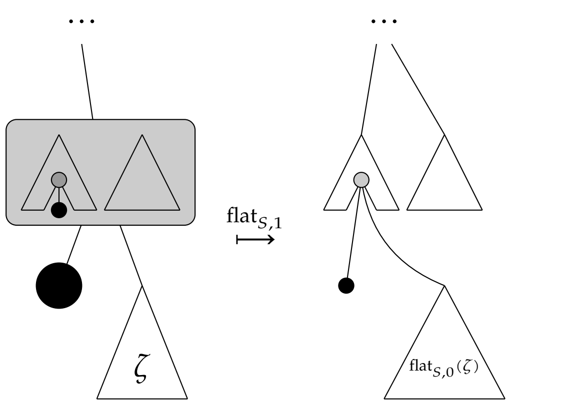


Figure 2.7: The map $\text{flat}_{S,1}$ – the special case

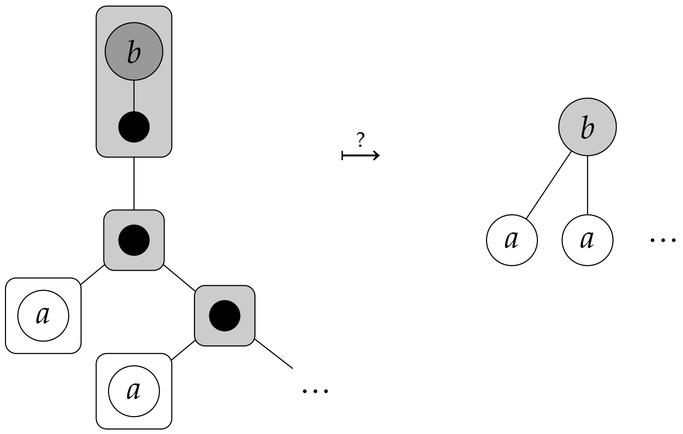
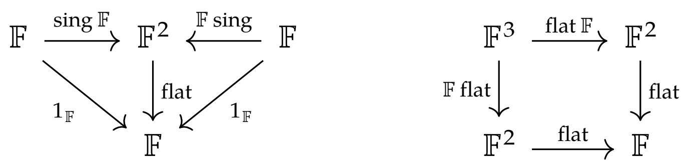


Figure 2.8: Infinite branching when holes in roots are allowed

The requirement for forests to be nonempty can be easily justified by noting that if $\mathbf{0}$ were an empty forest and we had $b \in S_1$, there would be no way to define $b(\mathbf{0}) = \mathrm{sing}_{S,1}(b) \cdot \mathbf{0}$, as the resulting forest would have a leaf labeled by an element of S_1. Subsequently, $\mathrm{flat}_{S,0}$ could not be defined.

Note that this section only aims to describe the flattening map in sufficient detail; the formal definition is given in the appendix. Similarly, the fact that flattening is a natural transformation and that together, the natural transformations satisfy the monad identities are proved there.

Proposition 2.13. *The family of biset morphisms* $(\mathrm{flat}_{S,0}, \mathrm{flat}_{S,1})_S$ *forms a natural transformation* $\mathrm{flat}\colon \mathbb{F}^2 \to \mathbb{F}$. *Furthermore, the following diagrams commute in the category of endofunctors of* **BiSet**.



The triple $(\mathbb{F}, \mathrm{flat}, \mathrm{sing})$ *is thus a monad on the category* **BiSet**.

That flat is a natural transformation can be intuitively justified by noting how the compositions $\mathrm{flat}_{T,0} \circ \mathbb{F}_0(\mathbb{F}(f))$ and $\mathbb{F}_0(f) \circ \mathrm{flat}_{S,0}$ (or $\mathrm{flat}_{T,1} \circ \mathbb{F}_1(\mathbb{F}(f))$ and $\mathbb{F}_1(f) \circ \mathrm{flat}_{S,1}$) act on the trees in Figures 2.6 and

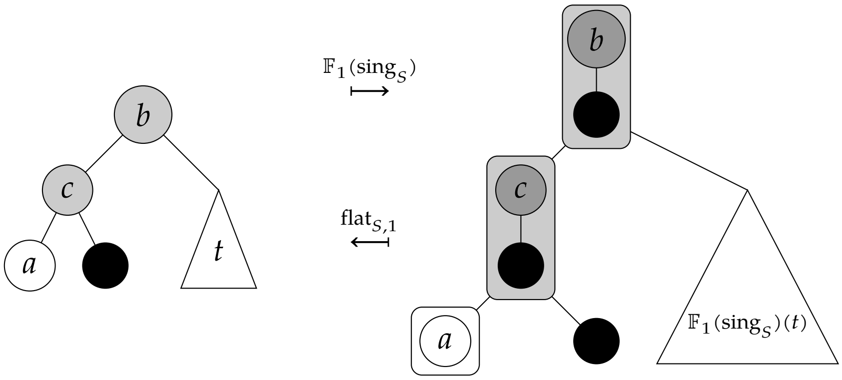


Figure 2.9: The right triangle identity for $\mathbb{F}$

2.7. The triangle identity flat $\circ$ sing $\mathbb{F} = 1_{\mathbb{F}}$ can be seen from the definitions of $\mathrm{sing}_{S,0}$ and $\mathrm{sing}_{S,1}$ and the identity flat $\circ \mathbb{F}$ sing $= 1_{\mathbb{F}}$ is illustrated by Figure 2.9.

The final identity, expressing associativity of the flattening maps, is proved in the appendix.

2.2 ω-forest algebras

Having described our notion of forests and the necessary categorical machinery, we are in a position to describe the algebraic structures we will use as recognisers for languages. Many definitions presented here are simply instances of the corresponding general notions from the theory of Eilenberg–Moore algebras.

Definition 2.14. An *ω-forest algebra* is a pair (A, α), where A is a biset and $\alpha\colon \mathbb{F}A \to A$ is a biset morphism making the following diagrams commute.

$$\begin{array}{ccc} A & \xrightarrow{\mathrm{sing}_A} & \mathbb{F}A \\ & \searrow^{\mathrm{id}_A} & \downarrow \alpha \\ & & A \end{array} \qquad\qquad \begin{array}{ccc} \mathbb{F}^2 A & \xrightarrow{\mathrm{flat}_A} & \mathbb{F}A \\ \mathbb{F}(\alpha) \downarrow & & \downarrow \alpha \\ \mathbb{F}A & \xrightarrow{\alpha} & A \end{array}$$

From now on, ω-forest algebras will be usually called just *algebras* for simplicity.

In this definition, the biset $\mathbb{F}A$ should be understood as containing terms built from elements of the carrier set A and α as an evaluation map. Note that the identites required in the definition then express natural requirements on such a map; the first one requires that the "atomic terms", in both sorts, evaluate back to whatever values they contain, while the second identity expresses that evaluating a composite term (represented by the right leg of the square) is the same as evaluating the subterms and combining the results using the algebra operations (the left leg).

Given an algebra (A, α), we will use the notation for compositions of forests also for the corresponding operations in (A, α), e.g. given $a \in A_0$ and $b \in A_1$, we will write $a \oplus b \in A_1$ for $\alpha_1(a \oplus b)$, $b \cdot a \in A_0$ for $\alpha_0(ba)$, $b^\omega \in A_0$ for the infinite power $\alpha_0((b\square)^\omega)$ and so on. Furthermore, if A is finite, the context sort of (A, α) forms a finite semigroup under the operation of vertical composition. Hence it makes sense to define, for $b \in A_1$, $b^\pi \in A_1$ as the unique idempotent of the subsemigroup generated by b.

Note that the identities required for the singleton and flattening maps ensure, among others, that $(\mathbb{F}S, \mathrm{flat}_S)$ is an ω-forest algebra for every biset S.

Definition 2.15. Given ω-forest algebras (A, α) and (B, β), a *homomorphism* $\phi\colon (A, \alpha) \to (B, \beta)$ is a biset map $\phi\colon A \to B$ such that the following square commutes.

$$\begin{array}{ccc} \mathbb{F}A & \xrightarrow{\mathbb{F}(\phi)} & \mathbb{F}B \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ A & \xrightarrow{\phi} & B \end{array}$$

Observe that the second identity from Definition 2.14 also implies that the map α is itself a homomorphism $\alpha\colon (\mathbb{F}A, \mathrm{flat}_A) \to (A, \alpha)$.

By a standard result from semigroup theory, the operation b^π is preserved by ω-forest algebra homomorphisms, since they preserve vertical composition.

Another standard result is that the algebra $(\mathbb{F}S, \mathrm{flat}_S)$ is the free algebra over the biset S: for every algebra (A, α) and a map $f\colon S \to A$, there is a unique homomorphism $f^\sharp\colon (\mathbb{F}S, \mathrm{flat}_S) \to (A, \alpha)$ such that

$f^\sharp \circ \mathrm{sing}_S = f$. This homomorphism equals $\alpha \circ \mathbb{F}(f)$, as might be expected. *Every* homomorphism $h\colon (\mathbb{F}S, \mathrm{flat}_S) \to (A, \alpha)$ is actually of the form $f^\sharp$, for $f = h \circ \mathrm{sing}_S$.

3 Language theory

In this chapter, we introduce a framework of recognisability for languages of infinite forests, including a notion of regularity for ω-forest algebras and the construction of syntactic algebras.

Definition 3.1. Given a biset Σ, a *language* L over Σ is a sub-biset of $\mathbb{F}\Sigma$, i.e. a biset such that $L_0 \subseteq \mathbb{F}_0\Sigma$ and $L_1 \subseteq \mathbb{F}_1\Sigma$. In this situation, we will write $L \subseteq \mathbb{F}\Sigma$.

As usual, we say that a language $L \subseteq \mathbb{F}\Sigma$ is *recognized* by a homomorphism $\phi\colon (\mathbb{F}S, \mathrm{flat}_S) \to (A, \alpha)$ into an algebra (A, α) if there is a subset $T \subseteq A$ such that $L = \phi^{-1}(T)$ (where the inverse image is taken in both components separately).

3.1 Regular languages and automata

We adapt the approach of Blumensath (2018b) in characterizing the class of algebras that recognize precisely the regular forest languages. We start by fixing our terminology regarding regular languages, i.e. languages accepted by a finite automaton defined as follows.

Definition 3.2. Let Σ be an alphabet. A *nondeterministic forest automaton* over Σ is a tuple $(Q, \Delta_0, \Delta_1, I, \Omega)$ with components as follows.

1. Q is a finite semigroup of *states* whose (not neccessarily commutative) operation shall be denoted by $+$.

2. $\Delta_0 \subseteq Q \times \Sigma_0^{\square}$ is the *nullary transition relation*.

3. $\Delta_1 \subseteq Q \times \Sigma_1^{\square} \times Q$ is the *non-nullary transition relation*.

4. $I \subseteq Q$ is the set of *initial states*.

5. $\Omega\colon Q \to \mathbb{N}$ is the *priority mapping*.

To define the language accepted by an automaton, we first need the notion of a run (both on forests and on contexts). From now on, given a set X, we may simply write X as a shorthand for the biset (X, X).

Definition 3.3. For an automaton $\mathcal{A}$ as in the previous definition and a forest $t \in \mathbb{F}_i\Sigma$, a *run* of $\mathcal{A}$ on t is a forest $r \in \mathbb{F}_0 Q$ of the same shape (i.e. with the same domain) which satisfies the following conditions.

1. For every $w \in \text{Leaf}(t)$, we have $(r(w), t(w)) \in \Delta_0$.
2. For every $w \in \text{Int}(t)$ and $w_1, \dots, w_n$ the successor vertices of w, we have $(r(w), t(w), r(w_1) + \cdots + r(w_n)) \in \Delta_1$.

If r is a run, $w_1, \dots, w_n$ are the roots of r and $q = r(w_1) + \cdots + r(w_n)$, we say that the run R is *from* the state q. The set of all runs of $\mathcal{A}$ on t will be denoted $\text{Run}_{\mathcal{A}}(t)$.

An *infinite branch* of a forest t' over an arbitrary biset X is an infinite sequence of vertices $w_1, w_2, \dots$ of t' such that w_1 is a root and each w_{i+1} is a successor of w_i. We say that a run r on $t \in \mathbb{F}_i S$, $i = 0, 1$ is *accepting* if it starts in a state $q \in I$ and satisfies the following parity condition.

3. For every infinite branch $(w_i)_i$ of t, $\liminf_{i \in \mathbb{N}_+} \Omega(r(w_i))$ is even.

Finally, the *nullary language* accepted by $\mathcal{A}$ is the set

$$\text{Lang}_0(\mathcal{A}) := \{t \in \mathbb{F}_0\Sigma \,|\, \text{there is an accepting run of } \mathcal{A} \text{ on } t\},$$

the *non-nullary language* is the set $\text{Lang}_1(\mathcal{A}) \subseteq \mathbb{F}_1\Sigma$ defined similarly and the *language* accepted by the automaton $\mathcal{A}$ is the biset $\text{Lang}(\mathcal{A})$ defined as $(\text{Lang}_0(\mathcal{A}), \text{Lang}_1(\mathcal{A}))$.

The definition given above is a fairly standard definition of a parity automaton on unranked forests. Let us highlight two points of particular interest.

First, to deal with arbitrary branching and multiple trees in forests, we impose the structure of a semigroup on the state space, as in (Bojańczyk and Idziaszek, 2009). Note that this approach is more general, since the "usual" definition of tree automata, with every state and label prescribing a fixed number of successors and automaton states, can be recovered as a special case of this definition by taking Q to be the finite quotient of the free semigroup on the state space obtained by identifying all strings longer than the longest right side of a rule in Δ_1.

Second, the same automaton is used to accept both forests and contexts simply by treating the hole as a special leaf which may or may not be accepted by a state q.

Let us recall the following well-known facts about regular languages.

Fact 3.4. *Regular languages are closed under unions, complementation, intersections, substitutions and inverse substitutions, i.e. direct and inverse images of homomorphisms of the form $\mathbb{F}(f): \mathbb{F}\Sigma \to \mathbb{F}\Pi$.*

3.2 Regular forest algebras

We can now describe the class of algebras characterising regular languages. Our definition is a straightforward adaptation of the notion of a regular algebra by Blumensath (2018b).

Definition 3.5. An ω-forest algebra (A, α) is *regular* if it is finite and for both sorts i and every $a \in A_i$, the set $\alpha_i^{-1}(a)$ is a regular language (we treat this set as a forest language by considering the other sort to be empty).

Since we have only finitely many sorts, the finitarity requirement of Blumensath (2018b) translates to simple finiteness, which means that the entire algebra A can be taken as the required finite set of generators for concreteness, as regularity does not depend on the particular choice.

The following theorem is a straightforward reformulation of the one in (Blumensath, 2018b).

Theorem 3.6. *A finite algebra (A, α) is regular if and only if all languages it recognises are regular.*

Proof. In one direction, if (A, α) is nonregular, i.e. if there is $a \in A_i$ such that $\alpha_i^{-1}(a)$ is not regular, then $\alpha: (\mathbb{F}A, \mathrm{flat}_A) \to (A, \alpha)$ is a homomorphism recognizing a nonregular language.

In the other direction, we observe that if $h^\sharp: (\mathbb{F}\Sigma, \mathrm{flat}_\Sigma) \to (A, \alpha)$ is a homomorphism, for $h: \Sigma \to A$, it follows from the fact that $h^\sharp$ equals $\alpha \circ \mathbb{F}(h)$ that any language $(h^\sharp)^{-1}(S)$, for $S \subseteq A$, is an inverse substitution of a finite union of regular languages, which is regular. □

That regular forest algebras characterise regular languages is expressed by the following theorem.

Theorem 3.7. *A language $L \subseteq \mathbb{F}\Sigma$ is regular if and only if it is recognised by a regular forest algebra.*

In one direction, the proof is obvious; by Theorem 3.6, a language recognised by a regular algebra is regular. In the other direction, it is sufficient to construct, from an automaton $\mathcal{A}$ recognising $L \subseteq \mathbb{F}\Sigma$, a regular algebra and a homomorphism recognising the same language. This construction will be described now.

To construct a homomorphism recognising a language accepted by an automaton, we need the following simple construction. Given a biset S, we can apply the power set operation to both components to obtain the biset $\mathcal{P}S := (\mathcal{P}S_0, \mathcal{P}S_1)$. We then have the relation $\in$ from S to $\mathcal{P}S$ whose components are given by (the appropriate restrictions of) the ordinary containment relation $\in$. The construction will make use of the extended relation $\in^{\mathbb{F}}$ from $\mathbb{F}S$ to $\mathbb{F}\mathcal{P}S$.

Theorem 3.8. *Every language $L \subseteq \mathbb{F}\Sigma$ accepted by a finite automaton $\mathcal{A}$ is recognised by a regular algebra* $\mathrm{M}(\mathcal{A})$.

Proof. As in (Blumensath, 2018b), the algebra $\mathrm{M}(\mathcal{A})$ will simulate the automaton $\mathcal{A}$, with its product corresponding to composition of runs of $\mathcal{A}$.

Given a forest $t \in \mathbb{F}_0\Sigma$, what is the relevant information about possible behaviors of $\mathcal{A}$ on t, i.e. what are the *profiles* of $\mathcal{A}$? We need to keep track of the states in which a successful run r on t can start. We do not have to track the priorities, however, since if we want to compose r with some with some previous behavior, any infinite branch of this larger run passing through r can only have a finite prefix outside r and this prefix can be ignored with regard to the limit. This suggest that elements of $\mathrm{M}(\mathcal{A})_0$ should simply be sets of states of $\mathcal{A}$.

For a context $t \in \mathbb{F}_1\Sigma$, the starting state is still relevant. However, we also need to keep track of the state in which the automaton is right before the hole, so that we can decide if two runs actually "fit together". Furthermore, since we in general need to compose an infinite number of such context runs, we also need to track the priorities; since the limit taken over infinite branches is a limit inferior, we need to remember the *lowest* priority encountered in the run on the path from the initial state to the hole state; the other priorities may be ignored as in the previous paragraph. Therefore, the elements of $\mathrm{M}(\mathcal{A})_1$ should be sets of triples (q, n, q'), where q and q' are the extremal states and n is the remembered priority.

Therefore, let $D \subseteq \mathbb{N}$ be the range of the priority map Ω and let $K_{\mathcal{A}}$ be the biset $(Q, Q \times D \times Q)$. Intuitively, a forest $t \in \mathbb{F}_0 K_{\mathcal{A}}$ in which all the profiles fit together and where the parity condition is satisfied has a defined product given by the sum of the root states, and similarly for $t \in \mathbb{F}_1 K_{\mathcal{A}}$. The carrier of $\mathrm{M}(\mathcal{A})$ will be $\mathcal{P}K_{\mathcal{A}}$ and the multiplication map $\mathrm{comp}\colon \mathbb{F}\mathcal{P}K_{\mathcal{A}} \to \mathcal{P}K_{\mathcal{A}}$ will extract from a forest labelled by sets of profiles the composable forests of profiles and return the set of resulting profiles.

We will start by fixing some terminology. For a forest $t \in \mathbb{F}_0 K_{\mathcal{A}}$ and $w \in \mathrm{Dom}(t)$, define $\ell_w \in Q$ to be $t(w)$ if $w \in \mathrm{Leaf}(t)$ and q when $w \in \mathrm{Int}(t)$ and w is labelled by (q, n, q'). We say that t is *consistent* if for every $w \in \mathrm{Int}(t)$ with label (q, n, q') and sucessors $w_1, \dots, w_k$, we have $q' = \ell_{w_1} + \cdots + \ell_{w_k}$.

Furthermore, we say that the forest t *satisfies the parity condition* if for every infinite branch $(w_i)_i$ of t with labels (q_i, n_i, q_i'), we have that $\liminf_i n_i$ is even.

Finally, define $\mathrm{Init}(t) \in Q$ to be the state $\sum_{w \in \mathrm{Root}(F)} \ell_w$ (the sum is computed in the natural order of the roots).

Putting these concepts together, we can define the nullary part of the algebra product, the map $\mathrm{comp}_0\colon \mathbb{F}_0 \mathcal{P}K_{\mathcal{A}} \to \mathcal{P}Q$, by

$$\begin{aligned}\mathrm{comp}_0(p) := \{\mathrm{Init}(t) \mid t \in \mathbb{F}_0 K_{\mathcal{A}}, t \in_0^{\mathbb{F}} p, t \text{ is consistent and} \\ \text{satisfies the parity condition}\}\end{aligned}$$

The map $\mathrm{comp}_1\colon \mathbb{F}_1 \mathcal{P}K_{\mathcal{A}} \to \mathcal{P}(Q \times D \times Q)$ can be defined as follows.

Given a forest $t \in \mathbb{F}_1 K_{\mathcal{A}}$, we say that a triple $(q, n, q') \in Q \times D \times Q$ *corresponds* to t if $\mathrm{Init}(t) = q$, the plain forest obtained from t by labelling the hole with q' is consistent and satisfies the parity condition and n is the lowest priority on the path from the corresponding root to the leaf which was the hole in t.

The map can then be defined by setting

$$\begin{aligned}\mathrm{comp}_1(p) := \{(q, n, q') \mid &\text{there is a forest } t \in \mathbb{F}_1 K_{\mathcal{A}} \text{ such that} \\ &t \in_1^{\mathbb{F}} p \text{ and } (q, n, q') \text{ corresponds to } t\}\end{aligned}$$

We need to verify that these two multiplication maps make $\mathcal{P}K_{\mathcal{A}}$ into a forest algebra.

The first condition is verified easily. First, for every $X \subseteq Q$, the forests $t \in \mathbb{F}_0 K_{\mathcal{A}}$ such that $t \in_0^{\mathbb{F}} \mathrm{sing}_{K_{\mathcal{P}\mathcal{A}},0}(X)$ are precisely the singleton forests $\mathrm{sing}_{K_{\mathcal{A}},0}(q)$ for $q \in X$. All these forests are trivially consistent and satisfy the parity condition, and since $\mathrm{Init}(\mathrm{sing}_{K_{\mathcal{A}},0}(q)) = q$, we have $\mathrm{comp}_0(\mathrm{sing}_{\mathcal{P}K_{\mathcal{A}},0}(X)) = X$.

Similarly, for $X \subseteq Q \times D \times Q$, the forests $\mathrm{sing}_{K_{\mathcal{A}},1}(q,n,q')$ for every triple $(q,n,q') \in X$ are precisely the ones related by $\in_1^{\mathbb{F}}$ to $\mathrm{sing}_{\mathcal{P}K_{\mathcal{A}},1}(X)$ and to each of these forests corresponds precisely the single triple (q,n,q'), hence $\mathrm{comp}_1(\mathrm{sing}_{\mathcal{P}K_{\mathcal{A}},1}(X)) = X$.

To verify the associative law, first consider a forest $p \in \mathbb{F}_0\mathbb{F}\mathcal{P}K_{\mathcal{A}}$ and assume we have $q \in \mathrm{comp}_0(\mathbb{F}_0(\mathrm{comp})(p))$. The argument proceeds in the following steps.

1. There exists a forest $t \in \mathbb{F}_0 K_{\mathcal{A}}, t \in_0^{\mathbb{F}} \mathbb{F}_0(\mathrm{comp})(p)$ which is consistent and satisfies the parity condition and has $\mathrm{Init}(t) = q$.

2. Therefore for every $w \in \mathrm{Leaf}(t)$, there is a forest $t_w \in \mathbb{F}_0 K_{\mathcal{A}}$, $t_w \in_0^{\mathbb{F}} p(w)$ with $\mathrm{Init}(t_w) = t(w)$. Likewise, for $w \in \mathrm{Int}(t)$, there is a context $t_w \in \mathbb{F}_1 K_{\mathcal{A}}, t_w \in_1^{\mathbb{F}} p(w)$ such that the triple $t(w)$ corresponds to t_w. By taking the domain of t, which is identical to that of p, and labelling each vertex w by t_w, we obtain a new forest $p' \in \mathbb{F}_0\mathbb{F}K_{\mathcal{A}}$ with the property that $p' \ (\in^{\mathbb{F}})_0^{\mathbb{F}} \ p$ (here, the relation is extended twice). It follows that $\mathrm{flat}_{K_{\mathcal{A}},0}(p') \in_0^{\mathbb{F}} \mathrm{flat}_{\mathcal{P}K_{\mathcal{A}},0}(p)$.

3. The forest $t' := \mathrm{flat}_{K_{\mathcal{A}},0}(p')$ is consistent; if both an internal vertex and all of its successors come from the same forest $p'(w)$, the equality is guaranteed by consistency of t_w, if w comes from a vertex that precedes the hole of some t_v, its successors are

   $$a_1, \ldots, a_n, b_1, \ldots, b_m, c_1, \ldots, c_k$$

   where the vertices a_i, c_i come from t_v and the ones denoted b_i come from the roots of all the forests labelling the successor vertices of v. Since the triple $t(v) = (q,n,q')$ corresponds to t_v and the forest t is consistent, we have $t'_{b_1} + \cdots + t'_{b_m} = q'$; since the forest obtained from t_v by labelling the hole with q' is also consistent, the state $a_1 + \cdots + a_n + q' + c_1 + \cdots + c_k$ is equal to the corresponding state of the triple above the hole of t_v.

The forest t' also satisfies the parity condition; given an infinite branch $(w_i)_i$ and a sequence of corresponding labels $(q_i, n_i, q_i')_i$, either from some index j on, all the subsequent vertices come from a single forest t_v which itself satisfies the condition and removing a finite prefix of the sequence does not affect the limit, or we can partition the sequence into infinitely many finite parts

$$n_{1,1}, \dots, n_{1,k_1}, n_{2,1} \dots n_{2,k_2}, \dots$$

where each $n_{i,1}, \dots, n_{i,k_i}$ is the sequence of priorities occuring on the path from root to hole of a forest $p'(v_i)$ for some infinite branch $(v_i)_i$ of p'. Since the sequence $(m_i)_i$ obtained by choosing from each finite part the least element has even limit inferior (the forest t satisfies the parity condition), it follows that $\liminf_i n_i$ is also even.

Finally, it can be easily verified that $\mathrm{Init}(t') = q$ and therefore $q \in \mathrm{comp}_0(\mathrm{flat}_{\mathcal{P}K_{\mathcal{A}},0}(p))$.

The previous argument may be easily inverted to show the opposite inclusion, and the argument for the map comp_1 is essentially similar.

We have shown that $\mathrm{M}(\mathcal{A})$ is a forest algebra. The next step is to construct the required homomorphism. Since $(\mathbb{F}\Sigma, \mathrm{flat}_\Sigma)$ is the free forest algebra over Σ, we only need to define a biset map $f : \Sigma \to \mathcal{P}K_{\mathcal{A}}$, which we do by setting

$$f_0(a) := \{q \in Q \mid (q, a) \in \Delta_0\}$$

and

$$f_1(b) := \{(q, n, q') \mid (q, b, q') \in \Delta_1, n = \min\{\Omega(q), \Omega(q')\}\}.$$

The accepting set $T \subseteq \mathcal{P}K_{\mathcal{A}}$ is defined as

$$T_0 := \{X \subseteq Q \mid X \cap I \neq \emptyset\}$$

and

$$T_1 := \{X \subseteq Q \times D \times Q \,|\, \text{there exists } (q,n,q') \in X \text{ such that } q \in I \text{ and } (q', \square) \in \Delta_0\}$$

and it remains to argue that $(f^\sharp)^{-1}(T) = \mathrm{Lang}(\mathcal{A})$.

Assume $t \in \mathrm{Lang}_0(\mathcal{A})$, i.e. there is an accepting run R of $\mathcal{A}$ on t. Since $f_0^\sharp(t) = \mathrm{comp}_0(\mathbb{F}_0(f)(t))$, it is easily seen that there is a consistent forest $t' \in_0^{\mathbb{F}} \mathbb{F}_0(f)(t)$ which satisfies the parity condition and has $\mathrm{Init}(t') \in I$. Conversely, given $q \in f_0^\sharp(t) \cap I$, one can use the fact that $f_0^\sharp(t) = \mathrm{comp}_0(\mathbb{F}_0(f)(t))$ to construct an accepting run of $\mathcal{A}$ on F. The argument for $f_1^\sharp$ is similar.

To finish the proof, we have to show that the algebra $\mathrm{M}(\mathcal{A})$ is regular.

For any given state q_0 of $\mathcal{A}$, an automaton that recognises the set of forests $t \in \mathbb{F}_0\mathrm{M}(\mathcal{A})$ such that $q_0 \in \mathrm{comp}_0(t)$ can be constructed in a straightforward manner. For its semigroup, we take $Q \times D$, with the operation defined by $(q,n) + (q',n') := (q + q', n)$ (the choice of operation for the priorities does not really matter here). The priority mapping is given simply by projection onto the second argument, while the sets of transitions Δ'_0 and Δ'_1 are defined as

$$\Delta'_0 := \{((q,n),X) \in (Q \times D) \times \mathrm{M}(\mathcal{A})_0 \,|\, q \in X\}$$

and

$$\Delta'_1 := \{((q,n),X,(q',n')) \in (Q \times D) \times \mathrm{M}(\mathcal{A})_1 \times (Q \times D) \,|\, (q,n,q') \in X\}.$$

For the set of accepting states, we take $\{q_0\} \times D$.

We can also construct, for every triple $(q_0, n_0, q'_0) \in Q \times D \times Q$, an automaton recognising the set of contexts $t \in \mathbb{F}_1\mathrm{M}(\mathcal{A})$ such that $(q_0, n_0, q'_0) \in \mathrm{comp}_1(t)$. In this case, the construction is essentially similar: for the semigroup, we take the set $\{0,1\} \times Q \times D \times \mathcal{P}(D)$ extended with an error value which we will denote $\bot$. Intuitively, the first coordinate specifies whether the automaton expects a forest or a context, the second and third components are the expected state and priority and the last component keeps track of the priorities encountered on the path to the hole (it is empty if the automaton expects a forest).

The operation on states is given by

$$\begin{aligned}(0,q,n,\emptyset) + (0,q',n',\emptyset) &:= (0,q+q',n,\emptyset)\\(1,q,n,P) + (0,q',n',\emptyset) &:= (1,q+q',n,P)\\(0,q,n,\emptyset) + (1,q',n',P) &:= (1,q+q',n,P)\end{aligned}$$

(in all other cases, the result is $\perp$) and the transitions by

$$((1,q_0',n,\emptyset),\square) \in \Delta_0' \text{ for every } n \in D$$
$$((0,q,n,\emptyset),X) \in \Delta_0' \text{ if } q \in X$$

and

$$((1,q,n,P \cup \{n\}),X,(1,q',n',P)) \in \Delta_1' \text{ if } (q,n,q') \in X.$$

For the priority map, we take the priority of an element $(0,q,n,P)$ to be n, of $(1,q,n,P)$ to be 1 (to ensure that the automaton accepts only contexts) and of $\perp$ (arbitarily) also 1. Finally, for the accepting states we take those elements $(1,q_0,n,P)$ such that P is nonempty and n_0 is the smallest element of P.

The automata recognising $\mathrm{comp}_{\mathcal{A},0}^{-1}(X)$ and $\mathrm{comp}_{\mathcal{A},1}^{-1}(X)$ can then be constructed using closure under complementations and intersections. $\square$

We have shown the other direction, namely that for a regular language L we can construct a regular algebra recognizing L, thus proving Theorem 3.7.

3.2.1 Closure properties of regular algebras & their equational definition

As the following theorem shows, the class of regular ω-forest algebras forms a pseudovariety. By an abstract result of Chen et al. (2016, Theorem 4.12), this pseudovariety is specified by a system of equations of "profinite ω-forests". This suggests possible future work in determining what such equations look like and describing the system that defines regular algebras.

Theorem 3.9. *The class of regular algebras is closed under subalgebras, homomorphic images and finite products.*

Proof. Let us first show that a subalgebra S of a regular algebra (A, α), i.e. a subset S such that for every $t \in \mathbb{F}_i S$, we have $\alpha_i(t) \in S_i$, is again regular. Given an element $a \in S_i$, the language $\alpha_i^{-1}(a) \subseteq \mathbb{F}_i A$ is regular by assumption. We need to show that $\alpha_i^{-1}(a) \cap \mathbb{F}_i S$ is regular, which is easily seen, since $\mathbb{F}_i S \subseteq \mathbb{F}_i A$ is recognised by the automaton which rejects forests containing a label outside of S and accepts all others.

Next, we need to show that given a surjective homomorphism $h\colon (A, \alpha) \twoheadrightarrow (B, \beta)$ with (A, α) regular, the algebra (B, β) is also regular. As is easily seen, we have $\beta_i^{-1}(b) = \mathbb{F}_i(h)(\alpha_i^{-1}(h_i^{-1}(b)))$, which is regular since it is an image under substitution of a finite union of regular languages (since $h_i^{-1}(b)$ is finite).

Finally, we have to show that the final algebra **1** is regular (which is obvious) and that given regular algebras (A, α) and (B, β), their product is also regular. Let us denote the multiplication of the product algebra $\gamma\colon \mathbb{F}(A \times B) \to (A \times B)$. Then the required inverse image $\gamma_i^{-1}(a, b)$ equals $(\mathbb{F}_i(\pi_1))^{-1}(\alpha_i^{-1}(a)) \cap (\mathbb{F}_i(\pi_2))^{-1}(\beta_i^{-1}(b))$, where $\pi_1\colon A \times B \to A$ and $\pi_2\colon A \times B \to B$ are the projection homomorphisms. Therefore, $\gamma_i^{-1}(a, b)$ is an intersection of inverse substitutions of regular languages, which is again regular. □

3.3 The syntactic algebra

The usual notion of a syntactic congruence of a language L and its associated syntactic algebra might not be well-behaved when dealing with infinite objects; for languages of infinite forests, the equivalence obtained might fail to actually be a congruence. However, following (Blumensath, 2018b), we can show that for regular languages, a universal recognizing algebra can be constructed and is, in fact, regular.

Definition 3.10. Given an algebra (A, α), a *congruence* of (A, α) is an equivalence relation $\sim$ on the biset A with the property that for both sorts i and every pair of forests $t_1, t_2 \in \mathbb{F}_i A$ with $t_1 \sim_i^{\mathbb{F}} t_2$, we have $\alpha_i(t_1) \sim_i \alpha_i(t_2)$.

The following definitions are adapted from (Bojańczyk, 2015). They can be naturally made fully formal if required.

Definition 3.11. For an algebra (A, α), and a sort i a (single-variable) *polynomial* of sort i is a forest (or a context) p labelled by elements of

A plus a single fresh label of sort i, which will be denoted x. The biset of such polynomials will be denoted $\mathrm{Pol}((A, \alpha), i)$.

Given an element $a \in A_i$, we define $p(a)$ to be the product, in (A, α), of the forest obtained by replacing each x with a.

Definition 3.12. Let (A, α) be an algebra and $L \subseteq A$ its subset. The *syntactic equivalence* of L is the relation $\sim_L$ given by setting $a_1 \sim_{L,i} a_2$ if for both sorts j and every polynomial $p \in \mathrm{Pol}((A, \alpha), i)_j$, either both $p(a_1)$ and $p(a_2)$ are in L_j or neither is.

Lemma 3.13. *For every finite algebra (A, α) and $L \subseteq A$, the syntactic equivalence $\sim_L$ is a congruence.*

Proof. Intuitively, replacing a product $\alpha_i(t)$ by $\alpha_i(t')$ in a polynomial p can be accomplished by replacing the individual labels of t, of which there is only a finite number, by those of t'.

Suppose we have forests $t, t' \in \mathbb{F}_i A$, and that $t\ (\sim_L^{\mathbb{F}})_i\ t'$. Since both forests have only finitely many distinct labels, there is a partition $D_1, \dots, D_n$ of their domain D and a sequence of pairs of elements $(a_1, a'_1), \dots, (a_n, a'_n)$ such that for every k, all the vertices from D_k are labelled by a_k in t and by a'_k in t' and $a_k \sim_{L,i_k} a'_k$, where $i_k \in \{0, 1\}$ is the appropriate arity.

Let $p \in \mathrm{Pol}(\mathcal{A}, i)_j$ be a polynomial; we need to demonstrate that $[p](\alpha_i(F))$ is in L_j if and only if $[p](\alpha_i(F'))$ is. Define a sequence of forests $t_0, \dots, t_n$ by setting $t_0 := t$ and letting t_{k+1} be the forest obtained from t_k by relabelling all the vertices in D_{k+1} (which are labelled by a_{k+1}) by a'_{k+1}. Then $t_n = t'$. Also, define a sequence of polynomials $p_1, \dots, p_n$, $p_k \in \mathrm{Pol}(\mathcal{A}, i_k)_i$, where p_k is obtained by labelling D_k in t_k by the single variable of sort i_k, so that for $0 \le k < n$, we have $[p_{k+1}](a_{k+1}) = \alpha_i(t_k)$ and $[p_{k+1}](a'_{k+1}) = \alpha_i(t_{k+1})$. As is easily seen, we have

$$[p](\alpha_i(t_k)) = [p]([p_{k+1}](a_{k+1})) = [[p](p_{k+1})](a_{k+1})$$

and

$$[p](\alpha_i(t_{k+1})) = [p]([p_{k+1}](a'_{k+1})) = [[p](p_{k+1})](a'_{k+1})$$

for every $0 \le k < n$ and since each a_{k+1} is equivalent to a'_{k+1}, the element $[p](\alpha_i(t_k))$ is in L_j if and only if $[p](\alpha_i(t_{k+1}))$ is. By transitivity, $[p](\alpha_i(t))$ is in L_j if and only if $[p](\alpha_i(t'))$ is. $\square$

The central statement of this section is the following. The proof is analogous to that of Blumensath (2018b).

Theorem 3.14. *If $L \subseteq \mathbb{F}\Sigma$ is a regular language, then $\sim_L$ is a congruence on $\mathbb{F}\Sigma$ and the quotient algebra* $\mathrm{Synt}(L) := \mathbb{F}\Sigma/{\sim_L}$ *is regular and, together with the quotient homomorphism $p_{\sim_L} \colon \mathbb{F}\Sigma \to \mathrm{Synt}(L)$, recognizes L. Moreover, it satisfies the usual universal property of syntactic algebras: for every surjective homomorphism $\phi \colon (\mathbb{F}\Sigma, \mathrm{flat}_\Sigma) \twoheadrightarrow (A, \alpha)$ which recognises L, there is a unique homomorphism $g \colon (A, \alpha) \twoheadrightarrow \mathrm{Synt}(L)$ such that the following diagram commutes.*

$$\begin{array}{ccc} \mathbb{F}\Sigma & \xrightarrow{p_{\sim_L}} & \mathrm{Synt}(L) \\ & \searrow^{\phi} & \uparrow g \\ & & A \end{array}$$

4 The logic EF

Our final goal is to characterise regular languages definable in the logic EF (a fragment of CTL) by transfering the proof of Bojańczyk and Idziaszek (2009) into our framework. Our definition of EF formulae over an alphabet Σ is taken from there.

Definition 4.1. The set of *forest formulae* over an alphabet Σ is defined inductively as follows.

1. For both sorts i and every $a \in \Sigma_i$, the atomic formula a is a forest formula.

2. If ϕ and ψ are formulae, their disjunction $\phi \vee \psi$ is also a formula. Likewise, if ϕ is a formula, so is its negation $\neg\phi$.

3. For every formula ϕ, **EF** ϕ is also a formula. Here, *EF* stands for *exists finally*.

Such formulae will be interpreted over forests; for contexts, we define *context formulae* over Σ to be simply forest formulae over $\Sigma^{\square}$.

Our definition of satisfaction of a formula ϕ by a tree t is standard.

1. A tree t satisfies an atomic formula a if the root label is a.

2. The semantics of $\vee$ and $\neg$ are standard.

3. A formula of the form **EF** ϕ is satisfied by t if a proper subtree of t satisfies ϕ.

For forests with more than one tree, however, we change the definition of Bojańczyk and Idziaszek (2009) slightly.

1. No atomic formula a is satisfied by a forest t with at least two trees.

3. A formula of the form **EF** ϕ is satisfied by such a forest t simply if there is a subtree of t satisfying ϕ.

Intuitively, according to our definition forests behave as trees with an "invisible" root vertex. In effect, a formula of the form **EF** ψ is true over a forest according to our semantics if and only if **EF*** ψ is true over the same forest according to Bojańczyk and Idziaszek (2009).

We also define the formulae **EF*** ϕ, **AG** ϕ (always globally) and **AG*** ϕ as abbreviations for $\phi \vee$ **EF** ϕ, $\neg$**EF** $\neg\phi$ and $\phi \wedge$ **AG** ϕ respectively.

As in (Bojańczyk and Idziaszek, 2009), we define a notion of bisimilarity of forests which corresponds to equivalence under the logic EF.

Definition 4.2. Given an alphabet Σ, the relation $\sim_{\mathrm{EF}}$ of *EF-bisimilarity* can be defined on trees in $\mathbb{T}_0\Sigma$ as the greatest relation[1] satisfying the following properties.

1. If $s \sim_{\mathrm{EF}} t$, then the root labels of s and t are the same.

2. If $s \sim_{\mathrm{EF}} t$, then for every proper subtree $s|v$ of s (i.e. such that $|v| \geq 2$), there is a proper subtree $t|w$ of t such that $s|v \sim_{\mathrm{EF}} t|w$. Similarly, for every proper subtree $t|v$ of t, there is a proper subtree $s|w$ of w such that $s|w \sim_{\mathrm{EF}} t|v$.

Similarly, for each $n \in \mathbb{N}$, we have the relation $\sim_{\mathrm{EF}(n)}$ of n-bisimilarity, which can be defined by induction as follows:

1. If the root labels of s and t are the same, then $s \sim_{\mathrm{EF}(0)} t$.

2. If the root labels of s and t are the same and for every proper subtree $s|v$ of s there is a proper subtree $t|w$ of t such that $s|v \sim_{\mathrm{EF}(n)} t|w$ and vice versa, then $s \sim_{\mathrm{EF}(n+1)} t$.

To extend the definition of bisimilarity to forests with more than one tree, we simply say that two such forests s and t are bisimilar if each subtree $s|v$ of s has a bisimilar counterpart $t|w$ and vice versa, and analogously for n-bisimilarity – two forests are said to be $(n+1)$-bisimilar if every $s|v$ has a n-bisimilar counterpart $t|w$ and vice versa (we consider all proper forests to be 0-bisimilar).

Two contexts $s, t \in \mathbb{F}_1\Sigma$ are then said to be bisimilar if they are bisimilar as forests over $\Sigma^{\square}$.

1. This relation exists, since the defining properties are satisfied by the empty relation and are preserved by taking arbitrary unions.

There is a well-known characterisation of bisimilarity in game-theoretic terms; the *EF game* is played on a pair of forests $t_0, t_1 \in \mathbb{F}_0\Sigma$ by two players, Spoiler and Duplicator. In each round, Spoiler selects a forest t_i and a subtree $t_i|w$ of t_i (a proper subtree when t_i is a tree); if such choice cannot be made, i.e. if t_i is a tree consisting only of a root, Spoiler loses. Otherwise, Duplicator must counter Spoiler's move by selecting a (proper) subtree $t_{1-i}|v$ of t_{1-i} such that the root labels of $t_i|w$ and $t_{1-i}|v$ agree. If there is no such choice, Duplicator loses; otherwise, the game advances into the next round, which is played on the pair $t_i|w, t_{1-i}|v$. The forests t_0, t_1 are then said to be bisimilar if Duplicator has a winning strategy for this pair, i.e. if each play either lasts for infinitely many rounds or ends with Spoiler losing. It is easily seen that if we add a further condition, namely that if t_0 and t_1 are trees, then their root labels match, this definition becomes equivalent to Definition 4.2. Similarly, n-bisimilarity has a game-theoretic definition obtained by requiring Duplicator to last for n rounds instead of infinitely many.

In the following, we will call languages L such that $L_1 = \emptyset$ *pure forest languages*. Similarly, a *context language* is a language L such that $L_0 = \emptyset$.

We recall that a forest (context) $t \in \mathbb{F}_i\Sigma$ is *regular* if it has, up to isomorphism, only finitely many distinct subtrees. A well-known property of regular languages is that they are entirely determined by the regular forests they contain, much like regular languages of ω-words are determined by the ultimately periodic words they contain. This fact will be significant for the proof of the following lemma.

Lemma 4.3. *Let (A, α) be a regular algebra and $h\colon \Sigma \to A$ a map such that the homomorphism $h^\sharp\colon (\mathbb{F}\Sigma, \mathrm{flat}_\Sigma) \twoheadrightarrow (A, \alpha)$ is surjective. Assume that $h^\sharp_0$ and $h^\sharp_1$ are invariant under EF-bisimarity (i.e. $h^\sharp_i$ maps two bisimilar forests from $\mathbb{F}_i\Sigma$ to the same value) and that (A, α) satisfies the equation*

$$v^\pi \cdot a = (v \oplus (v^\pi \cdot a))^\omega \tag{4.1}$$

for all $v \in A_1, a \in A_0$. Then for every $a \in A_0$, there is a forest EF formula ϕ_a such that a forest $t \in \mathbb{F}_0\Sigma$ satisfies ϕ_a if and only if $h^\sharp_0(t) = a$.

Proof. We proceed as in (Bojańczyk and Idziaszek, 2009). We first observe some facts about the structure of the algebra A.

Since $h_0^\sharp$ is invariant under bisimilarity, it follows that the horizontal composition $\oplus$ on A_0 is commutative; indeed, we have

$$a \oplus b = h_0^\sharp(s) \oplus h_0^\sharp(t) = h_0^\sharp(s \oplus t)$$

for suitable forests $s, t \in \mathbb{F}_0\Sigma$ and the forests $s \oplus t$ and $t \oplus s$ are bisimilar. In the same manner, we can observe that $\oplus$ is idempotent and that the following equation is satisfied for every $a \in A_0$ and $c \in A_1$.

$$(c \cdot a) \oplus a = c \cdot a \tag{4.2}$$

Next, Bojańczyk and Idziaszek (2009) define a reachability relation on A_0 by calling an element $a \in A_0$ *reachable* from $b \in A_0$ if there is a $c \in A_1$ with $a = c \cdot b$. To mirror this definition in our setting, however, we have to add another condition, since in this work, we do not allow contexts in which the hole vertex is a root. We also choose to make reachability reflexive by definition. Therefore, we define a to be reachable from b if either

1. a is equal to b, or
2. there exists an element $c \in A_1$ such that $a = c \cdot b$, or
3. $a = b \oplus a'$ for some $a' \in A_0$.

Note that the other two naturally expected cases, when a is of the form $a' \oplus b$ or $a' \oplus b \oplus a''$, are subsumed by case 3, since $\oplus$ is commutative.

It is easily seen that reachability is transitive. We also need to show that it is antisymmetric, and hence a partial order. Therefore, assume that $a, b \in A_0$ are reachable from each other. If $a = c \cdot b$ and $b = d \cdot a$, we have

$$a = c \cdot b = c \cdot d \cdot a = (c \cdot d \cdot a) \oplus (d \cdot a) = a \oplus (d \cdot a) = (d \cdot a) = b$$

as in (Bojańczyk and Idziaszek, 2009). In the other cases, the proof is similar, e.g. if $a = c \cdot b$ and $b = a \oplus b'$, we have

$$a = c \cdot b = c \cdot (a \oplus b') = c \cdot (a \oplus b') \oplus (a \oplus b') = a \oplus a \oplus b' = a \oplus b' = b,$$

while if $a = b$, the proof is trivial.

Exactly as in (Bojańczyk and Idziaszek, 2009), we can also show that there is a unique element $\bot$ reachable from every $a \in A_0$, the *minimal* element of A_0. An element $b \in A_0, b \neq \bot$ is called *subminimal* if only b and $\bot$ are reachable from b.

After these remarks, we are able to prove the lemma by induction on $|A_0|$. We take ϕ_a to be an arbitrary tautology if $|A_0| = 1$. For the induction step, we distinguish several cases.

1. If $a = \bot$, we simply take the conjunction of the negations of the formulae obtained in the other cases.

2. If a is the unique subminimal element, we will obtain the required formula ϕ_a by the construction shown below.

3. If a is subminimal and there are other subminimal elements, or if a is neither minimal or subminimal, we will take the quotient of A by an appropriate congruence and proceed by induction on this smaller algebra; this step will also be described below.

(2.) To construct the formula ϕ_a in case 2, we observe that thanks to case 3, we have a formula ϕ_f defining $(h^\sharp)_0^{-1}(f)$ for every f in the set $F := A_0 - \{\bot, a\}$; this is the set of elements above a in the reachability relation. Hence, we also have the formula $\phi_F := \bigvee_{f \in F} \phi_f$, which charaterises the set $(h^\sharp)_0^{-1}(F)$.

Next, we define a partition of the vertices of a tree t into *components* by saying that two vertices x and y belong to the same component if $t|x$ is a subtree of $t|y$ and vice versa. Using the same arguments as in (Bojańczyk and Idziaszek, 2009), we can observe:

1. If t is a regular forest, it has only finitely many components.

2. If x and y are in the same component, the trees $t|x$ and $t|y$ are bisimilar; hence we can define the *type* of a component $[x]$ to be the value $h_0^\sharp(t|x)$.

3. We call a tree t *prime* if it has exactly one component whose type lies outside of F. In such a case, this component is necessarily the root component, by the definition of reachability. Since every component is either a *singleton* one, with a single vertex,

or a *proper* one, with infinitely many vertices, we analogously speak about either *singleton* or *proper* prime trees.

The *profile* of a prime tree t is a pair in $\mathcal{P}(F) \times (\Sigma_\cup + \mathcal{P}(\Sigma_\cup))$ (recall that $\Sigma_\cup$ is the union of the components of Σ); the first component stores the set of types of components of t which have type in F, the second one stores the labels of vertices in the root component – either a single label if the tree is singleton prime, or a set of labels (possibly a singleton set) if the tree is proper prime.

Again, we make some observations, whose proofs can be lifted directly from (Bojańczyk and Idziaszek, 2009).

1. If two prime trees s, t have the same profile, then $h_0^\sharp(s) = h_0^\sharp(t)$. Hence, for $b \in A_0$ we can define the set P_b of profiles of prime trees t with $h_0^\sharp(t) = b$.
2. For every profile p, there is an EF formula ϕ_p which is satisfied by every prime tree with profile p and such that any regular forest which satisfies ϕ_p has type b if $p \in P_b$ and $\bot$ otherwise.

Using these facts, we can construct the formula ϕ_a. Observe that the formula $\mathbf{AG}\ \phi_F$ is satisfied by a regular forest t precisely if t does not have a prime subtree. Hence we can take ϕ_a to be the disjunction of the formulae $(\mathbf{AG}\ \phi_F) \wedge \phi_{a,\text{nonprime}}$ and $(\neg \mathbf{AG}\ \phi_F) \wedge \phi_{a,\text{prime}}$, where $\phi_{a,\text{nonprime}}$ characterizes forests without a prime subtree and $\phi_{a,\text{prime}}$ those that have a prime subtree. We can take $\phi_{a,\text{nonprime}}$ to be

$$\Big(\neg \bigvee_{G \subseteq F, \sum G = \bot}\ \bigwedge_{g \in G} \mathbf{EF}\ \phi_g\Big) \wedge \Big(\bigvee_{G \subseteq F, \bigoplus G = a}\ \bigwedge_{g \in G} \mathbf{EF}\ \phi_g\Big),$$

where $\bigoplus\{g_1, \dots, g_n\}$ denotes the sum $g_1 \oplus \cdots \oplus g_n$, which does not depend on either order or multiplicity since $h^\sharp$ is bisimulation invariant, and $\phi_{a,\text{prime}}$ to be

$$\Big(\neg \bigvee_{p \notin P_a} \mathbf{EF}\ \phi_p\Big) \wedge \Big(\bigwedge_{f \in F, f \oplus a = \bot} \neg \mathbf{EF}\ \phi_f\Big) \wedge \Big(\mathbf{AG} \bigwedge_{c \notin C} (c \to \mathbf{AG}\ \phi_F)\Big),$$

where $C \subseteq \Sigma_1$ is the set of labels c such that $h_1^\sharp(c) \cdot a = a$.

The argument that a regular forest t satisfies the formula ϕ_a precisely if $h_0^\sharp(t) = a$ remains the same as in (Bojańczyk and Idziaszek, 2009). The proof of case 2 is then finished by noting that both $(h_0^\sharp)^{-1}(a)$ and $\{t \in \mathbb{F}_0\Sigma \mid t \text{ satisfies } \phi_a\}$ are regular languages (considering their context sorts to be empty); the first one by regularity of (A, α), the second one since EF is a fragment of the logic MSO; since these languages contain the same regular forests, they are actually identical.

(3.) For case 3, suppose that the element a is neither minimal nor subminimal, or that it is one of at least two subminimal elements. Let $M \subseteq A_0$ be the set of elements from which a is not reachable and define an equivalence relation $\approx$ on A by setting $b_1 \approx_0 b_2$ if either $b_1 = b_2$ or if both b_1 and b_2 belong to M.

Note that $\approx_0$ is an equivalence relation which in either of the two cases identifies at least two elements of A_0; $\bot$ and the subminimal one "between" a and $\bot$ if a is neither minimal nor subminimal and $\bot$ and the other subminimal elements if a is subminimal but not unique. On the other hand, the element a is only identified with itself. Thus it suffices to extend $\approx_0$ to a congruence $\approx$ on A; then we can form the quotient $(A, \alpha)/\approx$ and apply the induction hypothesis to $h^\sharp$ postcomposed with the quotient homomorphism and the equivalence class of a.

We say that an element $c \in A_1$ is reachable from $b \in A_0$ if c is of the form $\alpha_1(t)$ for some context $t \in \mathbb{F}_1 A$ which contains a leaf labelled by b. Define the set $M' \subseteq A_1$ to contain those elements $c \in A_1$ which are reachable from some element $b \in M$ and put $c_1 \approx_1 c_2$ if either $c_1 = c_2$ or both c_1 and c_2 belong to M'.

We have to verify that $\approx$ is a congruence. Therefore, suppose we have $s \approx_0^{\mathbb{F}} t$ for forests $s, t \in \mathbb{F}_0 A$. If $s = t$, the statement is trivial. Otherwise, suppose there is a leaf v such that $s(v) \neq t(v)$. Then a cannot be reached from either $s(v)$ or $t(v)$. Hence, a cannot be reachable from $\alpha_0(s)$ or $\alpha_0(t)$ either. Finally, suppose there is an internal vertex v such that $s(v) \neq t(v)$. Then $s(v)$ and $t(v)$ are reachable from an element of M, i.e. there is a context $s' \in \mathbb{F}_1 A$ whose product is $s(v)$ and has a leaf labelled by an element of M and an analogous context $t' \in \mathbb{F}_1 A$ for $t(v)$. Let σ be the forest obtained from $\mathbb{F}_0(\mathrm{sing}_A)(s)$ by relabelling v by s'. Then we have $\alpha_0(s) = \alpha_0(\mathbb{F}_0(\alpha)(\sigma)) = \alpha_0(\mathrm{flat}_{A,0}(\sigma))$. This value is reachable from an element of M and therefore contained in M. Similarly, we have $\alpha_0(t) \in M$. Hence, $\alpha_0(s) \approx_0 \alpha_0(t)$.

We also need to show that $\approx$ is respected by α_1. Suppose we have $s \approx_1^{\mathbb{F}} t$. Again, if $s = t$, the statement is trivial. If there is a leaf v such that $s(v)$ and $t(v)$ belong to M, then $\alpha_1(s)$ and $\alpha_1(t)$ belong to M' by definition. Finally, if there is an internal vertex v such that $s(v)$ and $t(v)$ belong to M', we can replace them by a suitable forest containing an element of M. Thus, their products again belong to M'. □

Theorem 4.4. *A regular pure forest language $L \subseteq \mathbb{F}\Sigma$ is definable by an EF formula if and only if it is closed under EF bisimilarity and its syntactic algebra satisfies equation 4.1.*

Proof. ($\Leftarrow$) If L has the required properties, we can apply the previous lemma to its syntactic algebra and homomorphism, the invariance of which follows easily from closure of L under bisimilarity.

($\Rightarrow$) Conversely, a language specified by a formula ϕ is closed under bisimilarity. To show that $\mathrm{Synt}(L)$ satisfies equation 4.1, we proceed as in (Bojańczyk and Idziaszek, 2009). Namely, let $m \in \mathbb{N}$ be such that v^m is idempotent for every $v \in \mathrm{Synt}(L)_1$ (such an m exists since $\mathrm{Synt}(L)$ is finite) and let $m \cdot n \in \mathbb{N}$ be a multiple of m greater than the EF-nesting depth of the formula defining L. We will show that the equation

$$v^{m \cdot n} \cdot a = (v \oplus (v^{m \cdot n} \cdot a))^{\omega}$$

holds for all v, a. By definition, we have to show that for every context $c \in \mathbb{F}_1\Sigma$, every forest $t \in \mathbb{F}_0\Sigma$ and every forest polynomial Φ of sort i, the forest $[\Phi](c^{m \cdot n} \cdot t)$ is in L_i if and only if $[\Phi]((c \oplus (c^{m \cdot n} \cdot t))^{\omega})$ is.

To do this, it is sufficient to show that the two forests are $(m \cdot n)$-bisimilar, since then the formula ϕ cannot distinguish them. It is easily seen that $(m \cdot n)$-bisimilarity is a congruence. Hence it is actually sufficient to note that $c^{m \cdot n} \cdot t$ and $(c \oplus (c^{m \cdot n} \cdot t))^{\omega}$ are $(m \cdot n)$-bisimilar, which is readily seen. □

The same result, however, does not hold for languages containing contexts. Intuitively, in a context, the path to the hole becomes distinguishable. We discuss a counterexample next.

Lemma 4.5. *Let $s, t \in \mathbb{F}_1\Sigma$ be two EF-bisimilar contexts. Then the paths to the hole in s and t have the same length and are marked by the same labels in the same order.*

Proof. We will first show that the path to the hole in t is at least as long as in s. Let $v_1, \dots, v_n$ be the path in s. Then t has a subtree $t|w_1$ bisimilar to $s|v_1$. Since $s|v_1$ contains the hole of s, $t|w_1$ must necessarily contain the hole of t. Similarly, the tree $t|w_1$ must contain a proper subtree $t|w_2$ bisimilar to $s|v_2$ and this $t|w_2$ again must contain the hole vertex. Continuing this way, we obtain n distinct vertices $w_1, \dots, w_n$ on the path to the hole in t.

Since bisimilarity is symmetric, it follows that the paths have the same length. To see that they have the same labels in the same order, note that if $v_1, \dots, v_n$ is again the path to the hole in s, the subtree t' of t bisimilar to $s|v_k$ must be rooted at the vertex at depth k, otherwise $s|v_k$ and t' would differ in the lengths of their paths to the hole. □

Consider now the alphabet A with $A_0 = A_1 = \{a\}$ and the language $L \subseteq \mathbb{F}A$ such that L_0 is empty and L_1 contains those contexts t such the length of the path to the hole is odd. This language is regular; an automaton recognizing L can be constructed by taking Q to be the following semigroup on the set $\{\mathrm{e}, \mathrm{o}, \mathrm{p}, \bot\}$ (representing "even", "odd", "plain" and "error"):

+	e	o	p	⊥
e	⊥	⊥	e	⊥
o	⊥	⊥	o	⊥
p	e	o	p	⊥
⊥	⊥	⊥	⊥	⊥

We add the transitions $(\mathrm{e}, \square)$ and (p, a) to Δ_0 and $(\mathrm{e}, a, \mathrm{o})$, $(\mathrm{o}, a, \mathrm{e})$ and $(\mathrm{p}, a, \mathrm{p})$ to Δ_1, with the set of accepting states $I = \{\mathrm{o}\}$ and the priority mapping $\Omega(\bot) = \Omega(\mathrm{p}) = 0$ and $\Omega(\mathrm{o}) = \Omega(\mathrm{e}) = 1$.

By the above lemma, the language L is invariant under bisimilarity of contexts. Its syntactic algebra also satisfies Equation 4.1, since *all* forests $s, t \in \mathbb{F}_0 A$ are language-equivalent; replacing s by t within a forest-sorted polynomial does not change anything, since L_0 is empty, and replacing s by t in a context-sorted polynomial does not change the length of the path in the resulting context.

However, the set $L_1 \subseteq \mathbb{F}_1 A$ cannot be specified by a formula of EF, as the following lemma shows.

Lemma 4.6. *For every context formula ϕ over A, there are either finitely or cofinitely many natural numbers $n \geq 1$ such that the context $a^n\square$ satisfies ϕ.*

Proof. By induction on ϕ. The only interesting case is when ϕ is of the form **EF** ψ for some formula ψ. If some context $a^n\square$ satisfies ψ, then $a^{n+k}\square$ satisfies ϕ for every $k \geq 1$. If no context $a^n\square$ satisfies ψ, we need to consider if $\square$ (a plain forest over $A^\square$) satisfies ψ. If it does, then every context $a^n\square$ satisfies ϕ; if not, then no such context can satisfy ϕ. $\square$

It would be desirable to have a characterisation similar to Theorem 4.4 for arbitrary regular languages, as the restriction to pure forest languages is unnatural in our setting. There seem to be several directions in which future work could proceed.

First, we could search for additional equations to impose on the syntactic algebra. In general, this might be a part of discovering an equational characterisation of bisimilarity invariance of algebras – if, as suggested by Bojańczyk and Idziaszek (2009), we define an *algebra* to be invariant if every language it recognises is invariant, then the class of invariant algebras is easily seen to be a pseudovariety.

However, as Lemma 4.5 shows, contexts and bisimilarity might be somewhat "incompatible" notions. Another solution might be to enrich our algebra, for example by allowing contexts in which the hole might occur multiple times. We intend to explore this approach in the future as well.

5 Conclusion

We have introduced a basic algebraic framework for recognisability of infinite forests. We have defined the basic contepts dealing with infinite forests, we have described the monadic structure on the constructed functor and we have given a presentation of the intended algebraic structures as Eilenberg-Moore algebras for the monad.

We have specified the class of algebras which as recognisers exactly correspond to regular languages and we have shown that for regular languages, regular syntactic algebras exist.

We have also given a characterisation of definability of regular languages in the logic EF in the special case of pure forest languages. This result, however, does not carry over for general regular languages, as shown by a simple counterexample we have described.

Potential directions of future work include finding equations characterising the pseudovariety of regular algebras and invariance of algebras under EF-bisimilarity. The definition of ω-forest algebras might also be enriched to lend a more well-behaved notion of context bisimilarity and subsequently better characterisations.

A Technical results

A.1 Formal definition of the flattening map

Let S be a biset and suppose we have a forest $\sigma \in \mathbb{F}_0\mathbb{F}S$. Our aim is to formally define the forest $\text{flat}_{S,0}(\sigma)$ and to prove that the maps obtained this way form a natural transformation satisfying the laws of the multiplication map of a monad.

To construct the forest $\text{flat}_{S,0}(\sigma)$, we may observe that the vertices of $\text{flat}_{S,0}(\sigma)$ should be in direct correspondence with those of the individual forests $\sigma(w)$, with the exception of the holes of contexts, which should be replaced by the roots of the forests following in σ. Therefore, we will proceed by constructing a bijection between the domain of $\text{flat}_{S,0}(\sigma)$ and the disjoint union of the domains of the forests $\sigma(w)$, minus the holes. We will use this bijection to transfer the labels and prove the required properties.

First, given a forest s over some biset T, define $\text{Size}(s) := |\text{Root}(s)|$, i.e. the number of trees in s. Also, given a context $s \in \mathbb{F}_1 T$, denote by $\text{PreHole}(s)$ the vertex $w \in \text{Dom}(s)$ directly preceeding $\text{Hole}(s)$.

Now, given the forest σ, we will introduce some notation. For a vertex $w \in \text{Dom}(\sigma)$, we will define the set $\text{Dom}(\sigma, w)$ as follows.

$$\text{Dom}(\sigma, w) := \begin{cases} \text{Dom}(\sigma(w)) & \text{if } w \in \text{Leaf}(\sigma) \\ \text{Dom}(\sigma(w)) - \{\text{Hole}(\sigma(w))\} & \text{if } w \in \text{Int}(\sigma) \end{cases}$$

We will construct an injection $h\colon \coprod_{w \in \text{Dom}(\sigma)} \text{Dom}(\sigma, w) \to \mathbb{N}^+$ and prove that the range of h is a forest domain; the intended bijection will then be obtained simply by restricting the codomain.

How should this map h behave? As noted before, the vertices of a forest $\sigma(w\langle n\rangle)$ are appended to the vertex preceeding the hole of $\sigma(w)$; this suggests that $h(w, v)$ should be defined by recursion on w. However, if a vertex w of σ has a nonzero number of siblings on the left, the vertices of $\sigma(w)$ also have to be shifted in the first element by an appropriate number. Likewise, if $\sigma(w)$ is a context, the left siblings of the hole vertex introduce additional shifts to the roots of the appended forests, while right siblings in general have to be shifted themselves.

To better keep track of the individual shiftings, we will actually define h by double recursion. We distinguish three cases. For roots of forests labelling root vertices of σ, the values for shifts are straightforward; we simply set

$$h(\langle m\rangle, \langle n\rangle) := \langle n+s\rangle,$$

where $s := \sum_{k<m} \mathrm{Size}(\sigma(\langle k\rangle))$.

Suppose we already know where a parent of a vertex v inside a forest $\sigma(w)$ maps to. If v is not a right sibling of the hole vertex, its value stays the same. If it is, it has to be shifted by the total number of trees that should be in the place of the hole minus one (since the hole itself disappears). Therefore, we set

$$h(w, v^\frown\langle n\rangle) := \begin{cases} h(w,v)^\frown\langle n\rangle & \text{if } v\langle n\rangle \text{ is not a right sibling of} \\ & \mathrm{Hole}(\sigma(w)) \\ h(w,v)^\frown\langle n-1+s\rangle & \text{otherwise,} \end{cases}$$

where $s := \sum_{w^\frown\langle k\rangle\in \mathrm{Dom}(\sigma)} \mathrm{Size}(\sigma(w\langle k\rangle))$ (recall that $w^\frown v$ denotes the concatenation of w and v).

It remains to calculate the shift for roots that should replace a hole in a forest above. For this, we define

$$h(w\langle m\rangle, \langle n\rangle) := h(w, \mathrm{PreHole}(\sigma(w)))^\frown\langle n+k+s\rangle,$$

where $\mathrm{PreHole}(\sigma(w)) = v\langle k\rangle$ for some v and $s := \sum_{\ell<m} \mathrm{Size}(\sigma(w\langle\ell\rangle)$.

Using these definitions, it is now straightforward to prove by double induction that h is injective and its image is a valid forest domain. Considering now h as a bijection $h_\sigma \colon \coprod_{w\in\mathrm{Dom}(\sigma)} \mathrm{Dom}(\sigma, w) \to \mathrm{Im}(h)$, where the subscript σ is a reminder of the dependence of h on the actual forest, we can define the forest $\mathrm{flat}_{S,0}(\sigma) \in \mathbb{F}_0 S$ as $(\mathrm{Im}(h_\sigma), \ell_0, \ell_1)$, where both labellings can be defined by the same formula,

$$\ell_i(w) := \sigma(\pi_1(h_\sigma^{-1}(w)))(\pi_2(h_\sigma^{-1}(w))).$$

We also need to define the map $\mathrm{flat}_{S,1} \colon \mathbb{F}_1\mathbb{F}S \to \mathbb{F}_1 S$. This can be easily reduced to the nullary case: let $\iota\colon S \to S^\square$ be the obvious inclusion map. We will write $\circ \in S^\square_0$ for the new label added to S to distinguish it from the hole $\square \in (S^\square)^\square_0$. For a context $\sigma \in \mathbb{F}_1\mathbb{F}S$, denote by

$\sigma' \in \mathbb{F}_0\mathbb{F}(S^{\square})$ the forest obtained from $\mathbb{F}_1(\mathbb{F}(\iota))(\sigma)$ by relabelling the hole vertex with the forest $\mathrm{sing}_{S^{\square},0}(\bigcirc)$. The forest $\mathrm{flat}_{S^{\square},0}(\sigma')$ is then a valid context over S and can be defined as the result of $\mathrm{flat}_{S,1}(\sigma)$.

It is easily verified that the family $(\mathrm{flat}_{S,0}, \mathrm{flat}_{S,1})_S$ forms a natural transformation $\mathrm{flat}\colon \mathbb{F}^2 \to \mathbb{F}$. Likewise, the triangle identities $\mathrm{flat} \circ (\mathrm{sing}\, \mathbb{F}) = 1_{\mathbb{F}} = \mathrm{flat} \circ (\mathbb{F}\, \mathrm{sing})$ can be directly established.

It remains to prove associativity, i.e. the commutativity of the following diagram.

$$\begin{array}{ccc} \mathbb{F}^3 & \xrightarrow{\mathrm{flat}\,\mathbb{F}} & \mathbb{F}^2 \\ {\scriptstyle \mathbb{F}\,\mathrm{flat}}\downarrow & & \downarrow{\scriptstyle \mathrm{flat}} \\ \mathbb{F}^2 & \xrightarrow{\mathrm{flat}} & \mathbb{F} \end{array}$$

Suppose we have a forest $T \in \mathbb{F}_0\mathbb{F}\mathbb{F}S$. Consider the maps

$$f, g\colon \coprod_{w \in \mathrm{Dom}(T)} \coprod_{v \in \mathrm{Dom}(T,w)} \mathrm{Dom}(T(w), v) \to \mathbb{N}^+$$

which we define by setting $f(w, v, u) := h_{\mathrm{flat}_{\mathbb{F}S,0}(T)}(h_T(w, v), u)$ and $g(w, v, u) := h_{\mathbb{F}_0(\mathbb{F}(\mathrm{flat}_S))(T)}(w, h_{T(w)}(v, u))$. It is easily seen that the maps correspond to the two ways of flattening T as follows.

1. The image of f equals the domain of $\mathrm{flat}_{S,0}(\mathrm{flat}_{\mathbb{F}S,0}(T))$ and for every (w, v, u), we have

$$T(w)(v)(u) = \mathrm{flat}_{S,0}(\mathrm{flat}_{\mathbb{F}S,0}(T))(f(w, v, u)),$$

   and similarly,

2. The image of g equals the domain of $\mathrm{flat}_{S,0}(\mathbb{F}_0(\mathrm{flat}_S)(T))$ and for every (w, v, u), we have

$$T(w)(v)(u) = \mathrm{flat}_{S,0}(\mathbb{F}_0(\mathrm{flat}_S)(T))(g(w, v, u)).$$

However, by triple induction, we can show that $f(w, v, u) = g(w, v, u)$, establishing the required equality.

A.2 Definitions of common forest operations

Using the general flattening operation, we can define the particular operations mentioned in Chapter 2.

Given two forests $s, t \in \mathbb{F}_0 S$, we define their horizontal composition $s \oplus t \in \mathbb{F}_0 S$ as $\mathrm{flat}_{S,0}(\{\langle 0\rangle, \langle 1\rangle\}, (\langle 0\rangle \mapsto s, \langle 1\rangle \mapsto t), \emptyset)$. Similarly, given a forest $s \in \mathbb{F}_0 S$ and a context $t \in \mathbb{F}_1 S$, we define $s \oplus t \in \mathbb{F}_1 S$ as

$$\mathrm{flat}_{S,1}(\{\langle 0\rangle, \langle 1\rangle, \langle 10\rangle\}, \ell_0, \ell_1),$$

where the labellings are given by $\ell_0(\langle 0\rangle) := s$, $\ell_0(\langle 10\rangle) := \Box$ and $\ell_1(\langle 1\rangle) := t$, and $t \oplus s$ analogically.

Given a context $s \in \mathbb{F}_1 S$ and a forest $t \in \mathbb{F}_0 S$, their vertical composition $s \cdot t$ can be defined as $\mathrm{flat}_{S,0}(\{\langle 0\rangle, \langle 00\rangle\}, (\langle 00\rangle \mapsto t), (\langle 0\rangle \mapsto s))$. If $t \in \mathbb{F}_1 S$ is a context instead, $s \cdot t$ can be defined as

$$\mathrm{flat}_{S,1}(\{\langle 0\rangle, \langle 00\rangle, \langle 000\rangle\}, \ell_0, \ell_1),$$

where $\ell_0(\langle 000\rangle) := \Box$, $\ell_1(\langle 0\rangle) := s$ and $\ell_1(\langle 00\rangle) := t$.

Finally, given a context $s \in \mathbb{F}_1 S$, we can define the infinite unfolding s^ω as

$$\mathrm{flat}_{S,0}(\{\langle 0\rangle^n \,|\, n \geq 1\}, \emptyset, (w \mapsto s)).$$

The expected identities, such as associativity of $\oplus$, satisfied by these operations can be easily proven using associativity of the transformation flat.

Bibliography

ATKEY, Robert and Conor MCBRIDE, 2003. Productive coprogramming with guarded recursion. In: *ICFP '13*. Proceedings of the 18th ACM SIGPLAN international conference on Functional programming. New York: ACM. 197–208. ISBN 978-1-4503-2326-0.

BLUMENSATH, Achim, 2011. Recognisability for algebras of infinite trees. *Theoretical Computer Science*. Amsterdam: Elsevier. **412**(29), pp. 3436–3486. ISSN 0304-3975.

BLUMENSATH, Achim, 2018a. *Branch-continuous tree algebras* [online]. arXiv:1807.04568 [cit. 5. 12. 2018]. Available at: `http://arxiv.org/abs/1807.04568`.

BLUMENSATH, Achim, 2018b. *Regular tree algebras* [online]. arXiv:1808.03559 [cit. 5. 12. 2018]. Available at: `http://arxiv.org/abs/1808.03559`.

BOJAŃCZYK, Mikołaj and Tomasz IDZIASZEK, 2009. Algebra for Infinite Forests with an Application to the Temporal Logic EF. In: *CONCUR 2009 – Concurrency theory*. CONCUR 2009. Berlin, Heidelberg: Springer, pp. 131–145. ISBN 978-3-642-04081-8.

BOJAŃCZYK, Mikołaj, 2015. Recognisable Languages over Monads. In: *Developments in Language Theory*. 19th International Conference, DLT 2015. Cham: Springer, pp. 1–15. ISBN 978-3-319-21499-3.

BOJAŃCZYK, Mikołaj and Thomas PLACE, 2012. Regular Languages of Infinite Trees that are Boolean Combinations of Open Sets. In: *Automata, Languages, and Programming.* ICALP 2012. Berlin, Heidelberg: Springer, pp. 104–115. ISBN 978-3-642-31585-5.

BORCEUX, Francis, 1994. *Handbook of categorical algebra 2*. Categories and structures. Cambridge: Cambridge University Press. Encyclopedia of mathematics and its applications; vol. 51. ISBN 9780521441797.

CHEN, Liang-Ting, Jiří ADÁMEK, Stefan MILIUS and Henning URBAT, 2016. Profinite Monads, Profinite Equations, and Reiterman's Theorem. In: *Foundations of Software Science and Computation Structures*. FoSSaCS 2016. Berlin, Heidelberg: Springer, pp. 531–547. ISBN 978-3-662-49629-9.

IDZIASZEK, Tomasz, Mikołaj BOJAŃCZYK and Michał SKRZYPCZAK, 2016. Regular languages of thin trees. *Theory of Computing Systems*. New York: Springer, **58**(4), pp. 614–663. ISSN 1433-0490.