ω-Forest Algebras and Temporal Logics

Achim Blumensath[1]
Masaryk University Brno
blumens@fi.muni.cz

Jakub Lédl[2]
Masaryk University Brno
jakubledl@mail.muni.cz

Abstract

We use the algebraic framework for languages of infinite trees introduced in [5] to derive effective characterisations of various temporal logics, in particular, the logic EF (a fragment of CTL) and its counting variant cEF.

2012 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases forest algebras, wreath products, temporal logics

Digital Object Identifier 10.4230/LIPIcs...xxx

1 Introduction

Among the many different approaches to language theory, the algebraic one seems to be particularly convenient when studying questions of expressive power. While algebraic language theories for word languages (both finite and infinite) have already been fully developed a long time ago, the corresponding picture for languages of trees, in particular infinite ones, is much less complete. Seminal results contributing to such an algebraic framework for languages of infinite trees were provided by the group of Bojańczyk [7, 8] with one article considering languages of *regular* trees only, and one considering languages of *thin* trees. The first complete framework that could deal with arbitrary infinite trees was provided in [2, 3]. Unfortunately, it turned out to be too complicated and technical for applications. Recently, two new general frameworks have been introduced [1, 5] which seem to be more satisfactory: one is based on the notion of a *branch-continuous tree algebra,* while the other uses *regular tree algebras.* At the moment it is still unclear which of these two competing approaches is the right one. The first one seems to be more satisfactory from a theoretical point of view, while the second one is more useful for applications, in particular for characterisation results.

In this article we concentrate on the approach based on regular tree algebras from [5] and apply it to a few test cases to see how suitable it is for its intended purpose. While the definition of a regular tree algebra (given in Section 2 below) is a bit naïve and seems circular at first sight, it turns out that it is sufficient to guarantee the properties we need for applications: one can show that (i) the class of regular tree algebras forms a pseudo-variety and that (ii) every regular tree language has a syntactic algebra, which is in fact a regular tree algebra. By general category-theoretic results, such as those from [6] or [4], this implies that there exists a Reiterman type theorem for such algebras, i.e., the existence of equational characterisations for sub-pseudo-varieties. This is precisely what is needed for a characterisation theorem.

[1] Work supported by the Czech Science Foundation, grant No. GA17-01035S
[2] Work supported by the Czech Science Foundation, grant No. GA17-01035S

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

The applications we are looking at in the present paper concern certain temporal logics, in particular, the logic EF and its counting variant cEF, and we aim to derive decidable algebraic characterisations for them using our algebraic framework. Note that Bojańczyk and Idziaszek have already provided a decidable characterisation for EF in [7], but their result is only partially algebraic. They prove that a regular language is definable in EF if, and only if, the language is bisimulation-invariant and its syntactic algebra satisfies a certain equation, but they were not able to provide an algebraic characterisation of bisimulation invariance. Due to our more general algebraic framework we are able to fill this gap below.

We start in the next section with a short overview of the algebraic framework from [5]. We have to slightly modify this material since it was originally formulated in the setting of ranked trees while, when looking at temporal logics, it is more natural to consider unranked trees and forests. The remainder of the article contains our various characterisation results. In Section 3 we derive an algebraic characterisation of bisimulation-invariance, the result missing in [7]. Then we turn to our main result and present characterisations for the logic cEF and some of its fragments, including the logic EF. The result itself and some consequences are presented in Section 4, while the proof is deferred to Section 5.

2 Forest algebras

The main topic of this article are languages of (possibly infinite) forests and the logics defining them. Before introducing the algebras we will use to recognise such languages, let us start by fixing some notation and conventions. Although our main interest is in unranked forests, we will use a more general version that combines the ranked and the unranked cases. As we will see below (cf. Theorem 3.1), the ability to use ranks will increase the expressive power of equations for our algebras considerably. Thus, we will work with *ranked sets,* i.e., sets where every element a is assigned an *arity* $\mathrm{ar}(a)$. Formally, we consider such sets as families $A = (A_m)_{m<\omega}$, where A_m is the set of all elements of A of arity m. Functions between ranked sets then take the form $f = (f_m)_{m<\omega}$ with $f_m : A_m \to B_m$.

We will consider (unranked, finitely branching, possibly infinite) forests where each vertex is labelled by an element of a given ranked set A and each edge is labelled by a natural number with the restriction that, if a vertex is labelled by an element of arity m, the numbers labelling the outgoing edges must be less than m. If an edge $u \to v$ is labelled by the number k, we will call v a *k-successor* of u. Note that a vertex may have several k-successors, or none at all. We assume that all k-successors of a given vertex are ordered from left to right, while we impose no ordering between a k-successor and an l-successor, for $k \neq l$. We write $\mathbb{F}_0 A$ for the set of all such A-labelled forests. (We shall explain the index 0 further below.) We write $\mathrm{dom}(s)$ for the set of vertices of a forest $s \in \mathbb{F}_0 A$, and we will usually identify s with the function $s : \mathrm{dom}(s) \to A$ that maps vertices to their labels. We denote the empty forest by 0 and the disjoint union of two forests s and t by $s + t$. We will frequently use term notation to denote forests such as

$$a(b + c, 0, b) + b\,,$$

which denotes a forest with two components: the first one consisting of a root labelled by an element a of arity 3 which has two 0-successors labelled b and c, no 1-successor, and one 2-successor; the second component consists of a singleton with label b.

We use the symbol $\preceq$ for the forest ordering where the roots are the minimal elements and the leaves the maximal ones. For a forest s, we denote by $s|_v$ the subtree of s attached to the vertex v. The *successor forest* of v in s is the forest obtained from $s|_v$ by removing the root v.

For a natural number n, set $[n] := \{0, \dots, n-1\}$. An *alphabet* is a finite (unranked) set Σ of symbols. If we use an alphabet in a situation such as $\mathbb{F}_0\Sigma$ where a ranked set is expected, we will consider each symbol in Σ as having arity 1. Thus, for us a *forest language over an alphabet* Σ will be a set $L \subseteq \mathbb{F}_0\Sigma$ consisting of the usual unranked forests. (The power to have elements of various arities is useful when writing down algebraic equations, but it is rather unnatural when considering languages defined by temporal logics.) We denote by Σ^* the set of all finite words over Σ, by Σ^ω the set of infinite words, and $\Sigma^\infty := \Sigma^* \cup \Sigma^\omega$. A *family of (word, forest,…) languages* is a function $\mathcal{K}$ mapping each alphabet Σ to a class $\mathcal{K}[\Sigma]$ of (word, forest,…) languages over Σ.

Our algebraic framework to study forest languages is built on the notion of an Eilenberg–Moore algebra for a monad. To keep category-theoretical prerequisites at a minimum we will give an elementary, self-contained definition. The basic idea is that, in the same way we can view the product of a semigroup as an operation turning a sequence of semigroup elements into a single element, we view the product of a forest algebra as an operation turning a given forest that is labelled with elements of the algebra into a single element. The material in this section is taken from [5] with minor adaptions to accommodate the fact that we are dealing with unranked forests instead of ranked trees. We start by defining which forest we allow in this process.

▶ **Definition 2.1.** (a) We denote by $\mathbb{F}$ the functor mapping a ranked set A to the ranked set $\mathbb{F}A = (\mathbb{F}_m A)_m$ where $\mathbb{F}_m A$ consists of all $(A \cup \{x_0, \dots, x_{m-1}\})$-labelled forests such that

- the new labels $x_0, \dots, x_{m-1}$ have arity 0,
- each label x_i appears only finitely many times, and
- no root is labelled by an x_i.

(b) The *singleton function* $\mathrm{sing} : A \to \mathbb{F}A$ maps a label a of arity m to the forest $a(x_0, \dots, x_{m-1})$.

(c) The *flattening function* $\mathrm{flat} : \mathbb{F}\mathbb{F}A \to \mathbb{F}A$ takes a forest $s \in \mathbb{F}\mathbb{F}A$ and maps it to the forest $\mathrm{flat}(s)$ obtained by assembling all forests $s(v)$, for $v \in \mathrm{dom}(s)$, into a single large forest. This is done as follows. For every vertex of $s(v)$ that is labelled by a variable x_k, we take the disjoint union of all forests labelling the k-successors of v and substitute them for x_k. This is done simultaneously for all $v \in \mathrm{dom}(s)$ and all variables in $s(v)$ (see Figure 1 for an example.) ⌟

Now we can define a forest algebra to be a set A equipped with a product $\mathbb{F}A \to A$.

▶ **Definition 2.2.** (a) An *ω-forest algebra* $\mathfrak{A} = \langle A, \pi \rangle$ consists of a ranked set A and a function $\pi : \mathbb{F}A \to A$ satisfying the following two axioms:

$$\text{the \textit{associative law}} \quad \pi \circ \mathbb{F}\pi = \pi \circ \mathrm{flat} \qquad \text{and} \qquad \text{the \textit{unit law}} \quad \pi \circ \mathrm{sing} = \mathrm{id}\,.$$

We will denote forest algebras by fraktur letters $\mathfrak{A}$ and their universes by the corresponding roman letter A. We will usually use the letter π for the product, even if several algebras are involved.

(b) A *morphism* of ω-forest algebras is a function $\varphi : \mathfrak{A} \to \mathfrak{B}$ that commutes with the products in the sense that $\pi \circ \mathbb{F}\varphi = \varphi \circ \pi$. ⌟

▶ Remark. (a) In the following we will simplify terminology by dropping the ω and simply speaking of *forest algebras.* But note that, strictly speaking, this name belongs to the kind of algebras introduced by Bojańczyk and Walukiewicz in [10].

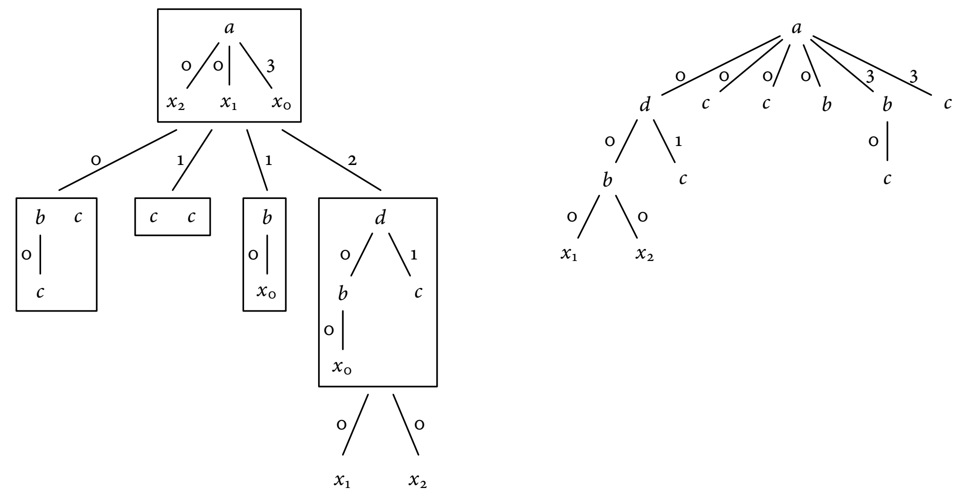

Figure 1 The flattening operation

(b) One can show that the functor $\mathbb{F}$ together with the two natural transformations flat and sing forms what is called a *monad* in category theory. In this terminology, we can define forest algebras as *Eilenberg-Moore algebras* for this monad.

(c) Note that a forest algebra $\mathfrak{A} = \langle A, \pi \rangle$ contains a monoid $\langle A_0, +, 0 \rangle$ and an ω-semigroup $\langle A_1, A_0, \cdot \rangle$. We call the former the *horizontal monoid* and the latter the *vertical ω-semigroup.*

Sets of the form $\mathbb{F}A$ can be equipped with a canonical forest algebra structure by using the flattening operation flat : $\mathbb{F}\mathbb{F}A \to \mathbb{F}A$ for the product. By general category-theoretical considerations it follows that algebras of this form are exactly the *free* forest algebras (generated by A). In this article we consider *forest languages* over an alphabet Σ as subsets $L \subseteq \mathbb{F}_0\Sigma$. Such a language is *recognised* by a morphism $\eta : \mathbb{F}\Sigma \to \mathfrak{A}$ of forest algebras if $L = \eta^{-1}[P]$ for some $P \subseteq A_0$. In analogy to the situation with word languages we would like to have a theorem stating that a forest language is regular if, and only if, it is recognised by a morphism into some finite forest algebra. But this statement is wrong for two reasons. The first one is that every forest algebras with at least one element of positive arity has elements of every arity and, thus, is infinite. To fix this, we have to replace the property of being finite by that of having only finitely many elements of each arity. We call such algebras *finitary.*

But even if we modify the statement in this way it still fails since one can find finitary forest algebras recognising non-regular languages. (An example for tree languages is given by Bojańczyk and Klin in [9].) Therefore we have to restrict our class of algebras. A simple way to do so is given by the class of (locally) regular algebras introduced in [5] where all of the following results are taken from (again in the case of trees instead of forests).

▶ **Definition 2.3.** Let $\mathfrak{A}$ be a forest algebra.

(a) A subset $C \subseteq A$ is *regularly embedded* if, for every element $a \in A$, the preimage $\pi^{-1}(a) \cap \mathbb{F}C$ is forms a regular (i.e., automaton recognisable) language over C.

(b) $\mathfrak{A}$ is *locally regular* if every finite subset is regularly embedded.

(c) $\mathfrak{A}$ is *regular* if it is finitary, finitely generated, and locally regular. ⌟

The definition of a regular forest algebra is not very enlightening. We refer the interested reader to [5] for a purely algebraic (but much more complicated) characterisations.

▶ **Theorem 2.4.** *Let $L \subseteq \mathbb{F}_0\Sigma$ be a forest language. The following statements are equivalent.*

(1) *L is regular (i.e., automaton recognisable).*
(2) *L is recognised by a morphism into a locally regular forest algebra.*
(3) *L is recognised by a morphism into a regular forest algebra.*

(The reason why we introduce two classes is that locally regular algebras enjoy better closure properties, while the regular ones are more natural as recognisers of languages.) One can show (see [5]) that the (locally) regular algebras form a pseudo-variety in the sense that locally regular algebras are closed under quotients, subalgebras, finite products, and directed colimits, while regular algebras are closed under quotients, finitely generated subalgebras, finitely generated subalgebras of finite products, and so-called 'rank-limits'. More important for our current purposes is the existence of syntactic algebras and the fact that these are always regular.

▶ **Definition 2.5.** Let $L \subseteq \mathbb{F}\Sigma$ be a forest language.
(a) The *syntactic congruence* of L is the relation

$$s \sim_L t \quad :\text{iff} \quad p[s] \in L \Leftrightarrow p[t] \in L\,, \quad \text{for every context } p\,,$$

where a context is a $(\Sigma \cup \{\Box\})$-labelled forest and $p[s]$ is the forest obtained from p by replacing each vertex labelled by $\Box$ by the forest s.
(b) The *syntactic algebra* of L is the quotient $\mathfrak{S}(L) := \mathbb{F}\Sigma/{\sim_L}$. ⌟

▶ **Theorem 2.6.** *The syntactic algebra $\mathfrak{S}(L)$ of a regular forest language L exists, it is regular, and it is the smallest forest algebra recognising L. Furthermore, $\mathfrak{S}(L)$ can be computed given an automaton for L.*

Regarding the last statement of this theorem, we should explain what we mean by computing a forest algebra. Since forest algebras have infinitely many elements, we cannot simply compute the full multiplication table. Instead, we say that a regular forest algebra $\mathfrak{A}$ is computable if, given a number $n < \omega$, we can compute a list $\langle \mathcal{A}_a \rangle_{a \in A_n}$ of automata such that $\mathcal{A}_a$ recognises the set $\pi^{-1}(a) \cap \mathbb{F}C$, for some fixed set C of generators.

3 Bisimulation

To illustrate the use of syntactic algebras let us start with a simple warm-up exercise: we derive an algebraic characterisation of bisimulation invariance. This example also explains why algebras with elements of higher arities are needed (this is the reason Bojańczyk and Idziaszek [7], whose framework supported only arity 1, had to leave such a characterisation as an open problem).

Recall that a *bisimulation* between two forests s and t is a binary relation $Z \subseteq \mathrm{dom}(s) \times \mathrm{dom}(t)$ such that $\langle u, v \rangle \in Z$ implies that

- $s(u) = t(v)$ and,
- for every k-successor u' of u, there is some k-successor v' of v with $\langle u', v' \rangle \in Z$ and vice versa.

Two trees are *bisimilar* if there exists a bisimulation between them that relates their roots. More generally, two forests are bisimilar if every component of one is bisimilar to some component of the other.

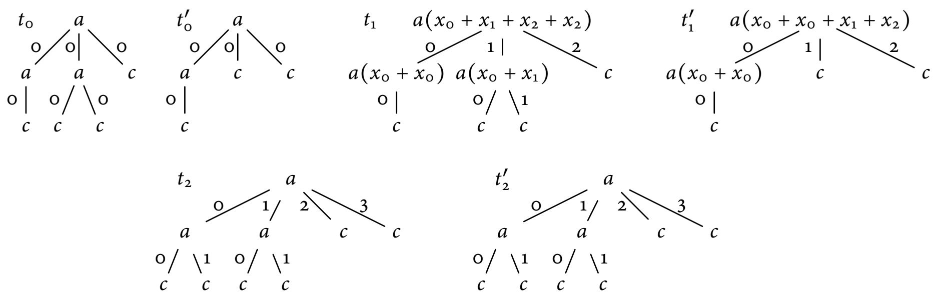

Figure 2 Transforming bisimilar forests

▶ **Theorem 3.1.** *A forest language $L \subseteq \mathbb{F}_0\Sigma$ is bisimulation-invariant if, and only if, the syntactic algebra $\mathfrak{S}(L)$ satisfies the following equations:*

$$c + c = c\,, \qquad a(x_0 + x_0) = a(x_0)\,,$$
$$c + d = d + c\,, \qquad a(x_0 + x_1 + x_2 + x_3) = a(x_0 + x_2 + x_1 + x_3)\,,$$

for all $a \in S_1$ and $c, d \in S_0$.

Proof. Let $\eta : \mathbb{F}\Sigma \to \mathfrak{S}(L)$ be the syntactic morphism mapping a forest to its $\sim_L$-class.

($\Rightarrow$) Given elements $c, d \in S_1$, we fix forests $s \in \eta^{-1}(c)$ and $t \in \eta^{-1}(d)$. If L is bisimulation-invariant, we have

$$p[s] \in L \quad \text{iff} \quad p[s+s] \in L \qquad \text{and} \qquad p[s+t] \in L \quad \text{iff} \quad p[t+s] \in L\,,$$

for every context p. Consequently, $s \sim_L s + s$ and $s + t \sim_L t + s$, which implies that $c = c + c$ and $c + d = d + c$.

The remaining two equations are proved similarly. Fix $a \in S_1$ and $s \in \eta^{-1}(a)$. Setting $s' := s(x_0 + x_0)$, bisimulation-invariance of L implies that

$$p[s] \in L \quad \text{iff} \quad p[s'] \in L\,, \quad \text{for every context } p\,.$$

Consequently $s \sim_L s'$ and $a(x_0) = \eta(s) = \eta(s') = a(x_0 + x_0)$.

Similarly, for $t := s(x_0 + x_1 + x_2 + x_3)$ and $t' := s(x_0 + x_2 + x_1 + x_3)$, we have

$$p[t] \in L \quad \text{iff} \quad p[t'] \in L\,, \quad \text{for every context } p\,.$$

Hence, $t \sim_L t'$ and $a(x_0 + x_1 + x_2 + x_3) = a(x_0 + x_2 + x_1 + x_3)$.

($\Leftarrow$) Suppose that $\mathfrak{S}(L)$ satisfies the four equations above and let s and s' be bisimilar forests. We claim that $\eta(s) = \eta(s')$, which implies that $s \in L \Leftrightarrow s' \in L$.

Fix a bisimulation relation $Z \subseteq \mathrm{dom}(s) \times \mathrm{dom}(s')$. W.l.o.g. we may assume that Z only relates vertices on the same level of the respective forests and that it only relates vertices whose predecessors are also related. (If not, we can always remove the pairs not satisfying this condition without destroying the fact that Z is a bisimulation.) Let $\approx$ be the equivalence relation on $\mathrm{dom}(s) \cup \mathrm{dom}(s')$ generated by Z.

We will transform the forests s and s' in several steps while preserving their value under η until both forests are equal. (Note that each of these steps necessarily modifies the given forest at every vertex.) An example of this process can be found in Figure 2. The first step consists in translating the problem into the algebra $\mathfrak{S}$. We define two new forests $t_0, t'_0 \in \mathbb{F}_0 S$

with the same domains as, respectively, s and s' and the following labelling. If $v \in \mathrm{dom}(s)$ has the 0-successors $u_0, \dots, u_{n-1}$, we set

$$t_0(v) := \eta(s(v))(x_0 + \cdots + x_{n-1})$$

and we make u_i an i-successor of v in t_0. We obtain t'_0 from s' in the same way. By associativity it follows that $\pi(t_0) = \eta(s)$ and $\pi(t'_0) = \eta(s')$.

Next we make the shapes of the forests t_0 and t'_0 the same. Let t_1 and t'_1 be the forests with the same domains as t_0 and t'_0 and the following labelling. For every vertex v of t_0 with successors $u_0, \dots, u_{n-1}$ and labelling

$$t_0(v) = a(x_0 + \cdots + x_{n-1}),$$

we set

$$t_1(v) := a(x_0 + \cdots + x_0 + \cdots + x_{n-1} + \cdots + x_{n-1}),$$

where each variable x_i is repeated m_i times and the numbers m_i are determined as follows. Let M be some number such that, for every $i < n$, no vertex $v' \approx v$ has at more than M successors u' with $u' \approx u_i$. (Note that there are only finitely many such vertices.) We choose the constants m_i such that

$$\sum_{k \in U_i} m_k = M, \quad \text{where} \quad U_i := \{\, k < n \mid u_k \approx u_i \,\}.$$

We obtain the forest t'_1 in the same way from t'_0. By the top right equation above, the value of the product is not affected by this modification. Hence, $\pi(t_1) = \pi(t_0)$ and $\pi(t'_1) = \pi(t'_0)$.

Finally, let t_2 and t'_2 be the unravelling of, respectively, t_1 and t'_1, i.e., the forest where for every vertex v with successors $u_0, \dots, u_{n-1}$ and label

$$t_1(v) = a(x_0 + \cdots + x_0 + \cdots + x_{n-1} + \cdots + x_{n-1}),$$

we set

$$t_2(v) := a(x_0 + \cdots + x_k + \cdots + x_l + \cdots + x_m)$$

(where we number the variables from left-to-right, e.g., $a(x_0 + x_0 + x_1 + x_2 + x_2)$ becomes $a(x_0 + x_1 + x_2 + x_3 + x_4)$), and we duplicate each attached subforest a corresponding number of times such that the value of the product does not change. We do the same for t'_2.

We have arrived at a situation where, for each component r of the forests t_2, there is some component r' of t'_2 that differs only in the ordering of successors, but not in their number. Consequently, there exists a bijection $\sigma : \mathrm{dom}(t) \to \mathrm{dom}(r')$ such that, for a vertex v of r with successors $u_0, \dots, u_{n-1}$,

$$r'(v) = r(v)(x_{\sigma_v(0)} + \cdots + x_{\sigma_v(n-1)}),$$

where the function $\sigma_v : [n] \to [n]$ is chosen such that $\sigma(u_i)$ is the $\sigma_v(i)$-successor of $\sigma(v)$.

Let $\hat{r}$ be the tree obtained from r as follows. For a vertex v with successors $u_0, \dots, u_{n-1}$ and labelling

$$r(v) = a(x_0 + \cdots + x_{n-1}),$$

we set

$$\hat{r}(v) := a(x_{\sigma_v(0)} + \cdots + x_{\sigma_v(n-1)}),$$

and we reorder the attached subtrees accordingly. By associativity and the bottom right equation, this does not change the value of the product. It follows that $\hat{r} = r'$. Consequently, $\pi(r) = \pi(r')$.

We have shown that, for every component of t_0 there is some component of t'_0 with the same product. Therefore, we can write

$$\pi(t_0) = a_0 + \cdots + a_{m-1} \quad \text{and} \quad \pi(t'_0) = b_0 + \cdots + b_{n-1}$$

where the sets $\{a_0, \dots, a_{m-1}\}$ and $\{b_0, \dots, b_{m-1}\}$ coincide. Using the equations $c + c = c$ and $c + d = d + c$ we can therefore transform $\pi(t_0)$ into $\pi(t'_0)$. Consequently,

$$\eta(s) = \pi(t_0) = \pi(t'_0) = \eta(s')\,.$$

As η recognises L it follows that $s \in L \Leftrightarrow s' \in L$, as desired. ◀

Note that we immediately obtain a decision procedure for bisimulation-invariance from this theorem, since we can compute the syntactic algebra and check whether it satisfies the given set of equations.

▶ **Corollary 3.2.** *It is decidable whether a given regular language L is bisimulation-invariant.*

4 The Logic cEF

Let us now proceed to the main result of this article: a characterisation of the temporal logic cEF. For simplicity, the following definition of its semantics only considers forests instead of arbitrary transition systems.

▶ **Definition 4.1.** (a) *Counting* EF*,* cEF for short, has two kinds of formulae: *tree formulae* and *forest formulae,* which are inductively defined as follows.

- Every forest formula is a finite boolean combination of formulae of the form $\mathsf{E}_k\varphi$ where k is a positive integer and φ a tree formula.
- Every tree formula is a finite boolean combination of (i) forest formulae and (ii) formulae of the form P_a, for $a \in \Sigma$.

To define the semantics we introduce a satisfaction relation $\models_{\mathrm{f}}$ for forest formulae and one $\models_{\mathrm{t}}$ for tree formulae. In both cases boolean combinations are defined in the usual way. For a tree t, we define

$$\begin{aligned} t \models_{\mathrm{t}} P_a \quad &:\text{iff} \quad \text{the root of } t \text{ has label } a\,, \\ t \models_{\mathrm{t}} \varphi \quad &:\text{iff} \quad t' \models_{\mathrm{f}} \varphi\,, \quad \text{for a forest formula } \varphi\,, \text{ where } t' \text{ denotes the successor} \\ &\qquad\qquad\qquad\qquad \text{forest of the root of } t\,. \end{aligned}$$

For a forest s, we define

$$s \models_{\mathrm{f}} \mathsf{E}_k\varphi \quad :\text{iff} \quad \text{there exist at least } n \text{ vertices } v, \text{ distinct from the roots, such that } s|_v \models \varphi\,.$$

(b) For $k, m < \omega$, we denote by cEF_k the fragment of cEF that uses only operators E_l where $l \leq k$, and cEF_k^m is the fragment of cEF_k where the nesting depth of the operators E_l is restricted to m. For $k = 1$, we set $\mathrm{EF} := \mathrm{cEF}_1$ and $\mathrm{EF}^m := \mathrm{cEF}_1^m$. ⌟

The following is our main theorem. Before giving the statement a few technical remarks are in order. In the equations below we make use of the *ω-power* a^ω of an element $a \in A_1$ (which is the infinite vertical product $aaa\dots$), and the *idempotent power* a^π (which is the defined as $a^\pi = a^n$ for the minimal number n with $a^n a^n = a^n$). For the horizontal semigroup we use multiplicative notation instead: $n \times a$ for $a + \dots + a$ and $\pi \times a$ for $n \times a$ with n as above.

When writing an ω-power of an element of arity greater than one, we need to specify with respect to which variable we take the power. We use the notation a^{ω_i} to indicate that the variable x_i should be used. Note that, when using several ω-powers like in $(a(x_0, (b(x_0, x_1))^{\omega_1}))^{\omega_0}$, the intermediate term after resolving the inner power can be a forest with infinitely many occurrences of the variable x_0. But after resolving the outer ω-power, we obtain a forest without variables, i.e., a proper element of $\mathbb{F}_0 A$. Consequently, the equations below are all well-defined. Finally, to keep notation light we will frequently write x instead of x_0, if this is the only variable present.

▶ **Theorem 4.2.** *A forest language $L \subseteq \mathbb{F}_0 \Sigma$ is definable in the logic* cEF$_k$ *if, and only if, the syntactic algebra $\mathfrak{S}(L)$ satisfies the following equations:*

$$
\begin{aligned}
c + d &= d + c & (a(x) + b(x))^\omega &= (ab(x))^\omega \\
(ab)^\pi &= b(ab)^\pi & (a(x) + c)^\omega &= (a(x + c))^\omega \\
a^\omega + a^\omega &= a^\omega & (a(x + c + c))^\omega &= (a(x + c))^\omega \\
(abb')^\omega &= (ab'b)^\omega & \big[a(b(x_0, x_1))^{\omega_1}\big]^{\omega_0} &= [ab(x_0, x_0)]^{\omega_0} \\
(aab)^\omega &= (ab)^\omega & [a(x + bc + c)]^\omega &= [a(x + bc)]^\omega
\end{aligned}
$$

$$
\begin{aligned}
a_n(c, \dots, c) + (k - n) \times c &= a_n(c, \dots, c) + (k - n + 1) \times c\,, \\
[a(x + (a(k \times x))^\pi(c))]^\omega &= k \times (a(k \times x))^\pi(c)
\end{aligned}
$$

for all $a, b, b' \in S_1$, $c, d \in S_0$, $a_n \in S_n$, and $n \leq k$.

We defer the proof to Section 5. Let us concentrate on some of the consequences first.

▶ **Corollary 4.3.** *For fixed k, it is decidable whether a given regular language L is* cEF$_k$*-definable.*

For the logic cEF, where we do not care about the value of k, a similar result can now be derived as a simple corollary. The basic argument is contained in the following lemma.

▶ **Lemma 4.4.** *Given a forest algebra $\mathfrak{A}$ that is generated by $A_0 \cup A_1$, we can compute a number K such that, if $\mathfrak{A}$ satisfies the equations of Theorem 4.2 for some value of k, it satisfies them for $k = K$.*

Proof. Set $K := m_0^{2m_1} + m_0$ where $m_0 := |A_0|$ and $m_1 := |A_1|$. By assumption there is some number k for which $\mathfrak{A}$ satisfies the equations of Theorem 4.2. W.l.o.g. we may assume that $k \geq K$. The only two equations depending on k are

$(1)_k$ $a_n(c, \dots, c) + (k - n) \times c = a_n(c, \dots, c) + (k - n + 1) \times c$
$(2)_k$ $[a(x + (a(k \times x))^\pi(c))]^\omega = k \times (a(k \times x))^\pi(c)$

We have to show that $\mathfrak{A}$ also satisfies $(1)_K$ and $(2)_K$.

For $(2)_K$, note that $k \geq K \geq |A_0|$ implies that $K \times c = \pi \times c = k \times c$, for all $c \in A_0$. Consequently,

$$a(K \times x)(c) = a(k \times x)(c) \quad \text{and, therefore,} \quad (a(K \times x))^\pi(c) = (a(k \times x))^\pi(c)\,.$$

This implies the claim.

For $(1)_K$, fix $a \in A_n$ and $c \in A_0$. If $n \leq K - m_0$, then $K - n \geq m_0 = |A_0|$ implies that $(K-n) \times c = \pi \times c$. Consequently,

$$a(c,\dots,c) + (K-n) \times c = a(c,\dots,c) + \pi \times c = a(c,\dots,c) + \pi \times c + c$$

and we are done.

Thus, we may assume that $n > K - m_0 = m_0^{2m_1}$. As $\mathfrak{A}$ is generated by $A_0 \cup A_1$, there exists some forest $s \in \mathbb{F}_i(A_0 \cup A_1)$ with $\pi(s) = a$. We distinguish several cases.

If some of the variables $x_0, \dots, x_{n-1}$ does not appear in s, we can use $(1)_k$ to show that

$$\begin{aligned} a(c,\dots,c,\dots,c) + (K-n) \times c &= a(c,\dots,c + \dots + c,\dots c) + (K-n) \times c \\ &= a(c,\dots,k \times c,\dots,c) + (K-n) \times c \\ &= a(c,\dots,k \times c,\dots,c) + (K-n) \times c + c\,. \end{aligned}$$

Next, suppose that s is highly branching in the sense that it has the form

$$s = r(t_0 + \dots + t_{m_0^2-1})$$

where each subterm t_i contains some variable. Then there are indices $i_0 < \dots < i_{m_0-1}$ such that $\pi(t_{i_0}(\bar{c})) = \dots = \pi(t_{i_{m_0-1}}(\bar{c}))$ (where $\bar{c}$ denotes as many copies of c as appear in the respective term). Hence, $(1)_k$ again implies that

$$\begin{aligned} a(\bar{c}) + (K-n) \times c &= \pi(s(\bar{c})) + (K-n) \times c \\ &= \pi\big(r\big(t_0(\bar{c}) + \dots + t_{m_0^2-1}(\bar{c})\big)\big) + (K-n) \times c \\ &= \pi\big(r\big(t_0(\bar{c}) + \dots + t_{m_0^2-1}(\bar{c}) + k \times t_{i_0}(\bar{c})\big)\big) + (K-n) \times c \\ &= a(\bar{c}) + (K-n) \times c + c\,. \end{aligned}$$

Note that a tree of height $h := m_1$ where every vertex has at most $d := m_0^2$ successors has at most $d^h = m_0^{2m_1}$ leaves. Hence, if s is not highly branching in the sense above, the fact that it contains $n > m_0^{2m_1}$ variables implies that there must be a chain $v_0 \prec \dots \prec v_{m_1}$ of vertices such that, for every $i < m_1$, there is some leaf u labelled by a variable with $v_{i-1} \prec u$ and $v_i \not\preceq u$. (For $i = 0$, we omit the first condition.) Hence, we can decompose s as

$$s(\bar{c}) = r_0(\bar{c}, r_1(\bar{c}, \dots r_{m_1}(\bar{c})))\,,$$

and there are two indices $i < j$ such that

$$\pi(r_0(\bar{c}, \dots r_i(\bar{c}, x))) = \pi(r_0(\bar{c}, \dots r_j(\bar{c}, x)))\,.$$

Consequently, we can use pumping to obtain a term

$$\pi(s(\bar{c})) = \pi\big(r_0(\bar{c}, \dots, r_i(\bar{c}, x))\big[r_{i+1}(\bar{c}, \dots, r_j(\bar{c}, x))\big]^k r_{j+1}(\bar{c}, \dots, r_{m_1}(\bar{c}))\big)$$

which contains at least k occurrences of c. Therefore, the claim follows again by $(1)_k$. ◀

According to this lemma, we can check for cEF-definability of a language L, by computing its syntactic algebra $\mathfrak{S}(L)$, the associated constant K, and then checking the equations for $k = K$.

▶ **Corollary 4.5.** *It is decidable whether a given regular language L is* cEF*-definable.*

When taking the special case of $k = 1$ in Theorem 4.2, we obtain the following characterisation of EF-definability.

▶ **Theorem 4.6.** *A forest language $L \subseteq \mathbb{F}_0\Sigma$ is definable in the logic* EF *if, and only if, the syntactic algebra $\mathfrak{S}(L)$ satisfies the following equations:*

$$c + d = d + c \qquad (a(x) + b(x))^\omega = (ab(x))^\omega$$
$$(ab)^\pi = b(ab)^\pi \qquad (a(x) + c)^\omega = (a(x + c))^\omega$$
$$(abb')^\omega = (ab'b)^\omega \qquad (a(x + c + c))^\omega = (a(x + c))^\omega$$
$$(aab)^\omega = (ab)^\omega \qquad \big[a(b(x_0, x_1))^{\omega_1}\big]^{\omega_0} = [ab(x_0, x_0)]^{\omega_0}$$
$$ac = ac + c \qquad c = c + c \qquad [a(x + a^\pi c)]^\omega = a^\pi c\,,$$

for all $a, b, b' \in S_1$ and $c, d \in S_0$.

▶ **Corollary 4.7.** *It is decidable whether a given regular language L is* EF*-definable.*

5 The proof of Theorem 4.2

For the proof of Theorem 4.2, we need to set up a bit of machinery. We start by defining the suitable notion of bisimulation for cEF_k. The difference to the standard notion is that we use reachability instead of the edge relation and that we also have to preserve the number of reachable positions.

▶ **Definition 5.1.** Let $m, k < \omega$.

(a) For trees $s, t \in \mathbb{F}\Sigma$, we define

$$
\begin{aligned}
s \approx_k^0 t \quad &:\text{iff} \quad \text{the roots of } s \text{ and } t \text{ have the same label}\\
s \approx_k^{m+1} t \quad &:\text{iff} \quad \text{the roots of } s \text{ and } t \text{ have the same label}\,,\\
&\qquad \text{for every } k\text{-tuple } \bar x \text{ in } \operatorname{dom}(s) \text{ not containing the root, there is}\\
&\qquad\quad \text{some } k\text{-tuple } \bar y \text{ in } \operatorname{dom}(t) \text{ not containing the root such that}\\
&\qquad\qquad s|_{x_i} \approx_k^m t|_{y_i} \quad \text{for all } i < k \text{ and,}\\
&\qquad \text{for every } k\text{-tuple } \bar y \text{ in } \operatorname{dom}(t) \text{ not containing the root, there is}\\
&\qquad\quad \text{some } k\text{-tuple } \bar x \text{ in } \operatorname{dom}(s) \text{ not containing the root such that}\\
&\qquad\qquad s|_{x_i} \approx_k^m t|_{y_i} \quad \text{for all } i < k\,.
\end{aligned}
$$

To simplify notation, we will frequently write $x \approx_k^m y$ for vertices x and y instead of the more cumbersome $s|_x \approx_k^m t|_y$.

(b) For forests $s, t \in \mathbb{F}\Sigma$ with possibly several components, we set

$$
\begin{aligned}
s \sim_k^{m+1} t \quad :\text{iff} \quad &\text{for every } k\text{-tuple } \bar x \text{ in } s \text{ there is some } k\text{-tuple } \bar y \text{ in } t \text{ such that}\\
&\qquad s|_{x_i} \approx_k^m t|_{y_i} \quad \text{for all } i < k \text{ and,}\\
&\text{for every } k\text{-tuple } \bar y \text{ in } t \text{ there is some } k\text{-tuple } \bar x \text{ in } s \text{ such that}\\
&\qquad s|_{x_i} \approx_k^m t|_{y_i} \quad \text{for all } i < k\,.
\end{aligned}
$$
⌟

Let us show that this notion of bisimulation captures the expressive power of cEF. The proof is mostly standard. We start by introducing the following notion of a type.

▶ **Definition 5.2.** (a) We define the *type* $\mathrm{tp}^m_k(s)$ of a tree $s \in \mathbb{F}\Sigma$ by

$$\mathrm{tp}^0_k(s) := s(\langle\rangle)$$
$$\mathrm{tp}^{m+1}_k(s) := \langle s(\langle\rangle), \theta_s\rangle$$

where $\langle\rangle$ denotes the root of s and

$$\theta_s := \big\{\, \langle l, \sigma\rangle \;\big|\; l \leq k\,,\ x_0, \dots, x_{l-1} \in \mathrm{dom}(s) \text{ distinct, not equal to the root}\,,\ \sigma = \mathrm{tp}^m_k(s|_{x_0}) = \dots = \mathrm{tp}^m_k(s|_{x_{l-1}}) \,\big\}\,.$$

(b) For an arbitrary forest $s \in \mathbb{F}\Sigma$, we set

$$\mathrm{Tp}^{m+1}_k(s) := \theta_s\,,$$

where

$$\theta_s := \big\{\, \langle l, \sigma\rangle \;\big|\; l \leq k\,,\ x_0, \dots, x_{l-1} \in \mathrm{dom}(s) \text{ distinct}\,,\ \sigma = \mathrm{tp}^m_k(s|_{x_0}) = \dots = \mathrm{tp}^m_k(s|_{x_{l-1}}) \,\big\}\,.$$ ⌟

▶ **Lemma 5.3.** *Let $k, m < \omega$.*

(a) *For trees $s, t \in \mathbb{F}_0\Sigma$, the following statements are equivalent.*

(1) $s \approx^m_k t$
(2) $\mathrm{tp}^m_k(s) = \mathrm{tp}^m_k(t)$
(3) $s \models \varphi \Leftrightarrow t \models \varphi$, *for all* $\varphi \in \mathrm{cEF}^m_k$.

(b) *For arbitrary forests $s, t \in \mathbb{F}_0\Sigma$, the following statements are equivalent.*

(1) $s \sim^m_k t$
(2) $\mathrm{Tp}^m_k(s) = \mathrm{Tp}^m_k(t)$
(3) $s \models \varphi \Leftrightarrow t \models \varphi$, *for all* $\varphi \in \mathrm{cEF}^m_k$.

Proof. (a) (2) ⇒ (1) follows by a straightforward induction on m and (1) ⇒ (3) by induction on φ. For (3) ⇒ (2) it is sufficient to show that, for every type τ, there exists a formula $\chi_\tau \in \mathrm{EF}^m_k$ such that

$$s \models \chi_\tau \quad \text{iff} \quad \mathrm{tp}^m_k(s) = \tau\,, \quad \text{for every tree } s\,.$$

We proceed by induction on m. If $m = 0$, the type τ is of the form $a \in \Sigma$. Hence, we can set $\chi_\tau := P_a$. If $m > 0$, then $\tau = \langle a, \theta\rangle$ for some $a \in \Sigma$ and some set θ of types of lower rank. We can set

$$\chi_\tau := P_a \wedge \bigwedge_{\langle l,\sigma\rangle \in \theta} \mathrm{EF}_l \chi_\sigma \wedge \bigwedge_{\langle l,\sigma\rangle \notin \theta} \neg\mathrm{EF}_l \chi_\sigma\,.$$

(b) is proved in the same way. ◀

▶ **Corollary 5.4.** *A language $L \subseteq \mathbb{F}\Sigma$ is cEF^m_k-definable if, and only if, it is regular and satisfies*

$$s \sim^m_k t \quad \textit{implies} \quad s \in L \Leftrightarrow t \in L\,, \quad \textit{for all regular forests } s, t \in \mathbb{F}_0\Sigma\,.$$

Proof. (⇒) follows by the implication (1) ⇒ (3) of Lemma 5.3.

(⇐) Set

$$\varphi := \bigvee \{ \chi_\tau \mid \tau = \mathrm{Tp}^m_k(s) \text{ for some regular forest } s \in L \},$$

where χ_τ are the formulae from the proof of Lemma 5.3. For a regular forest $t \in \mathbb{F}_0\Sigma$, it follows that

$$\begin{aligned} t \models \varphi \quad &\text{iff} \quad \mathrm{Tp}^m_k(t) = \mathrm{Tp}^m_k(s)\,, \quad \text{for some regular forest } s \in L\,, \\ &\text{iff} \quad t \sim^m_k s\,, \quad \text{for some regular forest } s \in L\,, \\ &\text{iff} \quad t \in L\,. \end{aligned}$$

Let K be the language defined by φ. Since L and K are both regular languages that contain the same regular forests, it follows that $L = K$. Thus, L is cEF^m_k-definable. ◀

We want to show that an algebra recognises cEF_k-definable languages if, and only if, it satisfies the following equations.

▶ **Definition 5.5.** (a) A forest algebra $\mathfrak{A}$ is an *algebra for* cEF_k if it is finitary, generated by $A_0 \cup A_1$, and satisfies the following equations.

(G1)$_k$ $a_n(c, \dots, c) + (k-n) \times c = a_n(c, \dots, c) + (k-n+1) \times c$
(G2) $(ab)^\pi = b(ab)^\pi$
(G3) $a^\omega + a^\omega = a^\omega$
(G4) $c + d = d + c$
(G5) $(a(x) + b(x))^\omega = (ab(x))^\omega$
(G6) $(a(x) + c)^\omega = (a(x + c))^\omega$
(G7) $(a(x + c + c))^\omega = (a(x + c))^\omega$
(G8) $\big[a(b(x_0, x_1))^{\omega_1}\big]^{\omega_0} = [ab(x_0, x_0)]^{\omega_0}$
(G9) $(abb')^\omega = (ab'b)^\omega$
(G10) $(aab)^\omega = (ab)^\omega$
(G11) $[a(x + bc + c)]^\omega = [a(x + bc)]^\omega$
(G12)$_k$ $[a(x + (a(k \times x))^\pi(c))]^\omega = k \times (a(k \times x))^\pi(c)$

where $a, b, b' \in S_1$, $c, d \in S_0$, $a_n \in S_n$, and $n \leq k$.

(b) A forest algebra $\mathfrak{A}$ is an *algebra for* cEF if it is an algebra for cEF_k, for some $k \geq 1$. ⌟

In the proof that algebras for cEF recognise exactly the cEF-definable languages, we use one of the Green's relations (suitably modified for forest algebras).

▶ **Definition 5.6.** Let $\mathfrak{A}$ be a forest algebra. For $a, b \in A_0$, we define

$$a \leq_{\mathsf{L}} b \quad :\text{iff} \quad a = c(b) \quad \text{or} \quad a = b + d\,, \quad \text{for some } c \in A_1\,,\ d \in A_0\,.$$ ⌟

▶ **Lemma 5.7.** *Let $\mathfrak{A}$ be an algebra for cEF_k.*

(a) *The relation $\leq_{\mathsf{L}}$ is antisymmetric.*
(b) *For $a \in A_1$, $c \in A_0$, we have*

$$\begin{aligned} c = c + c \quad &\textit{implies} \quad ac = ac + c\,, \\ c = a(c, c) \quad &\textit{implies} \quad c = c + c\,. \end{aligned}$$

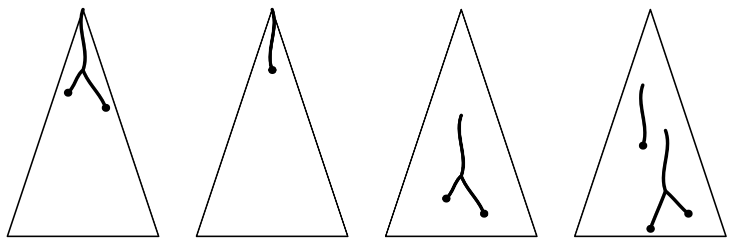

Figure 3 A forest s with a convex set U (in bold) that has three close U-ends (on the left) and five far ones (on the right). The height is $h(s,U) = 2$.

Proof. (a) For a contradiction, suppose that there are elements $a \neq b$ with $a \leq_{\mathsf{L}} b \leq_{\mathsf{L}} a$. By definition, we can find elements c and d such that (1) $a = c(b)$ or (2) $a = b + c$, and (i) $b = d(a)$ or (ii) $b = a + d$. We have thus to consider four cases. In each of them we obtain a contradiction via $(\mathrm{G1})_k$ or (G2).

$$
\begin{aligned}
(1,i) &\quad a = cb = cda = (cd)^\pi(a) = d(cd)^\pi(a) = da = b\,.\\
(1,ii) &\quad a = cb = c(a+d) = (c(x+d))^\pi(a) = (c(x+d))^\pi(a) + d = a + d = b\,.\\
(2,i) &\quad b = da = d(b+c) = (d(x+c))^\pi(b) = (d(x+c))^\pi(b) + c = b + c = a\,.\\
(2,ii) &\quad a = b + c = a + d + c = a + k \times (d+c) = a + k \times (d+c) + d = a + d = b\,.
\end{aligned}
$$

(b) By $(\mathrm{G1})_k$ we have

$$
\begin{aligned}
c = c + c \quad &\text{implies} \quad ac = a(c+c) = a(k \times c) = a(k \times c) + c = ac + c\,,\\
c = a(c,c) \quad &\text{implies} \quad c = a(c,c) = (a(x,c))^\pi(c) = (a(x,c))^\pi(c) + c = c + c\,.
\end{aligned}
$$ ◀

Let us take a look at the following situation (see Figure 3). Let s be a forest and U a set of vertices. We assume that U is *convex* in the sense that $u \preceq v \preceq w$ and $u, w \in U$ implies $v \in U$ (where $\preceq$ denotes the forest order). We call the maximal elements (w.r.t. $\preceq$) of U the *U-ends*. An U-end u is *close* if $u' \in U$, for all $u' \preceq u$. Otherwise, it is *far*. We would like to know how many of the U-ends are close.

▶ **Lemma 5.8.** *Let $m \geq 0$ and $k \geq 1$, let $s \sim_k^{m+k+2} t$ be two forests, $U \subseteq \mathrm{dom}(s)$ a convex set that is closed under $\approx_k^m$, and set*

$$V := \{\, v \in \mathrm{dom}(t) \mid u \approx_k^m v \text{ for some } u \in U \,\}\,.$$

(a) *V is convex and closed under $\approx_k^m$.*
(b) *The numbers of ends of U and V are the same, or both numbers are at least k.*
(c) *If U has less than k ends, then U is finite if, and only if, V is finite.*
(d) *If U is finite and has less than k ends, then U and V have the same numbers of close ends and of far ends.*

Proof. (a) If V is not convex, there are vertices $v \prec v' \prec v''$ of t with $v, v'' \in V$ and $v' \notin V$. Fix vertices $u \prec u' \prec u''$ with $u \approx_k^{m+2} v$, $u' \approx_k^{m+1} v'$, and $u'' \approx_k^m v''$. By definition of V, we have $u, u'' \in U$ and $u' \notin U$. This contradicts the fact that U is convex.

To see that V is closed under $\approx_k^m$, suppose that $v \in V$ and $v \approx_k^m v'$. By definition of V, there is some $u \in U$ with $u \approx_k^m v$. Hence, $u \approx_k^m v \approx_k^m v'$. As $\approx_k^m$ is transitive, this implies that $v' \in V$.

(b) For a contradiction, suppose that U has $n < k$ ends while V has more than n ends. (The other case follows by symmetry.) Choose $n+1$ ends $v_0, \dots, v_n \in V$. Since $s \approx_k^{m+2} t$, there are vertices $u_0, \dots, u_n$ in s with $u_i \approx_k^{m+1} v_i$. By definition of V, we have $u_i \in U$. By assumption, there is some index j such that u_j is not an end. Hence, we can find a vertex $u' \succ u_j$ with $u' \in U$. Fix a vertex $v' \succ v_j$ of t with $u' \approx_k^m v'$. Then $v' \in V$ and v_j is not an end. A contradiction.

(c) For a contradiction, suppose that U is finite, but V is not. (The other case follows by symmetry.) By (b), V has only finitely many ends. Hence, there is some element $v \in V$ such that $v \not\preceq v'$ for every end v' of V. Since $s \approx_k^{m+3} t$, we can find a vertex u of s with $u \approx_k^{m+2} v$. This implies that $u \in U$. As U is finite, we can find some end u' of U with $u \preceq u'$. Fix some $v' \succeq v$ with $u' \approx_k^{m+1} v'$. Then $u' \in U$ implies $v' \in V$. By choice of v, there is some $v'' \succ v'$ with $v'' \in V$. Choose $u'' \succ u'$ with $u'' \approx_k^m v''$. By choice of u', we have $u'' \notin U$. This contradicts the fact that $v'' \in V$.

(d) By (b), we only need to prove that the number of close ends is the same. Let $\hat{U}$ and $\hat{V}$ be the sets of U-ends and V-ends, respectively. We denote by $N(s, U)$ the number of close U-ends and by $F(s, U)$ the set of all proper subforests s' of s that are attached to some vertex v that does not belong to U but where at least one root belongs to U. (A forest s' is a *proper subforest* of s attached at v if s' can be obtained from the subtree $s|_v$ by removing the root v.) We define the following equivalence relation.

$$\begin{aligned}
\langle s, U\rangle \asymp_0 \langle t, V\rangle \quad &:\text{iff} \quad N(s,U) = N(t,V)\,, \\
\langle s, U\rangle \asymp_{i+1} \langle t, V\rangle \quad &:\text{iff} \quad N(s,U) = N(t,V) \text{ and} \\
&\qquad\quad \#_\tau(s,U) = \#_\tau(t,V)\,, \text{ for every } \asymp_i\text{-class } \tau,
\end{aligned}$$

where $\#_\tau(s, U)$ denotes the number of subforests $s' \in F(s, U)$ that belong to the class τ.

We define the U-*height* of s by

$$h(s,U) := \begin{cases} 0 & \text{if } F(s;U) = \emptyset \\ 1 + \max\{\, h(s',U) \mid s' \in F(s,U)\,\} & \text{otherwise.} \end{cases}$$

By induction on l, we will prove the following claim:

$$(*)\ s \sim_k^{m+l+2} t \quad\text{and}\quad h(s,U) \le l \quad\text{implies}\quad h(s,U) = h(t,V) \quad\text{and}\quad \langle s,U\rangle \asymp_l \langle t,V\rangle\,.$$

As $h(s, U) \le |\hat{U}| < k$, it then follows that $\langle s, U\rangle \asymp_k \langle t, V\rangle$. In particular, $N(s, U) = N(t, V)$, as desired.

It thus remains to prove $(*)$. First, consider the case where $l = 0$. If $h(t, V) > 0$, there is some V-end v that is not close. Fix some vertex $v' \prec v$ with $v' \notin V$. Since $s \sim_k^{m+2} t$, we can find vertices $u' \prec u$ of s with $u' \approx_k^{m+1} v'$ and $u \approx_k^m v$. By definition of V, it follows that $u' \notin U$ and $u \in U$. As U is finite, we can find some U-end $w \succeq u$. But $u' \prec u \preceq w$ implies that w is not close. Hence, $h(s, U) > 0$. A contradiction.

For the second part, suppose that $\langle s, U\rangle \not\asymp_0 \langle t, V\rangle$, that is, $N(s, U) \neq N(t, V)$. By symmetry, we may assume that $m := N(s, U) < N(t, v)$. Pick $m + 1$ distinct close V-ends $v_0, \dots, v_m$. Since $m + 1 \le k$ and $s \sim_k^{m+2} t$, there are elements $u_0, \dots, u_m \in \mathrm{dom}(s)$ with $u_i \approx_k^{m+1} v_i$. There must be some index j such that u_j is not a close U-end. As U is closed under $\approx_k^m$ and $u_j \approx_k^m v_j \approx_k^m u$, for some $u \in U$, it follows that $u_j \in U$. Furthermore, $u_j \approx_k^{m+1} v_j$ and the fact that v_j is a V-end implies that $u' \notin U$, for all $u' \succ u_j$. Thus, u_j is a U-end. But $h(s, U) = 0$ implies that all U-ends of s are close. A contradiction.

For the inductive step, suppose that $s \sim_k^{m+(l+1)+2} t$ holds but we have $h(s, U) \neq h(t, V)$ or $\langle s, U\rangle \not\asymp_{l+1} \langle t, V\rangle$. We distinguish several cases.

(i) Suppose that $h(s,U) > h(t,V)$. By definition of h, there is a subforest $s' \in F(s,U)$ with $h(s',U) = h(s,U) - 1$. Then there is some subforest t' of t with $s' \sim_k^{m+l+2} t'$. By inductive hypothesis it follows that

$$h(s,U) = h(s',U) + 1 = h(t',V) + 1 < h(t,V) + 1 \leq h(s,U)\,.$$

A contradiction.

(ii) Suppose that $h(s,U) < h(t,V)$. By definition of h, there is a subforest $t' \in F(t,V)$ with $h(t',V) = h(t,V) - 1$. Fix a subforest s' of s with $s' \sim_k^{m+l+2} t'$. By inductive hypothesis, it follows that

$$h(s,U) > h(s',U) = h(t',V) = h(t,V) - 1 \geq h(s,U)\,.$$

A contradiction.

(iii) Suppose that $N(s,U) \neq N(t,v)$ and there is no $\asymp_l$-class τ with $\#_\tau(s,U) \neq \#_\tau(t,V)$. Then we have $|\hat{U}| - N(s,U) = |\hat{V}| - N(t,V)$. Since $|\hat{U}| = |\hat{V}|$ it follows that $N(s,U) = N(t,V)$. A contradiction.

(iv) Finally, suppose that there is some $\asymp_l$-class τ with $\#_\tau(s,U) \neq \#_\tau(t,V)$. By symmetry, we may assume that $m := \#_\tau(s,U) < \#_\tau(t,V)$. We choose $m+1$ vertices $v_0, \dots, v_m$ of t such that the attached subforests have class τ. Since $s \sim_k^{m+(l+1)+2} t$ and $m+1 \leq k$, there are vertices $u_0, \dots, u_m$ of s such that $u_i \sim_k^{m+l+2} v_i$, for all $i \leq m$. Let s_i be the subforest of s attached to u_i, and t_i the subforest of t attached to v_i. By inductive hypothesis, it follows that $s_i \asymp_l t_i$, for $i \leq m$. Thus, s has at least $m+1$ different subforest in the class τ. A contradiction. ◀

Bevor presenting our main technical result, let us quickly recall how to solve a system of equations using a fixed-point operator. Suppose we are given a system of the form

$$\begin{aligned} x_0 &= r_0(x_0, \dots, x_{n-1})\,, \\ &\vdots \\ x_{n-1} &= r_{n-1}(x_0, \dots, x_{n-1})\,, \end{aligned}$$

where $r_0, \dots, r_{n-1} \in \mathbb{F}_n A$. Inductively defining

$$s_i(x_0, \dots, x_{i-1}) := (r_i(x_0, \dots, x_i, s_{i+1}, \dots, s_{n-1}))^{\omega_i}\,,$$

we obtain the new system

$$\begin{aligned} x_0 &= s_0\,, \\ x_1 &= s_1(x_0)\,, \\ &\vdots \\ x_{n-1} &= s_{n-1}(x_0, \dots, x_{n-2})\,, \end{aligned}$$

which can now be solved by substitution.

▶ **Proposition 5.9.** *Let $\mathfrak{A}$ be an algebra for cEF_k. Then*

$$s \approx_k^{(k+3)(|A_0|+1)} t \quad \textit{implies} \quad \pi(s) = \pi(t)\,, \quad \textit{for all regular trees } s, t \in \mathbb{F}_0(A_0 \cup A_1)\,.$$

Proof. Let m be the number of L-classes above $b := \pi(s)$ (including that of b itself). We will prove by induction on m that

$$s \approx_k^{f(m)} t \quad \text{implies} \quad \pi(t) = b\,,$$

where $f(m) := (m+1)(k+3)$. Set

$$S := \{\, x \in \mathrm{dom}(s) \mid \pi(s|_x) = b \,\}\,,$$
$$T := \{\, y \in \mathrm{dom}(t) \mid x \approx^{f(m-1)} y \text{ for some } x \in S \,\}\,.$$

As t is regular it is the unravelling of some finite graph G. For each $y \in T$, we will prove that $\pi(t|_y) = b$ by induction on the number of strongly connected components of G that are contained in T and that are reachable from y. Hence, fix $y \in T$, let C be the strongly connected component of G containing y, and choose some $x \in S$ with $x \approx_k^{f(m)-1} y$. We distinguish two cases.

(a) Let us begin our induction with the case where C is trivial, i.e., it consists of the single vertex y without self-loop. Then

$$t|_y = a(t_0 + \dots + t_{n-1} + t'_0 + \dots + t'_{q-1})$$

where $a := t(y)$ and the subtrees t_i lie outside of T while the t'_i contain vertices in T. Set $d_i := \pi(t_i)$. By our two inductive hypotheses, we already know that $\pi(t'_i) = b$ and that $b <_{\mathsf{L}} d_i$. Hence,

$$\pi(t|_y) = a(d_0 + \dots + d_{n-1} + q \times b)\,.$$

We have to show that this value is equal to b. Suppose that

$$s|_x = a(s_0 + \dots + s_{l-1} + s'_0 + \dots + s'_{p-1})\,,$$

where again the trees s_i lie outside of S, while the s'_i contain vertices of S. Setting $c_i := \pi(s_i)$ it follows that

$$\pi(s|_x) = a(c_0 + \dots + c_{l-1} + p \times b)\,.$$

Since $x \in S$, we already know that this value is equal to b. Hence, it remains to show that

$$a(c_0 + \dots + c_{l-1} + p \times b) = a(d_0 + \dots + d_{n-1} + q \times b)\,.$$

We start by proving that

$$c_0 + \dots + c_{l-1} = d_0 + \dots + d_{n-1}\,.$$

By (G4) it is sufficient to prove that, for every $c \in A_0$, the number of occurrences of the value c in the sum on the left-hand side is either the same as that on the right-hand side, or that we can add an arbitrary number of c on both sides without changing the respective values. Hence, consider some element $c \in A_0$ where these numbers are different. Let U be the set of all vertices $u \succ x$ such that $\pi(s|_u) = c$ and let V be the set of vertices $v \succ y$ with $\pi(t|_v) = c$. As $\leq_{\mathsf{L}}$ is antisymmetric, these two sets are convex. Furthermore, by inductive hypothesis on m, they are also closed under $\approx_k^{f(m-1)}$. Since $f(m) - 1 = f(m-1) + k + 2$, we can therefore apply Lemma 5.8 and we obtain one of the following cases.

(i) U and V both have at least k ends. Then we can write $s_0 + \dots + s_{l-1}$ as $r(s'_0, \dots, s'_{k-1})$ with $\pi(s'_i) = c$. Hence, it follows by $(\mathrm{G1})_k$ that

$$c_0 + \dots + c_{l-1} = \pi(r)(c, \dots, c) = \pi(r)(c, \dots, c) + \pi \times c = c_0 + \dots + c_{l-1} + \pi \times c\,.$$

For t it follows in the same way that

$$d_0 + \cdots + d_{n-1} = d_0 + \cdots + d_{n-1} + \pi \times c\,.$$

Consequently, we can add an arbitrary number of terms c to both sides of the above equation and thereby make their numbers equal.

(ii) Both U and V are infinite, but each has less than k ends. Thus, U contains an infinite path and we can use Ramsey's Theorem (or the fact that s is regular) to write $\pi(s_0 + \cdots + s_{l-1})$ as $a'e^\omega$ where $ec = c = e^\omega$. By (G3) and $(G1)_k$ it follows that

$$\begin{aligned} c_0 + \cdots + c_{l-1} &= a'e^\omega = a'(e^\omega + \cdots + e^\omega) = a'(c + \cdots + c) \\ &= a'(c + \cdots + c) + \pi \times c \\ &= c_0 + \cdots + c_{l-1} + \pi \times c\,. \end{aligned}$$

For $t|_y$, we similarly obtain

$$d_0 + \cdots + d_{n-1} = d_0 + \cdots + d_{n-1} + \pi \times c\,,$$

and we can equalise the number of c as in Case (i).

(iii) The last remaining case is where both U and V are finite and they have the same number of close ends. Then the sums $c_0 + \cdots + c_{l-1}$ and $d_0 + \cdots + d_{n-1}$ contain the same number of terms with value c and there is nothing to prove.

We have thus shown that

$$c_0 + \cdots + c_{l-1} = d_0 + \cdots + d_{n-1}\,.$$

If $p = q$, we are done. Hence, we may assume that $p \neq q$. To conclude the proof, we set

$$U := \{\, u \in S \mid x \prec u \,\} \quad \text{and} \quad V := \{\, v \in T \mid y \prec v \,\}\,.$$

If $p > 0$, then $x \approx_k^{f(m)-1} y$ and $U \neq \emptyset$ implies $V \neq \emptyset$. Hence, $q > 0$. In the same way, $q > 0$ implies $p > 0$. Consequently, we have $p, q > 0$. We consider several cases.

(i) If $b + b = b$, then

$$a(d_0 + \cdots + d_{n-1} + q \times b) = a(c_0 + \cdots + c_{l-1} + q \times b) = a(c_0 + \cdots + c_{l-1} + p \times b) = b\,,$$

as desired.

(ii) If U is not a chain, we obtain $b = a'(b, b)$, for some a', and Lemma 5.7 implies that we are in Case (i).

(iii) If U contains an infinite chain, we can use Ramsey's Theorem (or the fact that s is regular), to obtain a factorisation $b = e^\omega$, which implies that $b + b = b$ by (G3). Hence, we are in Case (i) again.

(iv) If U is a finite chain, then so is V, by Lemma 5.8. Hence, $p = 1 = q$ and we are done.

(b) It remains to consider the case where C is not trivial. Then we can factorise

$$t|_y = r(t_0, \ldots, t_{n-1}, t'_0, \ldots, t'_{q-1})\,,$$

where $r \in \mathbb{F}A$ is the unravelling of C, the subtrees t_i lie outside of T, while the subtrees t'_i contain vertices in T. Setting $d_i := \pi(t_i)$, it follows by the two inductive hypotheses that $d_i >_{\mathsf{L}} b$ and $\pi(t'_i) = b$. Consequently,

$$\pi(t|_y) = \pi(r)(d_0, \ldots, d_{n-1}, b, \ldots, b)\,.$$

Let us simplify the term r. Introducing one variable x_v, for every vertex $v \in C$, we can write r as a system of equations

$$x_v = a_v(x_{u_0} + \dots + x_{u_{l-1}} + c_0 + \dots + c_{q-1}), \quad \text{for } v \in C,$$

where $u_0, \dots, u_{l-1}$ are the successors of v that belong to C and $c_0, \dots, c_{q-1}$ are constants from $\{d_0, \dots, d_{n-1}, b\}$ that correspond to successors outside of C. Solving this system of equations in the way we explained above, we obtain a finite term r_0 built up from elements of $A_0 \cup A_1$ using as operations the horizontal product, the vertical product, and the ω-power operation, such that

$$\pi(t|_y) = \pi(r_0)(d_0, \dots, d_{n-1}, b).$$

With the help of the equations (G5)–(G10), we can transform r_0 in several steps (while preserving its product) until it assumes the form

$$\big[a_0 \cdots a_{j-1}\big(x + d_0 + \dots + d_{n-1} + b\big)\big]^\omega$$
$$\text{or} \quad \big[a_0 \cdots a_{j-1}\big(x + d_0 + \dots + d_{n-1}\big)\big]^\omega$$

where $a_0, \dots, a_{j-1}$ are the labels of the vertices in C.

We distinguish two cases. First suppose that there is no term with value b in the above sum. This means that every subtree attached to C lies entirely outside of the set T. Then $x \approx_k^{f(m)-1} y$ implies that we can factorise $s|_x$ as

$$s|_x = r'(s_0, \dots, s_{l-1})$$

where

- $\{\pi(s_0), \dots, \pi(s_{l-1})\} = \{d_0, \dots, d_{n-1}\}$,
- all labels of r' are among $a_0, \dots, a_{j-1}$,
- every vertex of r' has, for every $i < k$, some descendant labelled a_i.

As above we can transform $s|_x$ into

$$\big[a_0 \cdots a_{j-1}\big(x + c_0 + \dots + c_{l-1}\big)\big]^\omega$$

where $c_i := \pi(s_i)$. Since $\{c_0, \dots, c_{l-1}\} = \{d_0, \dots, d_{n-1}\}$ it follows that

$$\begin{aligned}\pi(t|_y) &= (a_0 \cdots a_{j-1}(x + d_0 + \dots + d_{n-1}))^\omega \\ &= (a_0 \cdots a_{j-1}(x + c_0 + \dots + c_{l-1}))^\omega = \pi(s|_x) = b.\end{aligned}$$

It thus remains to consider the case where some term has value b. Using (G7) and (G11) and the fact that $b <_{\mathsf{L}} d_i$, it then follows that

$$\pi(t|_y) = \big[a_0 \cdots a_{j-1}\big(x + d_0 + \dots + d_{n-1} + b\big)\big]^\omega = \big[a_0 \cdots a_{j-1}(x + b)\big]^\omega.$$

For every $i < j$, we fix some $z_i \in S$ with label a_i such that $x \prec z_i$ and some successor of z_i also belongs to S. Then

$$\pi(s|_{z_i}) = a_i(c_0^i + \dots + c_{l_i-1}^i + b + \dots + b),$$

for some $c_0^i, \dots, c_{l_i-1}^i >_{\mathsf{L}} b$. Since

$$b = \pi(s|_{z_i}) = a_i(c_0^i + \dots + c_{l_i-1}^i + b + \dots + b) \leq_{\mathsf{L}} c_0^i + \dots + c_{l_i+1}^i + b + \dots + b \leq_{\mathsf{L}} b$$

it follows by asymmetry of $\leq_{\mathsf{L}}$ that

$$c^i_0 + \cdots + c^i_{l_i+1} + b + \cdots + b = b \quad \text{and} \quad a_i(b) = a_i(c^i_0 + \cdots + c^i_{l_i+1} + b + \cdots + b) = b\,.$$

Consequently, $a_0 \cdots a_{j-1} b = b$, which implies that $a^\pi b = b$ where $a := a_0 \cdots a_{j-1}$. We claim that $b + b = b$. It then follows that

$$b = a(b) = a(k \times x)(b) = (a(k \times x))^\pi(b)\,,$$

which, by $(\mathrm{G12})_k$, implies that

$$\pi(t|_y) = [a(x+b)]^\omega = [a(x + a(k \times x)^\pi(b))]^\omega = k \times a(k \times x)^\pi(b) = k \times b = b\,,$$

as desired.

Hence, it remains to prove our claim that $b + b = b$. By our assumption on y and C, there is some vertex $u \in C$ that has some successor $v \notin C$ with $v \in T$. Since $s|_x \approx_k^{f(m)-1} t|_y$ and $f(m) \geq f(m-1) + k + 1$, there are vertices $x \preceq u_0 \prec \cdots \prec u_{k-1}$ each of which has some successor $v_i \in S$ with $v_i \not\preceq u_{i+1}$. Consequently, we can write

$$\pi(s|_x) = a'a''(b, \dots, b) \quad \text{and} \quad \pi(s|_{u_0}) = a''(b, \dots, b)\,,$$

where $a' \in A_1$ and $a'' \in A_k$. Hence, it follows by $(\mathrm{G1})_k$ that

$$b + b = \pi(s|_{u_0}) + b = a''(b, \dots, b) + b = a''(b, \dots, b) = \pi(s|_{u_0}) = b\,. \qquad \blacktriangleleft$$

▶ **Theorem 5.10.** *A regular forest algebra $\mathfrak{A}$ is an algebra for* cEF_k *if, and only if, there exists a number $m < \omega$ such that*

$$s \sim_k^m t \quad \textit{implies} \quad \pi(s) = \pi(t)\,, \quad \textit{for all regular forests } s, t \in \mathbb{F}(A_0 \cup A_1)\,.$$

Proof. ($\Leftarrow$) In each of the equations $(\mathrm{G1})_k$–$(\mathrm{G12})_k$, the two terms on both sides are $\sim_k^m$-equivalent.

($\Rightarrow$) By Proposition 5.9, there is some number m such that

$$s \approx_k^m t \quad \text{implies} \quad \pi(s) = \pi(t)\,, \quad \text{for regular trees } s, t \in \mathbb{F}(A_0 \cup A_1)\,.$$

Let $s, t \in \mathbb{F}(A_0 \cup A_1)$ be regular forests. We claim that

$$s \sim_k^{m+k+2} t \quad \text{implies} \quad \pi(s) = \pi(t)\,.$$

Suppose that $s = s_0 + \cdots + s_{l-1}$ and $t = t_0 + \cdots + t_{n-1}$, for trees s_i and t_i, and set $c_i := \pi(s_i)$ and $d_i := \pi(t_i)$. Analogous to Part (a) of the proof of Proposition 5.9, we can use Lemma 5.8 to show that

$$\pi(s) = c_0 + \cdots + c_{l-1} = d_0 + \cdots + d_{n-1} = \pi(t)\,. \qquad \blacktriangleleft$$

We complete the proof of Theorem 4.2 as follows.

▶ **Theorem 5.11.** *A regular language $L \subseteq \mathbb{F}_0\Sigma$ is* cEF_k*-definable if, and only if, its syntactic algebra $\mathfrak{S}(L)$ is an algebra for* cEF_k*.*

Proof. ($\Leftarrow$) Suppose that $\mathfrak{S}(L)$ is an algebra for cEF_k. By Theorem 5.10, every language recognised by $\mathfrak{S}(L)$ is invariant under $\sim_k^m$, for some m (when considering regular forests only). Consequently, the claim follows by Corollary 5.4.

($\Rightarrow$) If L is cEF_k-definable, it follows by Corollary 5.4 that L is $\sim_k^m$-invariant, for some m. Thus $\sim_k^m$ is contained in the syntactic congruence of L, which means that the syntactic morphism $\eta : \mathbb{F}\Sigma \to \mathfrak{S}(L)$ maps $\sim_k^m$-equivalent forests to the same value. Given forests $s, t \in \mathbb{F}(S_0 \cup S_1)$ with $s \sim_k^m t$, we can choose forests $s', t' \in \mathbb{F}\Sigma$ with $s' \sim_k^m t'$ and $s(v) = \eta(s'(v))$ and $t(v) = \eta(t'(v))$. Thus,

$$s \sim_k^m t \quad \text{implies} \quad \pi(s) = \eta(s') = \eta(t') = \pi(t)\,.$$

By Theorem 5.10, it follows that $\mathfrak{S}(L)$ is an algebra for cEF_k. ◀

References

1 A. Blumensath, *Branch-Continuous Tree Algebras.* arXiv:1807.04568, unpublished.

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

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

4 ———, *Algebraic Language Theory for Eilenberg-Moore Algebras.* unpublished, 2020.

5 ———, *Regular Tree Algebras*, Logical Methods in Computer Science, 16 (2020), pp. 16:1–16:25.

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 finite algebra*, Logical Methods in Computer Science, 15 (2019).

10 M. Bojańczyk and I. Walukiewicz, *Forest Algebras*, in Logic and Automata: History and Perspectives, J. Flum, E. Grädel, and T. Wilke, eds., Amsterdam University Press, 2007, pp. 107–132.