On logically defined recognizable tree languages

Zoltán Ésik[1] and Pascal Weil[2]

[1] Department of Computer Science, University of Szeged, ze@inf.u-szeged.hu
[2] LaBRI, CNRS and Université Bordeaux-1, pascal.weil@labri.fr

Abstract. We provide an algebraic characterization of the expressive power of various naturally defined logics on finite trees. These logics are described in terms of Lindström quantifiers, and particular cases include first-order logic and modular logic. The algebraic characterization we give is expressed in terms of a new algebraic structure, finitary preclones, and uses a generalization of the block product operation.

1 Introduction

The notion of recognizability emerged in the 1960s (Eilenberg, Mezei, Wright, and others, cf. [12, 22]) and has been the subject of considerable attention since, notably because of its close connections with automata-theoretic formalisms and with logical definability, cf. [4, 9, 13, 30] for some early papers.

Recognizability was first considered for sets (languages) of finite words, cf. [11] and the references contained in op. cit. The general idea is to use the algebraic structure of the domain, say, the monoid structure on the set of all finite words, to describe some of its subsets. More precisely, a subset of an algebra is said to be recognizable if it is a union of classes in a (locally) finite congruence. The same concept was adapted to the case of finite trees, traces, finite graphs, etc, cf. [12, 22, 8, 6].

It follows rather directly from this definition of (algebraic) recognizability that a finite – or finitary – algebraic structure can be canonically associated with each recognizable subset L, called its syntactic structure. Moreover, the algebraic properties of the syntactic structure of L reflect its combinatorial and logical properties. The archetypal example is that of star-free languages of finite words: they are exactly the languages whose syntactic monoid is aperiodic, cf. [26]. They are also exactly the languages that can be defined by a first-order (FO) sentence, cf. [21], and the languages that can be defined by a temporal logic formula, cf. [18, 16, 5]. In particular, if we want to decide whether a given regular language L is FO-definable, we do not know any algorithm that does not, in one form or another, verify that the syntactic monoid of L is aperiodic.

An important open problem is the analogous question concerning languages of finite trees [24]: can we decide whether a regular tree language is FO-definable? Based on the existing literature, it is tempting to guess that an answer to this problem could be found using algebraic methods. A central motivation for this paper is to present an algebraic framework which allows a nice characterization

of FO-definable tree languages. Let us say immediately that we do not know yet whether this characterization can be turned into a decision algorithm!

Let Σ be a ranked alphabet. The set of Σ-labeled trees can be seen in a natural way as a (free) Σ-algebra, where Σ is now seen as a signature. It has been known since [12, 22, 9] that the regular tree languages are exactly the recognizable subsets of this Σ-algebra. We refer the reader to [17, 23, 24] for attempts to use this algebraic framework and some of its variants to characterize FO-definable tree languages. In this paper, we propose a different algebraic view – which preserves however the recognizable sets of trees.

More precisely, we consider algebras called preclones (they lack some of the operations and axioms of clones [7]). Precise definitions are given in Section 2.1. Let us simply say here that, in contrast with the more classical monoids or Σ-algebras, preclones have infinitely many sorts, one for each integer $n \geq 0$. As a result, there is no nontrivial finite preclone. The corresponding notion is that of finitary preclones, that have a finite number of elements of each sort. An important class of preclones is given by the transformations $T(Q)$ of a set Q. The elements of sort (or rank) n are the mappings from Q^n into Q and the (preclone) composition operation is the usual composition of mappings. Note that $T(Q)$ is finitary if Q is finite.

It turns out that the finite Σ-labeled trees can be identified with the 0-sort of the free preclone generated by Σ. The naturally defined syntactic preclone of a tree language L is finitary if and only if L is regular. In fact, if S is the syntactic Σ-algebra of L, the syntactic preclone is the sub-preclone of $T(S)$ generated by the elements of Σ (if $\sigma \in \Sigma$ is an operation of rank r, it defines a mapping from S^r into S, and hence an element of sort r in $T(S)$). Note that this provides an effectively constructible description of the syntactic preclone of L.

One can develop the expected theory of varieties of recognizable tree languages and pseudovarieties of preclones, leading to an Eilenberg-type variety theorem, related to that presented in [14]. This requires combinatorially much more complex proofs than in the classical settings, and we give a brief overview of this set of results, as needed for the sequel of the paper.

However our main concern in this paper is to give algebraic characterizations of certain natural logically defined classes of tree languages. A representative example of such classes is that of the FO-definable tree languages, but our results also apply to the $(FO + MOD)$-definable tree languages of [23, 29] and many other classes. The common point of these classes of formulas is that they use new quantifiers, each of which is based on a regular tree language. For instance, the usual existential quantifier is associated with the language of those trees containing at least one vertex labeled by a symbol corresponding to the truth value 1. (See Example 10 for a more precise description).

The algebraic characterization which we obtain uses a notion of block product (or 2-sided wreath product) of preclones, inspired by Rhodes and Tilson's block product [25] and Eilenberg's bimachines [11].

Technically, if $\mathcal{K}$ is a family of regular tree languages and $\mathbf{V}$ is the pseudovariety of preclones generated by the syntactic preclones of the elements of $\mathcal{K}$, if

Lind($\mathcal{K}$) is the formal logic whose formulas use the language of Σ-labeled trees, the Boolean connectives and the Lindström quantifiers [19, 10] associated with the languages of $\mathcal{K}$, then a regular tree language is **Lind**($\mathcal{K}$)-definable if and only if its syntactic preclone lies in the least pseudovariety of preclones containing $\mathbf{V}$ and closed under block product. To be completely accurate, the preclones in this statement must formally be accompanied by a designated subset of generators, and $\mathcal{K}$ and **Lind**($\mathcal{K}$) must satisfy certain simple closure properties.

Returning to FO-definable tree languages, this tells us that a tree language is FO-definable if and only if its syntactic preclone lies in the least pseudovariety closed under block product and containing the sub-preclone of the preclone of transformations of the two-element set $\{0, 1\}$, generated by the binary or function and the (nullary) constants $0, 1$. As pointed out earlier, we do not know whether this yields a decidability proof for FO-definable tree languages, but it constitutes at least an avenue to be explored in the search for such a decision procedure.

In order to keep this paper within the required format, full proofs are reserved for a later publication.

2 The algebraic framework

Let Q be a set and let $T_n(Q)$ denote the set of n-ary transformations of Q, that is, mappings from Q^n to Q. Let then $T(Q) = (T_n(Q))_{n\geq 0}$, called the *preclone of transformations* of Q. The set $T_1(Q)$ of transformations of Q is a monoid under the composition of functions. Composition can be considered on $T(Q)$ in general: if $f \in T_n(Q)$ and $g_i \in T_{m_i}(Q)$ $(1 \leq i \leq n)$, then the composite $f(g_1, \ldots, g_n)$, defined in the natural way, is an element of $T_m(Q)$ where $m = \sum_{i=1}^{n} m_i$. This composition operation and its associativity properties are exactly what is captured in the notion of a preclone.

Remark 1. Preclones are an abstraction of sets of n-ary transformations of a set, which generalizes the abstraction from transformation monoids to monoids. Clones, [7], or equivalently, Lawvere theories [3, 14] are another such abstraction, more classical. We will not take the space to discuss the differences between clones and preclones, simply pointing here the fact that each of the m arguments of the composite $f(g_1, \ldots, g_n)$ above is used in exactly one of the g_i, in contrast with the definition of the clone of transformations of Q. Readers interested in this comparison will have no difficulty to trace those differences in the sequel. The category of preclones is equivalent to the category of strict monoidal categories [20] or magmoids [2] "generated by a single object".

2.1 Preclones

A *preclone* is a many-sorted algebra $S = ((S_n)_{n\geq 0}, \cdot, \mathbf{1})$, where n ranges over the nonnegative integers, equipped with a *composition operation* $\cdot$ such that for each $f \in S_n$ and $g_1 \in S_{m_1}, \ldots, g_n \in S_{m_n}$, $\cdot(f, g_1, \ldots, g_n) \in S_m$ where $m = \sum_{i\in[n]} m_i$.

We usually write $f \cdot (g_1 \oplus \cdots \oplus g_n)$ for $\cdot(f, g_1, \ldots, g_n)$. The constant $\mathbf{1}$ is in S_1. We require the following three equational axioms:

$$(f \cdot (g_1 \oplus \cdots \oplus g_n)) \cdot (h_1 \oplus \cdots \oplus h_m) = f \cdot ((g_1 \cdot \overline{h}_1) \oplus \cdots \oplus (g_n \cdot \overline{h}_n)), \quad (1)$$

where $f, g_1, \ldots, g_n$ are as above, $h_j \in S_{k_j}$, $j \in [m]$, and $\overline{h}_i = h_{m_1+\ldots+m_{i-1}+1} \oplus \cdots \oplus h_{m_1+\ldots+m_i}$, $i \in [n]$, and

$$\mathbf{1} \cdot f = f \quad (2)$$

$$f \cdot (\mathbf{1} \oplus \cdots \oplus \mathbf{1}) = f, \quad (3)$$

where $f \in S_n$ and $\mathbf{1}$ appears n times on the left hand side of the last equation. An element of S_n is said to have *rank* n.

The notions of morphism between preclones, sub-preclone, congruence and quotient are defined as usual. Note that a morphism maps elements of rank n to elements of the same rank, and that a congruence only relates elements of the same rank. It is not difficult to establish the following.

Fact 2. *Every preclone can be embedded in a preclone of transformations.*

We say that a preclone S is *finitary* if each S_n is finite. For instance, if Q is a finite set, then $T(Q)$ is finitary. Note that a finitary preclone S does not necessarily embed in the transformation preclone of a finite set.

For technical reasons it is sometimes preferable to work with *generated preclones* (gp's), consisting of a pair (S, A) where S is a preclone, A is a nonempty subset of S, and S is generated by A. The notions of morphisms and congruences must be revised accordingly: in particular, a morphism of gp's from (S, A) to (T, B) must map A into B. A gp (S, A) is said to be finitary if S is finitary and A is finite.

Example 3. Let Σ be a ranked alphabet, so that Σ is a finite set of ranked symbols, and let Q be a Σ-algebra. Recall that Q can also be described as (the set of states of) a tree automaton accepting Σ-labeled trees. The elements of Σ of rank n can be viewed naturally as elements of $T_n(Q)$. The preclone they generate within $T(Q)$, together with Σ, is called the *gp associated with* Q.

2.2 Trees and free preclones

Let Σ be a ranked alphabet and let $(v_k)_{k\geq 1}$ be a sequence of variable names. We let ΣM_n be the set of finite trees whose inner nodes are labeled by elements of Σ (according to their rank), whose leaves are labeled by elements of $\Sigma_0 \cup \{v_1, \ldots, v_n\}$, and whose *frontier* (the left to right sequence of variables appearing in the tree) is the word $v_1 \cdots v_n$: that is, each variable occurs exactly once, and in the natural order. Note that ΣM_0 is the set of finite Σ-labeled trees. We let $\Sigma M = (\Sigma M_n)_n$.

The composite tree $f \cdot (g_1 \oplus \cdots \oplus g_n)$ ($f \in \Sigma M_n$) is obtained by substituting the root of the tree g_i for the variable v_i in f, and renumbering consecutively

the variables in the frontiers of $g_1, \ldots, g_n$. Let also $\mathbf{1} \in \Sigma M_1$ be the tree with a single vertex (labeled v_1). Then $(\Sigma M, \cdot, \mathbf{1})$ is a preclone.

Each letter $\sigma \in \Sigma$ of rank n can be identified with the tree with root labeled σ, where the root's immediate successors are leaves labeled $v_1, \ldots, v_n$. Then every rank-preserving map from Σ to a preclone S can be extended in a unique fashion to a preclone morphism from ΣM into T. That is:

Fact 4. *ΣM is the free preclone generated by Σ, and $(\Sigma M, \Sigma)$ is the free gp generated by Σ.*

Note that ΣM_n is nonempty for all $n \geq 0$ exactly when Σ_0 and at least one Σ_n with $n > 1$ are nonempty. Below we will only consider such ranked sets. Moreover, we will only consider preclones S such that S_n is nonempty for all $n \geq 0$.

3 Recognizable tree languages

The algebraic framework described in Section 2 leads naturally to a definition of recognizable languages: a subset L of ΣM_k is recognizable if there exists a morphism α from ΣM to a finitary preclone S such that $L = \alpha^{-1}\alpha(L)$. As usual, the notion of recognizability can be expressed equivalently by stating that L is saturated by some locally finite congruence on ΣM (a congruence is locally finite if it has finite index on each sort).

If $L \subseteq \Sigma M_k$ is any tree language, recognizable or not, then there is a coarsest congruence $\sim_L$ saturating it, called the *syntactic congruence* of L. This congruence can be described as follows. First, an *m-ary context* in ΣM_k is a tuple (u, k_1, k_2, v) where

- v is an m-tuple $(v_1, \ldots, v_m)$, written $v = v_1 \oplus \cdots \oplus v_m$, with $v_i \in T_{\ell_i}$, $1 \leq i \leq m$,
- $u \in T_{k_1+1+k_2}$ and
- $k = k_1 + \ell + k_2$ with $\ell = \sum_{i=1}^{m} \ell_i$.

(u, k_1, k_2, v) is an *L-context* of an element $f \in \Sigma M_m$ if $u \cdot (\mathbf{k}_1 \oplus f \cdot v \oplus \mathbf{k}_2) \in L$. Here $\mathbf{n}$ denotes the $\oplus$-sum of n terms equal to $\mathbf{1}$. Then, for each $f, g \in \Sigma M_m$, we let $f \sim_L g$ iff f and g have the same L-contexts. We denote by (M_L, Σ_L) the quotient gp $(\Sigma M/\sim_L, \Sigma/\sim_L)$, called the *syntactic gp* of L. M_L is the *syntactic preclone* of L and the projection morphism $\eta_L \colon \Sigma M \to M_L$ is the *syntactic morphism* of L.

Fact 5. *The congruence $\sim_L$ of a language $L \subseteq \Sigma M_k$ is the coarsest preclone congruence that saturates L. In other words, if $\alpha \colon \Sigma M \to S$ is a preclone morphism, $L = \alpha^{-1}\alpha(L)$ if and only if α can be factored through η_L. In particular, L is recognizable if and only if $\sim_L$ is locally finite, if and only if M_L is finitary.*

One can also show the following proposition, relating preclone recognizability with the usual notion of regular tree languagess.

Proposition 6. *The syntactic gp of a tree language $L \subseteq \Sigma M_0$ is the gp associated with the syntactic Σ-algebra of L. In particular, L is recognizable if and only if L is a regular tree language.*

While not difficult, this result is important because it shows that we are not introducing a new class of recognizable tree languages: we are simply associating with each regular tree language a finitary algebraic structure which is richer than its syntactic Σ-algebra (a.k.a. minimal deterministic tree automaton). The proposition implies that the syntactic gp of a recognizable tree language has an (effectively computable) finite presentation.

One can define *pseudovarieties* of preclones as those nonempty classes of finitary preclones closed under direct product, sub-preclones, quotients, finitary unions of ω-chains and finitary inverse limits of ω-sequences. (The latter two constructs are needed because preclones have an infinite number of sorts.) Here, we say that a union $T = \cup_n T_n$ of an ω-chain of preclones T_n, $n \geq 0$ is finitary exactly when T is finitary. Finitary inverse limits $\lim_n T_n$ of ω-diagrams $h_n : T_{n+1} \to T_n,\ n \geq 0$ are defined in the same way. Also, one can define *pseudovarieties of gp's* as those classes of finitary gp's closed under direct product, sub-preclones, quotients and finitary inverse limits of ω-sequences. (Closure under finitary unions of ω-chains comes for free, since all finitary gp's are finitely generated.)

Suppose now that $\mathcal{V}$ is a nonempty class of recognizable tree languages $L \subseteq \Sigma M_k$, where Σ is any finite ranked set and $k \geq 0$. We call $\mathcal{V}$ a *variety of tree languages*, or a *tree language variety*, if it is closed under the Boolean operations, inverse morphisms between free preclones generated by finite ranked sets, and quotients defined as follows. Let $L \subseteq \Sigma M_k$ be a tree language and let (u, k_1, k_2, v) be an m-ary context in ΣM_k. Then the *left quotient* $(u, k_1, k_2)^{-1}L$ and the *right quotient* Lv^{-1} are defined by

$$(u, k_1, k_2)^{-1} = \{t \in \Sigma M_n \mid k_1 + n + k_2 = k,\ u \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2) \in L\}$$
$$Lv^{-1} = \{t \in \Sigma M_m \mid t \cdot v \in L\}.$$

A *literal variety of tree languages* is defined similarly, but instead of closure under inverse morphisms between finitely generated free preclones we require closure under inverse morphisms between finitely generated free gp's. Thus, if $L \subseteq \Sigma M_k$ is in a literal variety $\mathcal{V}$ and $h : \Delta M \to \Sigma M$ is a preclone morphism with $h(\Delta) \subseteq \Sigma$, where Δ is finite, then $h^{-1}(L)$ is also in $\mathcal{V}$.

Below we present an Eilenberg correspondence between pseudovarieties of preclones (gp's respectively), and varieties (literal varieties) of tree languages. For each pseudovariety $\mathbf{V}$ of preclones (resp. gp's), let $\mathcal{V}$ denote the class of those tree languages $L \subseteq \Sigma M_k$, where Σ is any ranked alphabet and $k \geq 0$, whose syntactic preclone (syntactic gp, resp.) belongs to $\mathbf{V}$.

Theorem 7. *The correspondence $\mathbf{V} \mapsto \mathcal{V}$ defines an order isomorphism between pseudovarities of preclones (gp's, resp.) and tree language varieties (literal varieties of tree languages, resp.).*

Remark 8. Two further variety theorems for finite trees exist in the literature. One uses minimal tree automata as syntactic algebra [1, 27], and the other uses syntactic Lawvere theories, i.e. clones [14]. No variety theorem is known for the 3-sorted algebras proposed in [31].

4 Logically defined tree languages

Let Σ be a ranked alphabet. We will define subsets of ΣM_k by means of logical formulas. Our atomic formulas are of the form

$$P_\sigma(x),\ x < x',\ \mathbf{Succ}_i(x,x'),\ \mathbf{left}_j(x) \text{ and } \mathbf{right}_j(x)$$

where $\sigma \in \Sigma$, i, j are positive integers, i is less than or equal to the maximal rank of a letter in Σ, and x, x' are first-order variables. If k is an integer, subsets of ΣM_k will be defined by *formulas of rank* k, composed using atomic formulas (with $j \in [k]$), the Boolean constants false and true, the Boolean connectives and a family of generalized quantifiers called *Lindström quantifiers*, defined below.

When a formula is interpreted on a tree $t \in \Sigma M_k$, first-order variables are interpreted as vertices of t, $P_\sigma(x)$ holds if x is labeled σ ($\sigma \in \Sigma$), $x < x'$ holds if x' is a proper descendant of x, and $\mathbf{Succ}_i(x,x')$ holds if x' is the i-th successor of x. Finally, $\mathbf{left}_j(x)$ holds (resp. $\mathbf{right}_j(x)$ holds) if the index of the highest numbered variable labeling a leaf to the left (resp. right) of the frontier of the subtree rooted at x is j.

We now proceed with the definition of (simple) Lindstöm quantifiers, adapted from [19, 10] to the case of finite trees. Let Δ be a ranked alphabet containing letters of rank m for each m such that $\Sigma_m \neq \emptyset$, and let $K \subseteq \Delta M_k$. Let x be a first-order variable. We describe the interpretation of the quantified formula $Q_K x.\langle\varphi_\delta\rangle_{\delta\in\Delta}$, where the quantifier Q_K binds the variable x – here $\langle\varphi_\delta\rangle_{\delta\in\Delta}$ is a family of (previously defined) formulas on Σ-trees which is *deterministic w.r.t.* x. We may assume that x is not bound in the φ_δ. Deterministic means that for each $t \in \Sigma M_k$, for each m such that $\Delta_m \neq \emptyset$, for each interpretation λ of the free variables in the φ_δ mapping x to a vertex of t labeled in Σ_m, then (t,λ) satisfies exactly one of the φ_δ, $\delta \in \Delta_m$.

Given this family $\langle\varphi_\delta\rangle$, a tree $t \in \Sigma M_k$ and an interpretation λ of the free variables in the φ_δ except for x, we construct a tree $\overline{t}_\lambda \in \Delta M_k$ as follows: the underlying tree structure of $\overline{t}_\lambda$ is the same as that of t, and the vertices labeled v_j ($j \in [k]$) are the same in both trees. For each vertex v of t labeled by $\sigma \in \Sigma_m$, let λ' be the interpretation obtained from λ by mapping variable x to vertex v: then the same vertex v in $\overline{t}_\lambda$ is labeled by the element $\delta \in \Delta_m$ such that (t,λ') satisfies φ_δ.

Finally, we say that (t,λ) satisfies $Q_K x \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$ if $\overline{t}_\lambda \in K$.

Example 9. Let Δ be a ranked alphabet such that each Δ_n is either empty or equal to $\{0_n, 1_n\}$ (such an alphabet is called *Boolean*), and let $k \geq 0$.

(1) Let $K = K(\exists)$ denote the (recognizable) language of all trees in ΔM_k containing at least one vertex labeled 1_n (for some n). Then the Lindström quantifier corresponding to K is a sort of existential quantifier. More precisely, let $(\varphi_\delta)_{\delta\in\Delta}$ be a collection of formulas: let us write φ_n for φ_{1_n} and note that φ_{0_n} is equivalent to $\neg\varphi_n$. Now let $t \in \Sigma M_k$ and let λ be an interpretation of the free variables in the φ_δ except for x. Then (t,λ) satisfies $Q_K x \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$ if and only if, for some n, φ_n is satisfied by (t,λ') for some extension λ' of λ which maps variable x to a vertex of rank n.

For instance, if Σ consists only of constants and one symbol of rank 2 and if $\varphi_0 = \mathsf{false}$, then (t, λ) satisfies $Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$ if and only if φ_2 is satisfied by some (t, λ') where λ' extends λ by mapping variable x to a vertex of rank 2 of t. In particular, if x is the only free variable in the φ_δ, then t satisfies $Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$ if and only if there exists x, a vertex of rank 2 of t which satisfies $\varphi_2(x)$.

(2) In the same manner, if $p \geq 1$, $r < p$ and $K = K(\exists_p^r)$ denotes the (recognizable) language of those trees in ΔM_k such that the number of vertices labeled 1_n (for some n) is congruent to r modulo p, then the Lindström quantifier Q_K is a modular quantifier.

(3) Let $K = K(\exists_{\text{path}})$ be the set of all trees in ΔM_k such that all the inner vertices along at least one maximal path from the root to a leaf are labeled 1_n (for the appropriate n). Then Q_K is an existential path quantifier.

(4) Let K_{next} denote the collection of all trees in ΔM_k such that each maximal path has length at least three and the vertices on the second level are labeled 1_n (for the appropriate n). Then K_{next} is a sort of next modality. Other next modalities can be expressed likewise.

For a class $\mathcal{K}$ of tree languages, we let $\mathbf{Lind}(\mathcal{K})$ denote the logic defined above, equipped with Lindström quantifiers associated to the languages in $\mathcal{K}$, and we let $\mathcal{L}ind(\mathcal{K})$ denote the class of $\mathbf{Lind}(\mathcal{K})$-*definable* tree languages: a language $L \subseteq \Sigma M_k$ belongs to $\mathcal{L}ind(\mathcal{K})$ iff there is a sentence φ of rank k over Σ all of whose Lindström quantifiers are associated to languages in $\mathcal{K}$ such that L is the set of those trees $t \in \Sigma M_k$ that satisfy φ.

Example 10. Let $\mathcal{K}_\exists$ be the class of all the languages of the form $K(\exists)$ on a Boolean ranked alphabet (see Example 9 (1)). One can verify that $\mathcal{L}ind(\mathcal{K}_\exists)$ is exactly the class of FO-definable tree languages. And when $\mathcal{K}_{\exists,\text{mod}}$ is the class of all languages of the form $K(\exists)$ or $K(\exists_p^r)$, then $\mathcal{L}ind(\mathcal{K}_{\exists,\text{mod}})$ is the class of $(FO + MOD)$-definable tree languages.

Theorem 11. *Let $\mathcal{K}$ be a class of tree languages.*
• $\mathcal{K} \subseteq \mathcal{L}ind(\mathcal{K})$, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \Rightarrow \mathcal{L}ind(\mathcal{K}_1) \subseteq \mathcal{L}ind(\mathcal{K}_2)$ and $\mathcal{L}ind(\mathcal{L}ind(\mathcal{K})) = \mathcal{L}ind(\mathcal{K})$ (that is, $\mathcal{L}ind$ is a closure operator).
• $\mathcal{L}ind(\mathcal{K})$ is closed under the Boolean operations and inverse morphisms between finitely generated free gp's. It is closed under quotients iff any quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$.

It will follow from our main result that if $\mathcal{K}$ consists of recognizable tree languages, then so does $\mathcal{L}ind(\mathcal{K})$. This can also be proved directly by expressing the Lindström quantifiers associated to the languages in $\mathcal{K}$ in monadic second-order logic.

Corollary 12. *Let $\mathcal{K}$ be a class of recognizable tree languages. Then $\mathcal{L}ind(\mathcal{K})$ is a literal variety iff any quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$ (e.g. if $\mathcal{K}$ is closed under quotients).*

$\mathcal{L}ind(\mathcal{K})$ is a variety iff any quotient and any inverse image of a language in $\mathcal{K}$ under a morphism between finitely generated free preclones belongs to $\mathcal{L}ind(\mathcal{K})$.

5 Algebraic characterization of $\mathcal{L}ind(\mathcal{K})$

5.1 Block product

The block product of monoids was introduced in [25] as a two sided generalization of the wreath product [11]. It is closely related to Eilenberg's bimachines and triple products [11]. Block products of monoids have been used extensively in [28] to obtain algebraic characterizations of the expressive power of certain logics on finite words. In this section we extend this operation to preclones and gp's.

Let S, T be preclones and $k \geq 0$. For each $m \geq 0$, let $I_{k,m}$ be the set of all m-ary contexts in T_k (see Section 3). Then for each $m \geq 0$, we let $(S \,\square_k\, T)_m = S_m^{I_{k,m}} \times T_m$. This defines the carriers of the block product $S \,\square_k\, T$. The operation of composition is defined as follows. For simplicity, we denote an element (u, k_1, k_2, v) of $I_{k,m}$ by just (u, v), where it is understood that u comes with a splitting of its argument sequence as $k_1 + 1 + k_2$. Let $(F, g) \in (S \,\square_k\, T)_n$, let $(F_i, g_i) \in (S \,\square_k\, T)_{m_i}$ for each $i \in [n]$ and let $m = \sum_{i=1}^{n} m_i$. Then we let

$$(F, g) \cdot ((F_1, g_1) \oplus \cdots \oplus (F_n, g_n))$$

be the element $(F', g \cdot (g_1 \oplus \cdots \oplus g_n))$ of $(S \square_k T)_m$ such that, for each $(u, v) \in I_{k,m}$ (using the notation of Section 3),

$$F'(u, v) = F(u,\ g_1 \cdot \overline{v}_1 \oplus \cdots \oplus g_n \cdot \overline{v}_n) \cdot \Big[F_1(u_1, \overline{v}_1) \oplus \cdots \oplus F_n(u_n, \overline{v}_n)\Big],$$

where $\overline{v}_1$ denotes the $\oplus$-sum of the first m_1 v_i's, $\overline{v}_2$ denotes the $\oplus$-sum of the next m_2 v_i's, etc, and where for each $1 \leq i \leq n$,

$$u_i = u \cdot (\mathbf{k}_1 \oplus g \cdot (g_1 \cdot \overline{v}_1 \oplus \cdots \oplus g_{i-1} \cdot \overline{v}_{i-1} \oplus \mathbf{1} \oplus g_{i+1} \cdot \overline{v}_{i+1} \oplus \cdots \oplus g_n \cdot \overline{v}_n) \oplus \mathbf{k}_2).$$

If we let $\overline{\ell}_1$ denote the sum of the first m_1 ℓ_i's, $\overline{\ell}_2$ denotes the sum of the next m_2 ℓ_i's, etc, one can verify that $(u_i, \overline{v}_i) \in I_{k,m_i}$, where the argument sequence of u_i is split as $k = (k_1 + \sum_{j<i} \overline{\ell}_j) + 1 + (\sum_{j>i} \overline{\ell}_j + k_2)$. It is long but routine to verify the following.

Proposition 13. *$S \,\square_k\, T$ is a preclone.*

When (S, A) and (T, B) are gp's and $k \geq 0$, we define the block product $(S, A) \,\square_k\, (T, B)$ as the gp (R, C), where C is the collection of all pairs (F, b) in $S \,\square_k\, T$ such that $b \in B$ and $F(u, v) \in A$, for all appropriate u, v, and where R is the sub-preclone generated by C in $S \,\square_k\, T$.

5.2 Main theorem

We need one more technical definition before stating the main result. Let $\mathcal{K}$ be a class of tree languages. We say that $\mathbf{Lind}(\mathcal{K})$ *admits relativization* if, for each sentence φ of $\mathbf{Lind}(\mathcal{K})$, each integer $i \geq 1$ and each first-order variable x, there exist formulas $\varphi[\geq_i x]$ and $\varphi[\not> x]$, with x as sole free variable, such that:

• $\varphi[\geq_i x]$ holds on t if the subtree of t whose root is the i-th child of vertex x satisfies φ;
• $\varphi[\not> x]$ holds on t if the tree r obtained from t by deleting the subtree rooted at x and relabeling that vertex with a variable, satisfies φ.

Formally, we require the following:
• given φ, a sentence of rank ℓ, and integers k_1, k_2, there exists a formula $\varphi[\geq_i x]$ of rank $k = k_1 + \ell + k_2$ such that, if $t \in \Sigma M_k$, the label of vertex w of t is in Σ_m with $m \geq i$, $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ for some $s \in \Sigma M_\ell$, and the root of s is the i-th successor of vertex w in t, then $(t, x \mapsto w) \models \varphi[\geq_i x]$ if and only if $s \models \varphi$;
• given integers k_1, k_2, a sentence φ of rank k_1+1+k_2 and an integer $k \geq k_1+k_2$ there exists a formula $\varphi[\not> x]$ of rank k such that, if $t \in \Sigma M_k$, $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ and vertex w of t is the root of the subtree s, then $(t, x \mapsto w) \models \varphi[\not> x]$ if and only if $r \models \varphi$.

We now state our main theorem, which extends the main result of [15] (relating to finite words) to finite trees.

Theorem 14. *Let $\mathcal{K}$ be a class of recognizable tree languages such that any quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$ and such that $\mathbf{Lind}(\mathcal{K})$ admits relativization. Then a language is in $\mathcal{L}ind(\mathcal{K})$ iff its syntactic gp belongs to the least pseudovariety of finitary gp's containing the syntactic gp's of the languages in $\mathcal{K}$ and closed under block products.*

5.3 Applications

The class of all recognizable tree languages is closed under taking quotients and one verifies easily that the corresponding logic admits relativizations. It follows that:

Corollary 15. *If $\mathcal{K}$ consists of recognizable languages, then so does $\mathcal{L}ind(\mathcal{K})$.*

By Example 10, the class of FO-definable tree languages is $\mathcal{L}ind(\mathcal{K}_\exists)$. One can verify that $\mathbf{Lind}(\mathcal{K}_\exists)$ admits relativization, and any quotient of a language in $\mathcal{K}_\exists$ belongs to $\mathcal{L}ind(\mathcal{K}_\exists)$. In order to use Theorem 14, we need to compute the syntactic gp's of the tree languages in $\mathcal{K}_\exists$. Let Δ be a Boolean alphabet, $k \geq 0$ and $K \subseteq \Delta M_k$ be as in Example 9 (1). It is not difficult to verify that the syntactic Δ-algebra of K has two elements, say $B = \{\mathsf{true}, \mathsf{false}\}$, and if $\Delta_n \neq \emptyset$, then 1_n is the constant function true and 0_n is the n-ary or function. By Proposition 6, the syntactic gp of K is the pair (T, Δ) where T is the sub-preclone of $T(B)$ generated by Δ.

Now let $T_\exists$ be the sub-preclone of $T(B)$ generated by the binary or function and the nullary constants true and false: then $(T_\exists)_n$ consists of the n-ary or function and the n-ary constant true. One can verify that no proper sub-preclone of $T_\exists$ contains the nullary constants true and false, and some n-ary or function ($n \geq 2$). Since we are assuming that $\Delta_n \neq \emptyset$ for some $n \geq 2$, it follows that the syntactic gp of K is a pair $(T_\exists, \Delta)$.

Next let $\mathbf{K}_\exists$ be the class of gp's whose underlying preclone is isomorphic to $T_\exists$. By Theorem 14, a tree language L is FO-definable if and only if its syntactic

gp lies in the least pseudovariety of gp's containing $\mathbf{K}_\exists$ and closed under block product. Next one verifies that a gp belongs to this pseudovariety if and only its underlying preclone lies in the least pseudovariety of preclones containing $T_\exists$ and closed under block product. Finally, we get the following result.

Corollary 16. *A tree language is FO-definable iff its syntactic preclone belongs to the least pseudovariety containing $T_\exists$ and closed under block product.*

Let $p \geq 2$ and let $B_p = \{0, 1, \ldots, p-1\}$. Let T_p be the sub-preclone of $T(B_p)$ whose rank n elements ($n \geq 0$) consists of the mappings $f_{n,r}\colon (r_1, \ldots, r_n) \mapsto r_1 + \cdots + r_n + r \bmod p$ for $0 \leq r < p$. By a reasoning similar to that used for Corollary 16, we can show the following.

Corollary 17. *A tree language is FO+MOD-definable iff its syntactic preclone belongs to the least pseudovariety containing $T_\exists$ and the T_p and closed under block product.*

Conclusion

We reduced the characterization of the expressive power of certain naturally defined logics on tree languages, a chief example of which is given by first-order sentences, to an algebraic problem.

For this purpose, we introduced a new algebraic framework to discuss tree languages. However, the resulting notion of recognizability coincides with the usual one: we simply gave ourselves a richer algebraic set-up to classify recognizable tree languages. This does not yield directly a decidability result for, say, first-order definable tree languages, but we can now look for a solution of this problem based on the methods of algebra. In this process, it will probably be necessary to develop the structure theory of preclones, to get more precise results on the block product operation.

A positive aspect of our approach is its generality: it is not restricted to the characterization of logics based on the use of Lindström quantifiers (nor indeed to the characterization of logics). For instance, the use of wreath products instead of block products, will yield algebraic characterizations for other natural classes of recognizable tree languages.

References

1. J. Almeida, On pseudovarieties, varieties of languages, filters of congruences, pseudoidentities and related topics, *Algebra Universalis*, 27(1990), 333–350.
2. Arnold and M. Dauchet, Theorie des magmoides. I. and II. (in French), *RAIRO Theoretical Informatics and Applications*, 12(1978), 235–257, 3(1979), 135–154.
3. S. L. Bloom and Z. Ésik, *Iteration Theories*, Springer, 1993.
4. J. R. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundlagen Math.*, 6(1960), 66–92.
5. J. Cohen, J.-E. Pin and D. Perrin, On the expressive power of temporal logic, *J. Computer and System Sciences*, 46(1993), 271–294.

6. B. Courcelle, The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, *Information and Computation*, 85(1990), 12–75.
7. K. Denecke and S. L. Wismath, *Universal Algebra and Applications in Theoretical Computer Science*, Chapman and Hall, 2002.
8. V. Diekert, Combinatorics on Traces, LNCS 454, Springer, 1987.
9. J. Doner, Tree acceptors and some of their applications, *J. Comput. System Sci.*, 4(1970), 406–451.
10. H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer, 1995.
11. S. Eilenberg, *Automata, Languages, and Machines*, vol. A and B, Academic Press, 1974 and 1976.
12. S. Eilenberg and J. B. Wright, Automata in general algebras. *Information and Control*, 11(1967), 452–470.
13. C. C. Elgot, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.*, 98(1961), 21–51.
14. Z. Ésik, A variety theorem for trees and theories, *Publicationes Mathematicae*, 54(1999), 711–762.
15. Z. Ésik and K. G. Larsen, Regular languages definable by Lindström quantifiers, *Theoretical Informatics and Applications*, to appear.
16. D. M. Gabbay, A. Pnueli, S. Shelah and J. Stavi, On the temporal analysis of fairness, in: proc. *12th ACM Symp. Principles of Programming Languages*, Las Vegas, 1980, 163–173.
17. U. Heuter, First-order properties of trees, star-free expressions, and aperiodicity, in: *STACS 88*, LNCS 294, Springer, 1988, 136–148.
18. J. A. Kamp, Tense logic and the theory of linear order, Ph. D. Thesis, UCLA, 1968.
19. P. Lindström: First order predicate logic with generalized quantifiers. *Theoria*, 32(1966), 186–195.
20. S. MacLane, *Categories for the Working Mathematician*, Springer, 1971.
21. R. McNaughton and S. Papert, *Counter-Free Automata*, MIT Press, 1971.
22. J. Mezei and J. B. Wright, Algebraic automata and context-free sets, *Information and Control*, 11(1967), 3–29.
23. A. Potthoff, Modulo counting quantifiers over finite trees, in: *CAAP '92*, LNCS 581, Springer, 1992, 265–278.
24. A. Potthoff, First order logic on finite trees, in: *TAPSOFT '95*, LNCS 915, Springer, 1995.
25. J. Rhodes and B. Tilson: The kernel of monoid morphisms, *J. Pure and Appl. Alg.*, 62(1989), 227–268.
26. M. P. Schützenberger, On finite monoids having only trivial subgroups. *Information and Control*, 8(1965), 190–194.
27. M. Steinby, General varieties of tree languages. *Theoret. Comput. Sci.*, 205(1998), 1–43.
28. H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity,* Birkhauser Boston, Inc., Boston, MA, 1994.
29. H. Straubing, D. Therien and W. Thomas, Regular languages defined with generalized quantifiers, *Information and Computation*, 118(1995), 289–301.
30. J. W. Thatcher and J. B. Wright, Generalized finite automata theory with an application to a decision problem of second-order logic. Math. Systems Theory, 2(1968), 57–81.
31. Th. Wilke, An algebraic characterization of frontier testable tree languages, *Theoret. Comput. Sci.*, 154(1996), 85–106.