Algebraic characterization of logically defined tree languages

Zoltan Esik, Pascal Weil

Algebraic characterization of logically defined tree languages*

Zoltán Ésik
GRLMC, Rovira i Virgili University, Tarragona, Spain
and
Department of Computer Science, University of Szeged, Hungary

Pascal Weil †
LaBRI, Université de Bordeaux, CNRS

Abstract

We give an algebraic characterization of the tree languages that are defined by logical formulas using certain Lindström quantifiers. An important instance of our result concerns first-order definable tree languages. Our characterization relies on the usage of preclones, an algebraic structure introduced by the authors in a previous paper, and of the block product operation on preclones. Our results generalize analogous results on finite word languages, but it must be noted that, as they stand, they do not yield an algorithm to decide whether a given regular tree language is first-order definable.

Classification: ACM: F.4.3, F.4.1. MSC: 03B70, 68Q70, 68Q45

One of Bret Tilson's lasting contributions is the introduction (with John Rhodes) of the notions of block product and two-sided semidirect product, and their use in the structure theory of finite monoids. This tool was initially introduced to derive iterated decompositions of morphisms and to refine the wreath product-based Krohn-Rhodes decomposition of finite monoids [28]. It quickly found applications in formal language theory (see [29, 30, 38, 2, 35] among others). One of the more fruitful applications of this work has been in the investigation of the logical aspects of automata theory (on finite words). For

*The first author acknowledges partial support from grant MTM2007 63422 from the Ministry of Education and Science of Spain. The second author acknowledges partial support from the French *ANR* (projet DOTS) and the Indo-French project TIMED DISCOVERI. Both authors acknowledge support from the European Science Foundation program AUTOMATHA.

†Corresponding author. pascal.weil@labri.fr. Postal address: LaBRI, 351 cours de la Libération, 33405 Talence Cedex, France.

instance, the expressive power of first-order formulas with a certain quantifier depth, can be captured by monoids which divide an iterated block product of semilattices of the same length, see Straubing's book [35].

Automata are particularly well suited to discuss the behavior of terminating sequential systems (languages of finite words), and this field of research has benefited from the start from the well-established connection between automata and monoid theory. There is however also much interest in automata-theoretic descriptions of languages of other structures than finite words, corresponding to other natural ideas of implementation and other natural models of computation (infinite, branching, concurrent, timed, etc). This paper is a contribution to the investigation of an important problem of this sort: can we decide whether a regular tree language is first-order definable? Here trees are finite, ranked and ordered. The latter properties signify that the nodes of the trees are labeled with symbols of a given arity (the rank of the node), and the children of a node of rank r form a totally ordered set of cardinality r. A tree language is said to be regular if it is accepted by a classical (deterministic) bottom-up tree automaton.

The notion of automata recognizability for (finite) word languages is easily translated to an algebraic notion of recognizability, expressed in terms of monoids: the set of all words on a given alphabet A is a monoid (the free monoid A^* over that alphabet), and one shows that a language is recognized by a finite state automaton if and only if it is the inverse image of a set, under a morphism from A^* into a *finite* monoid. Moreover, if a language is recognizable, then there is a least finite monoid recognizing it, called its syntactic monoid. This point of view opens vast possibilities for the classification and the discussion of the properties of recognizable languages, which can be characterized in terms of the algebraic properties of their syntactic monoid, see Eilenberg's variety theory [11, 22, 1]. As the syntactic monoid of the language accepted by a given automaton is computable, this can lead to interesting decision algorithms.

It is well-known (Büchi, 1960), that a language is recognizable if and only if it is definable in monadic second-order logic. It was also shown that a language is definable by a first-order formula if and only if its syntactic monoid is aperiodic. This statement is actually the combination of two classical theorems due to Schützenberger and to McNaughton and Papert. It can also be proved directly (as in [35]), using the Krohn-Rhodes decomposition theorem, which implies that a monoid is aperiodic if and only if it belongs to the least pseudovariety closed under block product and containing $U_1 = \{1, 0\}$. As the syntactic monoid of a regular language is computable and as aperiodicity is decidable, it is also decidable whether a regular language is FO-definable.

Considering logically defined sets of trees (or other discrete combinatorial structures) is just as natural as for words. But the literature on these questions shows that classification and decision results are much harder to reach, in part because we lack the versatile and powerful algebraic tool provided in the word case by finite monoid theory. The weakness of our understanding of automata theory for tree languages is highlighted by the fact that the decidability of first-order definability is still an open problem.

For most discrete structures, there is no obvious algebraic structure that can be used in lieu of monoids, or at least no algebraic structure that gives rise to the same wealth of structure theorems and variety characterizations, see [39]. In the tree case (for finite, ranked and ordered trees), several propositions can be found in the literature, see the work of Steinby, Salehi, Heuter, Podelski, Wilke, Ésik, etc [33, 34, 31, 32, 19, 25, 40, 12]. Until recently none was very convincing in terms of its capacity to characterize significant classes of languages, but there are recent encouraging results expressed in terms of minimal tree automata, that is, in terms of Σ-algebras by Benedikt and Ségoufin [3], Bojańczyk and Walukiewicz [7], Ésik [13] and Ésik and Iván [14, 15][1]. In a previous paper [18], the authors introduce a new algebraic framework – the so-called *preclones* – to classify and discuss the properties of recognizable tree languages.

It turns out that the setting of preclones makes it natural to discuss not only the recognizable sets of trees, but also recognizable sets of trees with variables. Variables can be seen as unlabeled leaves of the tree, and the rank of a tree is the number of such unlabeled leaves. Alternately, one can regard these leaves as labeled by particular letters $\{v_1, v_2, \ldots\}$, in such a way that in a rank k tree, the variable leaves are labeled $v_1, \ldots, v_k$ from left to right.

We verified in [18] that the notion of recognizability induced by the algebraic structure of preclones coincides with the usual notion of recognizable tree languages, that the syntactic preclone of a recognizable tree language is completely determined by the minimal deterministic bottom-up automaton of the language (all very reassuring facts), and that these notions are robust enough to allow for an Eilenberg-like development in terms of varieties of languages and pseudovarieties of preclones.

In this paper, we use this algebraic framework to derive an algebraic characterization of first-order definable tree languages, and more generally, of the classes of tree languages determined by certain families of Lindström quantifiers. This characterization requires the introduction of a block product operation on preclones, a complex algebraic operation which generalizes Tilson's block product of monoids. Our main result implies that the first-order definable tree languages are exactly those languages whose syntactic preclone sits in the least pseudovariety of preclones closed under block product and containing $T_\exists$, a very simple preclone whose properties were discussed in [18] and which can be viewed as an analogue of the monoid U_1. This result was announced without proof in the authors' communication at FST-TCS [17].

As it is, our result does not yield an algorithm to decide whether a given recognizable tree language is first-order definable. This question is briefly discussed in the conclusion of the paper, but whatever the case may be, such a decidability result remains one of the main goals in this field. Our result however suggests the feasibility of an algebra-based solution.

The plan of this paper is as follows. Section 1 summarizes the essential properties of preclones that are necessary for this study. Section 2 introduces

[1] After our results were announced in 2003 [17] and while this paper was in preparation or under refereeing, an interesting approach of tree languages in terms of so-called *forest algebras* was introduced by Bojańczyk and Walukiewicz. See the conclusion of this paper for a brief discussion.

the logical apparatus we will use, including the Lindström quantifiers and the closure properties of the associated operators on classes of languages. Finally, in Section 3, we introduce the block product operation on preclones, and we prove our main results. The paper closes on a conclusion where we discuss certain questions raised by these results.

Notation Let $n \geq 0$. We denote by $[n]$ the set $\{1, \ldots, n\}$ if $n > 0$, the empty set if $n = 0$.

1 The algebraic framework

Throughout the paper, we will be discussing sets of finite ranked trees, that is, trees in which the set of children of each inner node is linearly ordered; Σ designates a *ranked alphabet*, that is, $\Sigma = (\Sigma_n)_{n\geq 0}$ where the Σ_n are pairwise disjoint sets and $\bigcup_n \Sigma_n$ is finite. An element of Σ_n is said to have *rank* n.

This section summarizes the main facts relative to the algebraic framework, which we will use to establish our main theorem. Most of these results are taken from the authors' earlier paper [18] and are stated here without proof.

1.1 Preclones

A *preclone* is a many-sorted algebra $S = ((S_n)_{n\geq 0}, \bullet, \mathbf{1})$. The elements of S_n are said to have *rank* n, the element $\mathbf{1}$ belongs to S_1, and the composition operation $\bullet$ associates with each $f \in S_n$ and n-tuple $g = (g_1, \ldots, g_n)$ (with $g_i \in S_{m_i}$, $1 \leq i \leq n$), an element $f \bullet g \in S_m$ where $m = \sum_i m_i$. Moreover, $\mathbf{1}$ and $\bullet$ satisfy the axioms given below.

For convenience, a tuple g as above is written $g = g_1 \oplus \cdots \oplus g_n$, we say that g has *total rank* m, written $\mathsf{rank}(g) = m$, and we let $S_{n,m}$ be the set of n-tuples of total rank m. Note that $S_{1,m} = S_m$ for all m. We also write $\mathbf{n}$ for the $\oplus$-sum of n copies of $\mathbf{1}$, so that $\mathbf{n} \in S_{n,n}$. The axioms defining preclones are the following:

$$\begin{aligned} \mathbf{1} \cdot f \quad &= \quad f \quad = \quad f \cdot \mathbf{n} \quad \text{for each } f \in S_n,\ n \geq 0, \\ \text{and } (f \cdot g) \cdot h \quad &= \quad f \cdot (g_1 \cdot \bar{h}_1 \oplus \cdots \oplus g_n \cdot \bar{h}_n) \end{aligned}$$

where $f \in S_n$, $g = \bigoplus_{i=1}^n g_i$ with each $g_i \in S_{m_i}$, $h = \bigoplus_{j=1}^m h_j$ with each $h_j \in S_{p_j}$, $m = \sum_i m_i$, $\bar{h}_1$ equal to the $\oplus$-sum of the m_1 first h_j's, $\bar{h}_2$ equal to the $\oplus$-sum of the m_2 next h_j's, etc.

It is interesting to remark that S_1 is naturally equipped with a monoid structure.

Sub-preclones of preclones are defined in the natural way. A *morphism of preclones*, $\varphi\colon S \to T$, is a rank preserving map, which also preserves the unit element $\mathbf{1}$ and the composition operation. Similarly, a congruence is an equivalence relation, that relates only elements of equal rank, and which is stable under the composition operation. The quotient of a preclone by a congruence is naturally endowed with a preclone structure, and the projection map is an onto morphism.

The least sub-preclone of a preclone S, containing a given subset A is called *the sub-preclone of* S *generated by* A. If this preclone is S itself, we say that S is *generated* by A. A preclone is *finitely generated* if it admits a finite set of generators.

A preclone S is said to be *finitary* if each S_n is finite. Observe that as soon as some S_k, $k \geq 2$ is non-empty, then infinitely many S_k are non-empty, and hence S is not finite.

1.2 Examples of preclones

The following examples will be essential for our study.

Trees and free preclones Let Σ be a ranked alphabet. The free preclone generated by Σ, written ΣM, can be described as follows (see [18, Section 2.2]). 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 (where a rank k letter labels a node with k children), whose leaves are labeled by elements of $\Sigma_0 \cup \{v_1, \ldots, v_n\}$, and whose *frontier* (the left to right sequence of leaf labels) contains exactly one occurrence of v_1, …, v_n, in that order (that is, belongs to $\Sigma_0^* v_1 \Sigma_0^* \cdots v_n \Sigma_0^*$). The elements of ΣM_n are called *trees of rank* n or n*-ary trees* over Σ.

If t is such a tree, we let $\mathsf{NV}(t)$ be the set of nodes of t with a label in Σ (NV stands for *non-variable labeled*). In the logical discussion to follow (Section 2), we will give the nodes in $\mathsf{NV}(t)$ a particular rôle.

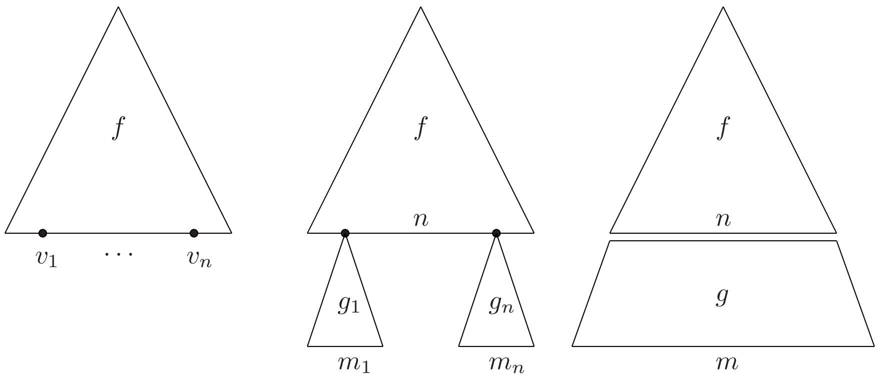

Figure 1: $f \in \Sigma M_n$, $g = g_1 \oplus \cdots \oplus g_n \in \Sigma M_{n,m}$ and two views of $f \cdot g$

ΣM is a preclone for the following operations: if $f \in \Sigma M_n$ and $g_i \in \Sigma M_{m_i}$ $(1 \leq i \leq n)$, then the composite tree $f \cdot (g_1 \oplus \cdots \oplus g_n)$ is obtained by replacing the v_i-labeled leaf of f by the root of g_i, and by renumbering the variable labeled leaves of the resulting tree with consecutively indexed v_j's, see Figure 1. The unit element **1** is the graph in ΣM_1 consisting of a single node labeled v_1.

Each element of Σ can be identified with an element of ΣM: the letter $\sigma \in \Sigma_n$ is identified with the tree with $n+1$ nodes, consisting of the root, labeled σ, and the n children of the root, labeled $v_1, \ldots, v_n$ in this order.

The elements of rank 0, ΣM_0, are the ordinary Σ-labeled trees.

Example 1.1 As discussed in the introduction, our results can be seen as generalizations to trees of known results on recognizable word languages. This meta-statement can be made precise in the following fashion: if A is a finite (unranked) alphabet, we can view A as a ranked alphabet, all of whose elements have rank 1. Then the elements of AM_1 can be seen as the words of the form wv_1, where $w \in A^*$. In particular, the monoid AM_1 is isomorphic to, and will be identified with the free monoid A^*. The sets AM_n $(n \neq 1)$ are empty. □

Preclone of transformations, preclone of an automaton If Q is a set, let $\mathbb{T}_n(Q)$ be the set of n-ary transformations of Q, that is, the set of mappings $Q^n \to Q$. Let also $\mathbb{T}(Q) = (\mathbb{T}_n(Q))_{n \geq 0}$. Composition of mappings endows $\mathbb{T}(Q)$ with a preclone structure in the following sense: if $f \in \mathbb{T}_n(Q)$, $g_i \in \mathbb{T}_{m_i}(Q)$ $(1 \leq i \leq n)$ and $m = \sum_i m_i$, then $f \cdot (g_1 \oplus \cdots \oplus g_n)$ maps $(q_1, \ldots, q_m)$ to

$$f\big(g_1(q_1, \ldots, q_{m_1}), g_2(q_{m_1+1}, \ldots, q_{m_1+m_2}), \ldots, g_n(q_{m-m_n+1}, \ldots, q_m)\big).$$

If Σ is a ranked alphabet and Q is a Σ-algebra, each element $\sigma \in \Sigma_n$ determines naturally an n-ary transformation of Q. The sub-preclone of $\mathbb{T}(Q)$ generated by Σ is called the *preclone associated* with Q.

Example 1.2 Let A be an unranked alphabet, viewed as a ranked alphabet as in Example 1.1. An A-algebra Q is simply a set, equipped with an action of A, that is, a deterministic complete automaton. Each letter $a \in A$ then defines a mapping $Q \to Q$. Thus the preclone associated with Q has elements of rank 1 only, which form the usual transition monoid of the automaton, see [11, 22]. □

We note that Σ-algebras are natural objects in our context: a deterministic complete bottom-up tree automaton accepting trees in ΣM_0 (see [9]), with state set Q, can be described as a finite Σ-algebra Q, equipped with a set $F \subseteq Q$ of *final states*.

The preclones $T_\exists$ and T_p Let $\mathbb{B}$ be the Boolean semiring $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$, and let $T_\exists$ be the subset of $\mathbb{T}(\mathbb{B})$ whose rank n elements are the n-ary or function and the n-ary constant true, written respectively or_n and true_n (by convention, or_0 is the nullary constant false_0). Then $T_\exists$ is a preclone, which is generated by the binary or_2 function and the nullary constants true_0 and false_0. The set $\mathbb{B}$ equipped with these 3 generators can be viewed as a finite tree automaton, and $T_\exists$ is the preclone associated with this automaton.

It is interesting to note that the rank 1 elements of $T_\exists$ form a 2-element monoid, isomorphic to the multiplicative monoid $\{0,1\}$, and known as U_1 in the literature on monoid theory, e.g. [22].

Similarly, if $p \geq 2$ is an integer and $\mathbb{B}_p = \{0,1,\ldots,p-1\}$, let T_p be the subset of $\mathbb{T}(\mathbb{B}_p)$ whose rank n elements ($n \geq 0$) are the mappings $f_{n,r}\colon (r_1,\ldots,r_n) \mapsto r_1 + \cdots + r_n + r \bmod p$ for $0 \leq r < p$. Again, T_p is a finitely generated preclone, generated by the nullary constant 0, the unary increment function $f_{1,1}$ and the binary sum $f_{2,0}$. Moreover, T_p can be seen as the preclone associated with a p-element automaton, and its rank 1 elements form a monoid isomorphic to the cyclic group of order p.

Preclone-generator pairs If S is a preclone and A is a set of generators of S, we say that (S,A) is a *preclone-generator pair*, or *pg-pair*. A *pg*-pair (S,A) is said to be *finitary* if S is finitary and A is finite. The notions of sub-*pg*-pair and morphisms of *pg*-pairs are defined naturally: (S,A) is a *sub-pg-pair* of (T,B) is $A \subseteq B$ (so that S is by construction a sub-preclone of T); and a *morphism of pg-pairs* $\varphi\colon (S,A) \to (T,B)$ is a preclone morphism from S to T such that $\varphi(A) \subseteq B$.

1.3 Syntactic preclones

Let S be a preclone and let $L \subseteq S_k$. We say that L is *recognizable* if there exists a morphism $\varphi\colon S \to T$ into a finitary preclone and a subset $P \subseteq T_k$ such that $L = \varphi^{-1}(P)$. Then we say that L is *recognized by* T, and by the morphism φ. If (S,A) and (T,B) are *pg*-pairs and φ is a morphism between these *pg*-pairs, we say that L is recognized by (T,B).

Let $f \in S_n$. A *context of f in L* is a pair (u,v) where u is an element of S and v is an n-tuple of elements of S, such that f can be inserted under u and above v to produce an element of L, see Figure 2. We would like this condition to read $u \cdot f \cdot v \in L$, but it has to be a little more technical, to specify where precisely f is attached under u.

Formally, for each $k \geq 0$, $n > 0$, let $I_{k,n}$ be the set of *n-ary contexts in* S_k, that is, the set of tuples of the form (u,k_1,v,k_2), where $k_1,k_2 \geq 0$ and $k_1+k_2 \leq k$, $u \in S_{k_1+1+k_2}$ and $v \in S_{n,\ell}$ with $\ell = k-(k_1+k_2)$, see Figure 2. If $L \subseteq S_k$, $f \in S_n$, we say that a context $(u,k_1,v,k_2) \in I_{k,n}$ is an *L-context of f* if $u \cdot (\mathbf{k}_1 \oplus f \cdot v \oplus \mathbf{k}_2) \in L$. We also let the set of 0-ary contexts in S_k be the set $I_{k,0}$ of tuples $(u,k_1,\mathbf{0},k_2)$ where $k_1,k_2 \geq 0$ and $u \in S_{k_1+1+k_2}$ (the symbol $\mathbf{0}$ is introduced here to preserve the uniformity of notation). We say that such a context is an L-context of $f \in S_0$ if $u \cdot (\mathbf{k}_1 \oplus f \oplus \mathbf{k}_2) = u \cdot (\mathbf{k}_1 \oplus f \cdot \mathbf{0} \oplus \mathbf{k}_2) \in L$.

Next we say that elements of f,g are *L-equivalent*, written $f \sim_L g$ if f and g have the same L-contexts. The relation $\sim_L$ is a congruence, called the *syntactic congruence* of L, the quotient preclone $S/\sim_L$ is called the *syntactic preclone* of L, and the projection morphism is the *syntactic morphism*. Finally, if A is a set of generators of S, the *syntactic pg-pair* of L is the pair (T,B) where $T = S/\sim_L$ and B is the image of A in the syntactic morphism. We note the following result, proved in [18, Proposition 3.2].

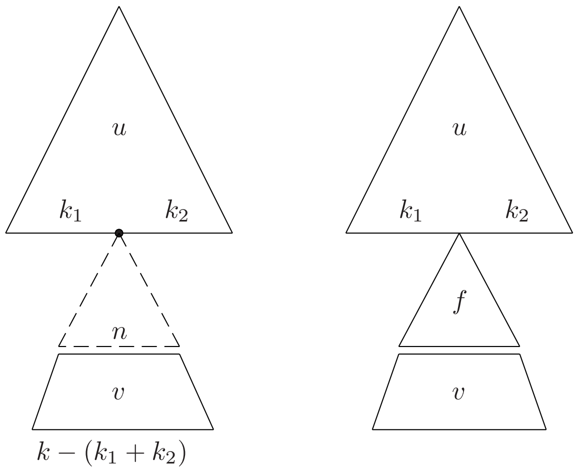

Figure 2: an n-ary context in ΣM_k; is it an L-context of f ($f \in \Sigma_n$)?

Proposition 1.3 *Let S be a preclone, $k \geq 0$ and $L \subseteq S_k$. Then the following statements hold.*

- *A morphism of preclones recognizes L if and only if it it can be factored through the syntactic morphism of L.*
- *If T is a sub-preclone or a quotient of T', and if T recognizes L, then so does T'.*
- *L is recognizable if and only if its syntactic preclone is finitary.*
- *The analogous statements hold for pg-pairs.*

We will be primarily concerned with the case where S is a finitely generated free preclone, $S = \Sigma M$ with Σ a ranked alphabet. The subsets of each ΣM_k are called *tree languages*. In that case, the notion of recognizable tree languages defined above coincides with the classical notion of recognizability, and in particular, the syntactic preclone of a recognizable tree language $L \subseteq \Sigma M_0$ coincides with the preclone of the minimal automaton of L, see [18, Section 3.2]. It is interesting to note that the syntactic Σ-algebra of L [34] is exactly the rank 0 part of the syntactic preclone of L, and that the syntactic tree monoid [25] of L is the monoid of rank 1 elements of its syntactic preclone. In particular, if $L \subseteq \Sigma M_0$, then the syntactic preclone of L is finitary if and only if its rank 0 part is finite.

We now consider two important examples. In each, the alphabet is a *ranked Boolean alphabet* Δ, that is, a ranked alphabet such that whenever $\Delta_n \neq \emptyset$, then $\Delta_n = \{1_n, 0_n\}$, see [18, Section 3.3] for more details.

Example 1.4 For $k \geq 0$, let $K_k(\exists)$ be the set of trees in ΔM_k containing at least one node labeled 1_n (for some n). Then $K_k(\exists)$ is recognizable and its syntactic preclone is $T_\exists$ defined in Section 1.2, see [18, Section 3.3]. More generally, let $\varphi\colon \Sigma M \to T_\exists$ be a morphism, with Σ an arbitrary ranked alphabet. Let

$$\Sigma^{(0)} = \bigcup_n \{\sigma \in \Sigma_n \mid \varphi(\sigma) = \mathsf{or}_n\}$$

and let $\Sigma^{(1)}$ be the complement of $\Sigma^{(0)}$ in Σ. Then the subsets of ΣM_k recognized by φ are $\emptyset$, ΣM_k, $\varphi^{-1}(\mathsf{or}_k) = \Sigma^{(0)} M_k$ and $\varphi^{-1}(\mathsf{true}_k)$, the set of trees in ΣM_k with at least one occurrence of a letter in $\Sigma^{(1)}$.

Similarly, if p, r are integers with $0 \leq r < p$ and if $K_k(\exists_p^r)$ consists of the trees in ΔM_k such that the number of nodes labeled 1_n (for some n) is congruent to r modulo p, then $K_k(\exists_p^r)$ is recognizable and its syntactic preclone is T_p. If $\varphi\colon \Sigma M \to T_p$ is a morphism, then the subsets of ΣM_k recognized by φ are the finite unions of the $\varphi^{-1}(f_{n,r})$ $(0 \leq r < p)$. For each such r, let

$$\Sigma^{(r)} = \bigcup_n \{\sigma \in \Sigma_n \mid \varphi(\sigma) = f_{n,r}\}.$$

For each $t \in \Sigma M_k$, let $w_r(t)$ be the number of nodes in $\mathsf{NV}(t)$ labeled by a letter in $\Sigma^{(r)}$, and let $w(t) = \sum_r r\, w_r(t)$. Then $\varphi^{-1}(f_{n,r})$ is the set of all $t \in \Sigma M_k$ such that $w(t) = r \pmod p$. □

1.4 Varieties of tree languages and pseudovarieties

A *pseudovariety of preclones* is a class of finitary preclones which is closed under taking finite direct products, sub-preclones, quotients, finitary unions of ω-chains and finitary inverse limits of ω-diagrams, see [18, Section 4]. Here, we say that a union $T = \bigcup_{n \geq 0} T^{(n)}$ is *finitary* if T is finitary. *Finitary inverse limits* of an ω-diagram of the form $\varphi^{(n)}\colon T^{(n+1)} \to T^{(n)}$ are defined similarly.

The definition of a *pseudovariety of pg-pairs* is similar: it is a class of finitary *pg*-pairs which is closed under taking finite direct products, sub-preclones, quotients, and finitary inverse limits of ω-diagrams (there is no need to consider unions of ω-diagrams, see [18, Section 4.4]).

We note the following proposition [18, Corollary 4.22].

Proposition 1.5 *Let* $\mathbf{V}$ *be a pseudovariety of preclones and let* S *be a finitary preclone such that, for all* $s \neq t \in S$*, there exists a morphism* φ *from* S *into a preclone in* $\mathbf{V}$ *with* $\varphi(s) \neq \varphi(t)$*. Then* $S \in \mathbf{V}$*.*

An analogous statement for pseudovarieties of *pg*-pairs also holds.

Proposition 1.6 *Let* $\mathbf{V}$ *be a pseudovariety of pg-pairs and let* (S, A) *be a finitary pg-pair such that, for all* $s \neq t \in S$*, there exists a morphism* φ *from* (S, A) *into a pg-pair in* $\mathbf{V}$ *with* $\varphi(s) \neq \varphi(t)$*. Then* $(S, A) \in \mathbf{V}$*.*

Proof. This statement is proved in the same fashion as Proposition 1.5 (see [18]), using also [18, Proposition 4.23]. □

If $\mathbf{K}$ is a class of finitary preclones (resp. *pg*-pairs), there exists a least pseudovariety containing $\mathbf{K}$, which is said to be *generated by* $\mathbf{K}$ and is denoted by $\langle \mathbf{K} \rangle$, see [18, Section 4.2]. We record in particular the following results, which follow from [18, Propositions 3.3 and 4.16, Corollary 4.8].

Proposition 1.7 *Let $\mathbf{K}$ be a class of finitary preclones (resp. pg-pairs) and let $\mathbf{V}$ be the pseudovariety generated by $\mathbf{K}$. The syntactic preclone (resp. pg-pair) of a recognizable tree language belongs to $\mathbf{V}$ if and only if it is a quotient of a sub-preclone (resp. sub-pg-pair) of a direct product of elements of $\mathbf{K}$.*

Proposition 1.8 *A pseudovariety of preclones (resp. pg-pairs) is entirely determined by the syntactic preclones (resp. pg-pairs) it contains.*

Note that pseudovarieties of preclones can be seen as particular examples of pseudovarieties of *pg*-pairs, in the sense of Proposition 1.9 below[2]. If $\mathbf{K}$ is a class of *pg*-pairs, we let $\mathsf{precl}(\mathbf{K})$ be the class of preclones S such that $(S, A) \in \mathbf{K}$ for some set A. Conversely, if $\mathbf{L}$ is a class of preclones, we let $\mathsf{pgp}(\mathbf{L})$ be the class of finitary *pg*-pairs (S, A) such that $S \in \mathbf{L}$. Let us say that a class $\mathbf{K}$ of *pg*-pairs is *full* if membership of a *pg*-pair (S, A) in $\mathbf{K}$ depends only on S; that is, $\mathbf{K} = \mathsf{pgp}(\mathsf{precl}(\mathbf{K}))$.

Proposition 1.9 *A pseudovariety $\mathbf{V}$ of pg-pairs is full if and only if there exists a pseudovariety $\mathbf{W}$ of preclones such that $\mathbf{V} = \mathsf{pgp}(\mathbf{W})$ and in that case, $\mathbf{W}$ is the pseudovariety generated by $\mathsf{precl}(\mathbf{V})$.*

Moreover, if $\mathbf{K}$ is a full class of pg-pairs and $\mathbf{V}$ is the pseudovariety generated by $\mathbf{K}$, then $\mathbf{V}$ is full as well and $\mathsf{precl}(\mathbf{K})$ and $\mathsf{precl}(\mathbf{V})$ generate the same pseudovariety of preclones.

Proof. Let $\mathbf{W}$ be a pseudovariety of preclones and let $\mathbf{V} = \mathsf{pgp}(\mathbf{W})$. The class $\mathbf{V}$ is full by definition. Let us first verify that it is closed under taking sub-*pg*-pairs, quotients, finite direct products and finitary inverse limits of ω-diagrams. Suppose for instance that (S, A) is a sub-*pg*-pair of (T, B) with $T \in \mathbf{W}$. Then S is a sub-preclone of T, so $S \in \mathbf{W}$ and $(S, A) \in \mathbf{V}$. The verification is equally routine for quotients and finite direct products. As for inverse limits of ω-diagrams, it was shown [18, Proposition 4.23] that if $(S, A) = \lim_n (S^{(n)}, A^{(n)})$, then $S = \lim_n S^{(n)}$. Thus, if the $(S^{(n)}, A^{(n)})$ are in $\mathbf{V}$, then the $S^{(n)}$ are in $\mathbf{W}$ and hence $S \in \mathbf{W}$ and $(S, A) \in \mathbf{V}$.

Now let us show that if $\mathbf{L}$ is a class of finitely generated finitary preclones, then $\mathsf{pgp}(\langle \mathbf{L} \rangle) \subseteq \langle \mathsf{pgp}(\mathbf{L}) \rangle$. Let (U, C) be a finitary *pg*-pair with $U \in \langle \mathbf{L} \rangle$: we want to show that $(U, C) \in \langle \mathsf{pgp}(\mathbf{L}) \rangle$. Combining technical results from [18] (namely Propositions 4.5 and 4.16), we may assume that there exist preclones $S^{(1)}, \ldots, S^{(n)}$ in $\mathbf{L}$ such that $U = \varphi(T')$ for some morphism $\varphi: T' \to U$ where T' is a sub-preclone of $\prod_i S^{(i)}$. Since $\mathbf{L}$ consists of finitely generated preclones, let $A^{(i)}$ be a finite set of generators of $S^{(i)}$. Let B be a finite subset of T' such that $\varphi(B) = C$ and for each i, let $B^{(i)}$ be the projection of B onto the i-th

[2]The notion of pseudovarieties of *pg*-pairs could be generalised to the notion of *varieties of stamps* as is done for word languages, see [36, 24].

component. Finally, let T be the sub-preclone of T' generated by B. Then $(U, C) = \varphi(T, B)$ and (T, B) is a sub-*pg*-pair of $\prod_i (S^{(i)}, A^{(i)} \cup B^{(i)})$. This establishes that $(U, C) \in \langle \mathsf{pgp}(\mathbf{L}) \rangle$.

Let now $\mathbf{K}$ be a full class of finitary *pg*-pairs and let $\mathbf{W} = \langle \mathsf{precl}(\mathbf{K}) \rangle$. We verify that $\langle \mathbf{K} \rangle = \mathsf{pgp}(\mathbf{W})$, which implies that $\langle \mathbf{K} \rangle$ is full. Indeed, since $\mathbf{K}$ is full, we have $\mathbf{K} = \mathsf{pgp}(\mathsf{precl}(\mathbf{K}))$ and hence $\langle \mathbf{K} \rangle = \langle \mathsf{pgp}(\mathsf{precl}(\mathbf{K})) \rangle \subseteq \langle \mathsf{pgp}(\mathbf{W}) \rangle$. The first part of the proof establishes that $\mathsf{pgp}(\mathbf{W})$ is a pseudovariety, so $\langle \mathbf{K} \rangle \subseteq \mathsf{pgp}(\mathbf{W})$. Moreover, the discussion in the previous paragraph, applied to $\mathbf{L} = \mathsf{precl}(\mathbf{K})$, shows that $\mathsf{pgp}(\mathbf{W}) \subseteq \langle \mathbf{K} \rangle$. The expected equality follows.

Applying this result to $\mathbf{K} = \mathbf{V}$, a full pseudovariety of *pg*-pairs, and to $\mathbf{W} = \langle \mathsf{precl}(\mathbf{V}) \rangle$, shows that $\mathbf{V} = \mathsf{pgp}(\mathbf{W})$, as announced. Finally, if $\mathbf{V} = \mathsf{pgp}(\mathbf{W}')$ for some other pseudovariety of preclones $\mathbf{W}'$, then $\mathbf{W}$ and $\mathbf{W}'$ have the same finitely generated elements, and hence must be equal by Proposition 1.8. This concludes the proof of the proposition. □

Before we discuss varieties of tree languages, let us define quotients of tree languages. Let $L \subseteq \Sigma M_k$, let k_1, k_2 be integers with $k_1 + k_2 \leq k$ and let $u \in \Sigma M_{k_1+1+k_2}$. The *left quotient* of L by (u, k_1, k_2) is the subset of $\Sigma M_{k-k_1-k_2}$

$$(u, k_1, k_2)^{-1} L = \{ f \in \Sigma M_{k-k_1-k_2} \mid u \cdot (\mathbf{k}_1 \oplus f \oplus \mathbf{k}_2) \in L \}.$$

If $v \in \Sigma M_{n,k}$, then the *right quotient* of L by v is

$$L v^{-1} = \{ f \in \Sigma M_n \mid f \cdot v \in L \}.$$

Remark 1.10 With the above notation, (u, k_1, v, k_2) is an L-context of an element f if and only if $f \in \big((u, k_1, k_2)^{-1} L\big) v^{-1} = (u, k_1, k_2)^{-1} \big(L(\mathbf{k}_1 \oplus v \oplus \mathbf{k}_2)^{-1}\big)$.

Moreover, if (u, k_1, v, k_2) and (u', k_1, v', k_2) are contexts such that $u \sim_L u'$ and $v \sim_L v'$, then $\big((u, k_1, k_2)^{-1} L\big) v^{-1} = \big((u', k_1, k_2)^{-1} L\big) v'^{-1}$. □

We say that a morphism $\varphi \colon \Sigma M \to \Sigma' M$ is a *literal morphism* if $\varphi(\Sigma) \subseteq \Sigma'$. A *variety of tree languages* (resp. a *literal variety of tree languages*) is a collection $\mathcal{V} = (\mathcal{V}_{\Sigma,k})_{\Sigma,k}$, where Σ runs over all ranked alphabets, k runs over non-negative integers, such that each $\mathcal{V}_{\Sigma,k}$ is a Boolean algebra of recognizable languages in ΣM_k, and $\mathcal{V}$ is closed under quotients and under inverse morphisms (resp. inverse literal morphisms) between free preclones. In particular, every variety of tree languages is a literal variety.

If $\mathbf{V}$ is a pseudovariety of preclones (resp. *pg*-pairs), we let $\mathsf{var}(\mathbf{V}) = (\mathcal{V}_{\Sigma,k})$ be such that $\mathcal{V}_{\Sigma,k}$ is the class of languages in ΣM_k with syntactic preclone (resp. *pg*-pair) in $\mathbf{V}$. If $\mathcal{V}$ is a variety (resp. literal variety) of tree languages, let $\mathsf{psv}(\mathcal{V})$ be the class of finitary preclones (resp. finitary *pg*-pairs) which only accept languages in $\mathcal{V}$. The following result was proved in [18].

Theorem 1.11 *The mappings* var *and* psv *are mutually inverse lattice isomorphisms between the lattice of pseudovarieties of preclones (resp. pg-pairs) and the lattice of varieties (resp. literal varieties) of tree languages.*

We note the following corollary of Theorem 1.11, which will be used in the sequel.

Corollary 1.12 *Let $\mathcal{V}$ be a literal variety and let* $\mathbf{V}$ *be the corresponding pseudovariety of pg-pairs. Then* $\mathbf{V}$ *is full if and only if* $\mathcal{V}$ *is a variety.*

Example 1.13 Let $\langle T_\exists \rangle$ be the pseudovariety of preclones generated by $T_\exists$, and let $\mathcal{V}$ be the corresponding tree language variety. Then a language $L \subseteq \Sigma M_k$ is in $\mathcal{V}_{\Sigma,k}$ if and only if L is a Boolean combination of languages of the form $\Sigma' M_k$, $\Sigma' \subseteq \Sigma$, see Example 1.4 and [18, Section 5.2.1]. □

More complex examples are discussed in [18, Section 5.2], and the main results of this article provide further examples.

2 Logically defined tree languages

Let Σ be a ranked alphabet. We will define tree languages by means of logical formulas. We consider the *atomic formulas* of the following form

$$P_\sigma(x),\ x < y,\ \mathsf{Succ}_i(x,y),\ \mathsf{root}(x),\ \mathsf{max}_{i,j}(x),\ \mathsf{left}_j(x) \text{ and } \mathsf{right}_j(x),$$

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, y are first-order variables. If $k \geq 0$, 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 in Section 2.1 below. As usual, each quantifier binds a first-order variable (within the scope of the quantifier), and variables that are not bound are called *free*. A formula without free variables is called a *sentence*. We denote by **Lind** the logic defined in this fashion.

When a **Lind**-formula is interpreted on a tree $t \in \Sigma M_k$, first-order variables are interpreted as nodes in $\mathsf{NV}(t)$ — and we assume $t \neq \mathbf{1}$, so that $\mathsf{NV}(t)$ is non-empty. Then $P_\sigma(x)$ holds if x is labeled σ ($\sigma \in \Sigma$), $x < y$ holds if y is a proper descendant of x, and $\mathsf{Succ}_i(x,y)$ holds if y is the i-th successor of x. Moreover, $\mathsf{root}(x)$ holds if x is the root of t and $\mathsf{max}_{i,j}(x)$ holds if the i-th successor of x is labeled by v_j, the j-th variable. Finally, $\mathsf{left}_j(x)$ (resp. $\mathsf{right}_j(x)$) holds if the index of the highest (resp. least) numbered variable labeling a leaf to the left (resp. right) of the frontier of the subtree rooted at x is j, see Figure 3. The interpretation of Lindström quantifiers is described in Section 2.1.

Recall that formally, an *interpretation* is a mapping λ from the set of free variables of a formula φ (or from a set containing the free variables of φ) to the set $\mathsf{NV}(t)$ of Σ-labeled nodes of a tree t. If t satisfies φ with this interpretation, we say that (t, λ) *satisfies* φ and we write $(t, \lambda) \models \varphi$. If φ is a sentence, we simply write $t \models \varphi$.

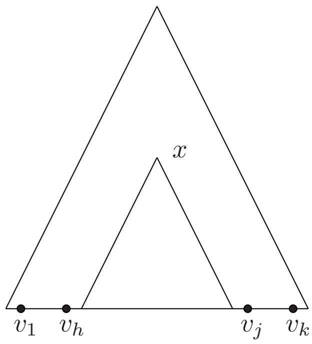

Figure 3: $\mathsf{left}_h(x) \wedge \mathsf{right}_j(x)$

Remark 2.1 In **Lind**-formulas, first-order variables are never interpreted as one of the v_i-labeled leaves. In fact, as far as logical constructs go, these particular leaves are not considered as proper nodes of the tree, but rather as place markers – which explains the fact that they are *labeled* by their position in the left-to-right order, and may be relabeled appropriately when trees are composed.

When we deal with traditional trees, that is, trees in ΣM_0, this peculiarity disappears, and we observe that in that case, our atomic formulas (P_σ, $<$, root and the Succ_i) are the atomic formulas of the usal logic on rooted ranked trees [37]. □

Example 2.2 Let A be an unranked alphabet, viewed as a ranked alphabet as in Example 1.1. Then AM_1 is equal to the set A^*v_1, and is isomorphic to the free monoid A^*. In this situation, the boundary of the trees in AM consist of a single node, labeled v_1, that is $\mathsf{left}_1(x)$ and $\mathsf{right}_1(x)$ always evaluate to false. Thus the relevant atomic formulas are $P_a(x)$ ($a \in A$), $x < y$, $\mathsf{Succ}_1(x, y)$, $\mathsf{root}(x)$ and $\mathsf{max}_{1,1}(x)$. Note that in this case, $\mathsf{root}(x)$ is the predicate usually denoted by $\mathsf{min}(x)$ (or $x = \mathsf{min}$) and $\mathsf{max}_{1,1}(x)$ is the predicate $\mathsf{max}(x)$ (or $x = \mathsf{max}$). That is, we have the same atomic formulas as in Büchi's classical sequential calculus [23, 37, 35]. The condition $t \neq \mathbf{1}$ imposed to interpret formulas, is equivalent to the fact that logical formulas are not interpreted on the empty word. □

Next to the atomic formulas defined above, we also use the following shorthand notation. Let $k > 0$ and let $\mathsf{left}_0(x)$ be the formula of rank k $\mathsf{left}_0(x) = \bigwedge_{j\in[k]} \neg\mathsf{left}_j(x)$. Then $\mathsf{left}_0(x)$ holds if no leaf situated to the left of the frontier of the subtree rooted at x, is labeled by a variable. We observe that for different values of k, we get different formulas $\mathsf{left}_0(x)$, and our notation assumes that k is clear from the context.

Similarly, if k is clear from the context, we define $\mathsf{right}_{k+1}(x)$ to be the formula of rank k $\mathsf{right}_{k+1}(x) = \bigwedge_{j\in[k]} \neg\mathsf{right}_j(x)$. Its meaning is that no leaf situated to the right of the frontier of the subtree rooted at x, is labeled by a variable.

2.1 Lindström quantifiers

Before we give formal definitions, we discuss an important example.

Example 2.3 Let us consider the first order formula $\exists x \cdot \varphi(x)$, where φ is a formula with free variables in a set $Y \cup \{x\}$ ($x \notin Y$). Let $\lambda: Y \to \mathsf{NV}(t)$. Recall that $(t, \lambda) \models \exists x \cdot \varphi$ if there exists a node v in $\mathsf{NV}(t)$ such that $(t, [\lambda; x \mapsto v]) \models \varphi$. For convenience, let λ_v denote the interpretation $[\lambda; x \mapsto v]$. We can express the satisfaction of $\exists x \cdot \varphi$ in the following, more generalizable fashion: we label each node $v \in \mathsf{NV}(t)$ with 1 if $(t, \lambda_v) \models \varphi$, with 0 otherwise (the variable labeled nodes are left unchanged). If $\bar{t}_\lambda$ denotes the resulting Boolean-labeled tree, then $(t, \lambda) \models \exists x \cdot \varphi$ if and only if $\bar{t}_\lambda$ belongs to the set of trees with at least one 1 label.

To be formally accurate, the nodes of $\bar{t}_\lambda$ must be labeled by a ranked alphabet, that is, we need to have, for each rank n, a letter 1_n and a letter 0_n. The definition of Lindström quantifiers below generalizes this example. □

Let Δ be a ranked alphabet containing letters of rank n for each n such that $\Sigma_n \neq \emptyset$ and let $\langle \varphi_\delta \rangle_{\delta \in \Delta}$ be a family of rank k formulas on Σ-trees. We say that this family is *deterministic with respect to* a first-order variable x if for each tree $t \in \Sigma M_k$, for each integer n, and for each interpretation λ of the free variables in the φ_δ mapping x to a rank n node of t, then (t, λ) satisfies exactly one of the φ_δ, $\delta \in \Delta_n$. Whenever needed, we will also assume that x is not bound in any of the φ_δ.

Example 2.4 If $\Delta = \Sigma$, a very simple example of such a family is given by letting $\varphi_\delta(x) = P_\delta(x)$ for each $\delta \in \Delta$. □

Example 2.5 Another natural example is given over a ranked Boolean alphabet Δ, that is, an alphabet such that whenever $\Delta_n \neq \emptyset$, then $\Delta_n = \{1_n, 0_n\}$. If for each such n, φ_{0_n} is logically equivalent to $\neg\varphi_{1_n}$, then $\langle \varphi_\delta \rangle_{\delta \in \Delta}$ is deterministic with respect to any first order variable x.

In later examples, when dealing with ranked Boolean alphabets, we will write φ_n instead of φ_{1_n} and we will assume that $\varphi_{0_n} = \neg\varphi_n$. Then a deterministic family will simply be written $\langle \varphi_n \rangle_n$. □

With this notion, we define (*simple*) *Lindström quantifiers*, a definition adapted from [20, 10] to the case of finite trees. Let $K \subseteq \Delta M_k$ be a language of rank k trees and let $\langle \varphi_\delta \rangle_{\delta \in \Delta}$ be a family of rank k formulas which is deterministic with respect to x. Then the quantified formula $Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$, where the quantifier Q_K binds the variable x, is interpreted in the following manner.

Given 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 $\bar{t}_\lambda \in \Delta M_k$ as follows: t and $\bar{t}_\lambda$ have the same underlying tree structure with the same variable-labeled nodes, that is, the same set of nodes and the same relations $<$, Succ_i, root, $\mathsf{max}_{i,j}$, left_j and right_j). Moreover, for each rank n node v of t (for some n), let λ_v be the interpretation

$[\lambda, x \mapsto v]$: then the node v in $\bar{t}_\lambda$ is labeled by the unique element $\delta \in \Delta_n$ such that (t, λ_v) satisfies φ_δ. The tree $\bar{t}_\lambda$ is called the *characteristic tree* determined by t, λ and the formulas φ_δ. If the φ_δ have no free variable other than x, we write $\bar{t}$ for $\bar{t}_\lambda$. Finally, we say that (t, λ) *satisfies* $Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$ if $\bar{t}_\lambda \in K$.

Remark 2.6 With the above notation, we note that $(t, \lambda_v) \models \varphi_\delta(x)$ if and only if $(\bar{t}_\lambda, [x \mapsto v]) \models P_\delta(x)$. Since x is the only free variable in $P_\delta(x)$, this is also equivalent to $(\bar{t}_\lambda, \lambda_v) \models P_\delta(x)$. □

Example 2.7 Suppose that $\Delta = \Sigma$ and $\varphi_\delta = P_\delta(x)$ as in Example 2.4. If $t \in \Sigma M_k$, it is easily verified that the trees $\bar{t}$ and t are equal. In particular, if $K \subseteq \Delta M_k$ is a language of k-ary trees, then t satisfies $Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$ if and only if $t \in K$. □

Example 2.8 Let A be an unranked alphabet, seen as a ranked alphabet as usual, suppose that $\Delta_k = \emptyset$ for all $k \neq 1$ and let $K \subseteq \Delta M_1$. Then K can also be seen as a word language since ΔM_1 is isomorphic to the free monoid Δ_1^*, and the logic **Lind** is analogous to the logic for word languages studied by Ésik and Larsen in [16] (the latter does not include min and max among its atomic formulas). □

In the next examples, Δ is a ranked *Boolean* alphabet such that Δ_n is non-empty whenever Σ_n is, and $\langle \varphi_n \rangle_n$ is a family of formulas which is deterministic with respect to a first order variable x, see Example 2.5.

Example 2.9 Let $K = K_k(\exists)$ denote the set of all trees in ΔM_k containing at least one node labeled 1_n (for some n), see Example 1.4. Then the Lindström quantifier Q_K is a generalization of the existential quantifier, as indicated in Example 2.3.

More precisely, (t, λ) satisfies $Q_K x \cdot \langle \varphi_n \rangle_n$ if and only if there exists a node $v \in \mathsf{NV}(t)$ such that (t, λ_v) satisfies φ_n, where n is the rank of v and λ_v is the interpretation $[\lambda, x \mapsto v]$.

Let finally A be an unranked alphabet, viewed as a ranked alphabet as in Example 1.1, and suppose that $k = 1$. Then $(t, \lambda) \models Q_K x \cdot \langle \varphi_n \rangle_n$ (where t is viewed as a tree in AM_1) if and only if $(t, \lambda) \models \exists x \cdot \varphi_1(x)$ (where t is viewed as a non-empty word in A^*). □

Example 2.10 In the same manner as in Example 2.9, if $p \geq 1$, $r < p$ and $K = K(\exists_p^r)$ denotes the set of those trees in ΔM_k such that the number of nodes labeled 1_n (for some n) is congruent to r modulo p (see Example 1.4), then the Lindström quantifier Q_K is a generalization of a modular quantifier.

More precisely, (t, λ) satisfies $Q_K x \cdot \langle \varphi_n \rangle_n$ if and only if, for some n, the number of nodes $v \in \mathsf{NV}(t)$ such that (t, λ_v) satisfies $\varphi_n(x)$ (where n is the rank of v) is congruent to r mod p.

If A is an unranked alphabet, then $(t, \lambda) \models Q_K x \cdot \langle \varphi_n \rangle_n$ (where t is viewed as a tree in AM_1) if and only if $(t, \lambda) \models \exists_p^r x \cdot \varphi_1(x)$ (where t is viewed as a non-empty word in A^*). □

Example 2.11 Let $K = K_k(\exists_{\mathsf{path}})$ be the set of all trees in ΔM_k such that all the nodes along at least one path from the root to a leaf are labeled 1_n (for appropriate values of n).

Then (t, λ) satisfies $Q_K x \cdot \langle \varphi_n \rangle_n$ if and only if there exists a root-to-leaf path such that, for every node $v \in \mathsf{NV}(t)$ along this path, $(t, \lambda_v) \models \varphi_n(x)$ (where n is the rank of v).

If A is an unranked alphabet, then $(t, \lambda) \models Q_K x \cdot \langle \varphi_n \rangle_n$ (where t is viewed as a tree in AM_1) if and only if $(t, \lambda) \models \forall x \cdot \varphi_1(x)$ (where t is viewed as a non-empty word in A^*). □

Example 2.12 Let $K_k(\forall_{\mathsf{next}})$ be the set of all trees in ΔM_k such that the children of the root are labeled 1_n (for the appropriate n).

Then (t, λ) satisfies $Q_K x \cdot \langle \varphi_n \rangle_n$ if and only if, for every child v of the root, $(t, \lambda_v) \models \varphi_n(x)$ (where n is the rank of v).

If A is an unranked alphabet, then $(t, \lambda) \models Q_K x \cdot \langle \varphi_n \rangle_n$ (where t is viewed as a tree in AM_1) if and only if $(t, \lambda) \models \exists x \cdot (\mathsf{Succ}(1, x) \wedge \varphi_1(x))$, — more formally, $(t, \lambda) \models \exists x \cdot \big((\forall y \cdot (\min(y) \rightarrow \mathsf{Succ}(y, x))) \wedge \varphi_1(x)\big)$ — (where t is viewed as a non-empty word in A^*).

Other next modalities can be expressed likewise, *e.g.*, requesting that at least one (resp. an even number, etc.) of the children of the root satisfies the appropriate φ_n. □

2.2 The language associated with a Lind-formula

Let φ be a **Lind**-sentence of rank k over Σ. We denote by L_φ the set of trees in ΣM_k that satisfy φ, and we say that L_φ is *defined* by the formula φ.

For a class $\mathcal{K}$ of tree languages, we let **Lind**$(\mathcal{K})$ denote the fragment of **Lind** consisting of the formulas in which all Lindström quantifiers are of the form Q_K with $K \in \mathcal{K}$. If φ is a **Lind**$(\mathcal{K})$ sentence, we say that L_φ is **Lind**$(\mathcal{K})$*-definable*, and we let $\mathcal{L}ind(\mathcal{K})$ denote the class of **Lind**$(\mathcal{K})$-definable tree languages.

Example 2.13 Let $\mathcal{K}_\exists$ be the class of all the languages of the form $K_k(\exists)$ on a Boolean ranked alphabet. In view of the discussion in Example 2.9, it is reasonable to say that $\mathcal{L}ind(\mathcal{K}_\exists)$ is exactly the class of FO-definable tree languages. Examples 2.9 and 2.10 show that if $\mathcal{K}_{\exists,\mathrm{mod}}$ is the class of all languages of the form $K_k(\exists)$ or $K_k(\exists_p^r)$, then $\mathcal{L}ind(\mathcal{K}_{\exists,\mathrm{mod}})$ is the class of $(\mathsf{FO} + \mathsf{MOD})$-definable tree languages. □

It will be useful to associate a tree language also with the **Lind**-formulas that contain free variables (as is done in [35, Section II-2] for word languages). Let Z be a finite set. We extend Σ to the ranked alphabet Σ_Z, whose set of letters of rank m ($m \geq 0$) is $\Sigma_m \times \mathcal{P}(Z)$. We identify each $\sigma \in \Sigma$ with the pair $(\sigma, \emptyset) \in \Sigma_Z$. An element $z \in Z$ is said to *occur* in $t \in \Sigma_Z M$ at node v if the label of v is of the form (σ, Z') and $z \in Z'$. If each $z \in Z$ occurs exactly once in $t \in \Sigma_Z M_k$, then t is called a *Z-structure of rank k* over Σ. We note that a Z-structure uniquely determines a tree $t \in \Sigma M$ and a mapping $\lambda : Z \to \mathsf{NV}(t)$. Conversely, any such pair (t, λ) determines a unique Z-structure, written $\mathsf{str}(t, \lambda)$. Now let φ be a rank k **Lind**-formula with free variables in a set Y. Let $\mathsf{str}(t, \lambda)$ be a Z-structure with $Z \subseteq Y$. If $\mu : Y \setminus Z \longrightarrow NV(t)$, we write $(\mathsf{str}(t, \lambda), \mu) \models \varphi$ if $(t, [\lambda; \mu]) \models \varphi$, where $[\lambda; \mu]$ is the map from Y to $\mathsf{NV}(t)$ determined by λ and μ. If $Z = Y$, we write simply $\mathsf{str}(t, \lambda) \models \varphi$ and we say that $\mathsf{str}(t, \lambda)$ *satisfies* φ. We let L_φ be the set of Y-structures satisfying φ.

Example 2.14 Let $\sigma \in \Sigma_m$, and let φ be the rank k formula $\varphi = P_\sigma(x)$. Let Y be a set containing x. Then L_φ is the collection of all Y-structures of rank k over Σ such that some (necessarily unique) node has a label of the form (σ, Y') with $x \in Y'$. It is immediate to observe that any two trees of equal rank in ΣM have the same contexts in L_φ, that is, the restriction of the syntactic congruence of L_φ to ΣM is the universal relation. The same holds for any atomic formula φ. □

2.3 Properties of the operator $\mathcal{L}ind$

We now explore the properties of the operator $\mathcal{L}ind$ on families of languages.

2.3.1 $\mathcal{L}ind$ is a closure operator

Theorem 2.15 *$\mathcal{L}ind$ is a closure operator on classes of languages. That is, for all language classes $\mathcal{K}$ and $\mathcal{K}'$, the following holds.*

(1) $\mathcal{K} \subseteq \mathcal{L}ind(\mathcal{K})$;

(2) if $\mathcal{K} \subseteq \mathcal{K}'$ *then* $\mathcal{L}ind(\mathcal{K}) \subseteq \mathcal{L}ind(\mathcal{K}')$;

(3) $\mathcal{L}ind(\mathcal{L}ind(\mathcal{K})) \subseteq \mathcal{L}ind(\mathcal{K})$.

Item (1) follows immediately from Example 2.7, and Item (2) is immediate from the definition. The rest of Section 2.3.1 is devoted to the proof of Item (3).

Let φ be a **Lind**$(\mathcal{L}ind(\mathcal{K}))$-formula of rank k over Σ. We argue by induction on the structure of φ to show that there is an equivalent formula $\hat{\varphi}$ of **Lind**$(\mathcal{K})$, that is, a formula with the same free variables as φ and such that $L_\varphi = L_{\hat{\varphi}}$ in $\Sigma_Y M_k$ for any finite set Y containing the free variables of φ. This will be sufficient to prove Theorem 2.15.

If φ is an atomic formula, we let $\hat{\varphi} = \varphi$, since φ is also a **Lind**$(\mathcal{K})$-formula. If $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), we let $\hat{\varphi} = \hat{\varphi}_1 \vee \hat{\varphi}_2$ (resp. $\hat{\varphi} = \neg\hat{\varphi}_1$). The equivalence of φ and $\hat{\varphi}$ is easily verified.

The last case occurs when φ is of the form $\varphi = Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$, where $K \in \mathcal{L}ind(\mathcal{K})$ and the φ_δ form a family of rank k formulas of **Lind**$(\mathcal{L}ind(\mathcal{K}))$ over Σ that is deterministic with respect to x. In particular, $K = L_\psi$ where ψ is a rank k **Lind**$(\mathcal{K})$-sentence over Δ.

By induction, for each $\delta \in \Delta$, there exists a $\mathbf{Lind}(\mathcal{K})$-formula $\hat{\varphi}_\delta$ equivalent to φ_δ, so that φ is equivalent to $Q_K x \cdot \langle \hat{\varphi}_\delta \rangle_{\delta\in\Delta}$. Thus, we may assume that the φ_δ are $\mathbf{Lind}(\mathcal{K})$-formulas.

Before we proceed with the end of the proof, we establish a technical fact. If χ is a formula and p, q are variables, we denote by $\chi[q/p]$ the formula obtained from χ by substituting the variable q for all free occurrences of p. (Bound occurrences of q in χ are renamed as usual.)

Let χ be a rank k formula over Δ. We then define $\tilde{\chi}$ to be the rank k formula over Σ obtained from χ by replacing each subformula of the form $P_\delta(z)$, where z is any first-order variable, by the formula $\varphi_\delta[z/x]$. Since the quantifiers in χ also occur in $\tilde{\chi}$, and the quantifiers in $\tilde{\chi}$ occur either in χ or in the φ_δ, it is clear that χ is a $\mathbf{Lind}(\mathcal{K})$-formula if and only if $\tilde{\chi}$ is one. In the sequel, we assume that neither x nor any free variable of one of the φ_δ, is free in χ, and that no free variable has bound occurrences in the formulas under consideration. Let us then assume that Y (the finite set containing the free variables of φ and not containing x) also contains the free variables of χ.

Fact 2.16 *With the notation above, let $t \in \Sigma M_k$, let $\lambda: Y \to \mathsf{NV}(t)$ be a function, and let $\bar{t}_\lambda \in \Delta M_k$ be the characteristic tree determined by t, λ and the formulas φ_δ. Then we have*

$$(t, \lambda) \models \tilde{\chi} \iff (\bar{t}_\lambda, \lambda) \models \chi.$$

Proof. We argue by induction on the structure of χ. Suppose first that $\chi = P_\delta(z)$. Then $\tilde{\chi} = \varphi_\delta[z/x]$. Let μ be the restriction of λ to $Y \setminus \{z\}$ and let $\bar{t}_\mu$ be the characteristic tree determined by t, μ and the $\varphi_\delta[z/x]$. A node v is labeled δ in $\bar{t}_\lambda$ if and only if $(t, \lambda_v) = (t, [\lambda; x \mapsto v]) \models \varphi_\delta$. Since z does not occur in φ_δ, this is equivalent to $(t, [\mu; z \mapsto v]) \models \varphi_\delta[z/x]$, and hence to the labeling of v by δ in $\bar{t}_\mu$. Thus $\bar{t}_\lambda = \bar{t}_\mu$. Then we have:

$$\begin{aligned}
(t, \lambda) \models \tilde{\chi} = \varphi_\delta[z/x] &\iff (t, [\mu; z \mapsto \lambda(z)]) \models \varphi_\delta[z/x] \text{ by definition of } \mu \\
&\iff (\bar{t}_\mu, [z \mapsto \lambda(z)]) = (\bar{t}_\lambda, [z \mapsto \lambda(z)]) \models P_\delta(z) \\
&\iff (\bar{t}_\lambda, \lambda) \models P_\delta(z) = \chi.
\end{aligned}$$

If χ is another atomic formula (namely, $z_1 < z_2$, $\mathsf{Succ}_i(z_1, z_2)$, $\mathsf{root}(x)$, $\mathsf{max}_{i,j}(z)$, $\mathsf{left}_j(z)$ or $\mathsf{right}_j(z)$ with $z, z_1, z_2 \in Y \cup \{x\}$), then $\tilde{\chi} = \chi$. Since t and $\bar{t}_\lambda$ have the same variable-labeled nodes and they differ only in the labeling of their nodes, and since χ does not depend on that labeling, we have in each case

$$(t, \lambda) \models \tilde{\chi} \iff (\bar{t}_\lambda, \lambda) \models \chi.$$

We have now established our claim for atomic formulas.

The induction step is immediate if χ is of the form $\chi = \chi_1 \vee \chi_2$ or $\chi = \neg\chi_1$. We now assume that $\chi = Q_L z \cdot \langle \chi_\omega \rangle_{\omega\in\Omega}$ where $L \subseteq \Omega M_k$, $z \notin Y \cup \{x\}$, and $\langle \chi_\omega \rangle_{\omega\in\Omega}$ is a family of rank k formulas over Δ with free variables in $Y \cup \{z\}$, which is deterministic with respect to z.

By construction $\tilde{\chi} = Q_L z \cdot \langle \tilde{\chi}_\omega \rangle_{\omega\in\Omega}$, and by induction hypothesis, for each node $w \in \mathsf{NV}(t)$ and for each $\omega \in \Omega$, we have

$$(t, [\lambda; z \mapsto w]) \models \tilde{\chi}_\omega \iff (\bar{t}_\lambda, [\lambda; z \mapsto w]) \models \chi_\omega.$$

It follows in particular that $\langle \tilde{\chi}_\omega \rangle_{\omega\in\Omega}$ is deterministic with respect to z. Moreover, the characteristic tree determined by t, λ and $\langle \tilde{\chi}_\omega \rangle_{\omega\in\Omega}$ is the same as that determined by $\bar{t}_\lambda$, λ and $\langle \chi_\omega \rangle_{\omega\in\Omega}$. Thus we have

$$(t, \lambda) \models \tilde{\chi} \iff (\bar{t}_\lambda, \lambda) \models \chi,$$

which concludes the induction and the proof. □

We now return to the proof of Theorem 2.15. Recall that $\varphi = Q_K x \cdot \langle \varphi_\delta \rangle_{\delta\in\Delta}$ and $K = L_\chi \subseteq \Delta M_k$ for some rank k $\mathbf{Lind}(\mathcal{K})$-sentence χ over Δ (without free variables). We want to construct a formula in $\mathbf{Lind}(\mathcal{K})$ that is equivalent to φ and we claim that $\tilde{\chi}$ is such a formula.

Indeed, let $t \in \Sigma M_k$, let λ be a mapping $\lambda\colon Y \mapsto \mathsf{NV}(t)$, and let $\bar{t}_\lambda \in \Delta M_k$ be the characteristic tree determined by t, λ and the φ_δ. By definition, $(t, \lambda) \models \varphi$ if and only if $\bar{t}_\lambda \in K$, that is, $\bar{t}_\lambda \models \psi$, or equivalently, $(\bar{t}_\lambda, \lambda) \models \psi$. It was established in Fact 2.16 that this is equivalent to $(t, \lambda) \models \tilde{\psi}$, which concludes the proof. □

2.3.2 Closure properties of $\mathcal{L}ind(\mathcal{K})$

The objective of this section is to prove the closure properties summarized in Theorem 2.17 below.

Theorem 2.17 *$\mathcal{L}ind(\mathcal{K})$ is closed under Boolean operations and inverse literal morphisms. Moreover, $\mathcal{L}ind(\mathcal{K})$ is closed under left (resp. right) quotients if and only if any left (resp. right) quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$.*

We now prove Theorem 2.17, by considering separately each closure property.

Boolean operations The fact that $\mathcal{L}ind(\mathcal{K})$ is closed under the Boolean operations follows directly from the fact that $\mathbf{Lind}(\mathcal{K})$-formulas are closed under disjunction and negation. □

Inverse literal morphisms Let $h\colon \Sigma' \to \Sigma$ be a rank-preserving mapping, and let us also denote by h the induced morphism $h\colon (\Sigma' M, \Sigma') \to (\Sigma M, \Sigma)$. Note that if t is a tree, then $h(t)$ differs from t only in the labeling of the nodes in $\mathsf{NV}(t)$. Let φ be a rank k $\mathbf{Lind}(\mathcal{K})$-formula over Σ with free variables in a finite set Y. We show by structural induction on φ that there exists a rank k $\mathbf{Lind}(\mathcal{K})$-formula φ' over Σ', with the same free variables as φ, and such that $(t, \lambda) \models \varphi'$ if and only if $(h(t), \lambda) \models \varphi$ for any tree $t \in \Sigma' M_k$ and any interpretation $\lambda\colon Y \to \mathsf{NV}(t)$.

If $\varphi = P_\sigma(x)$ for some $\sigma \in \Sigma$, we let $\varphi' = \bigvee P_{\sigma'}(x)$, where the disjunction runs over the letters $\sigma' \in \Sigma'$ such that $h(\sigma') = \sigma$. If φ is another type of atomic formula, then φ does not depend on the labeling of the tree, and it suffices to choose $\varphi' = \varphi$.

The inductive step for the Boolean connectives is equally natural: if $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), then we let $\varphi' = \varphi'_1 \vee \varphi'_2$ (resp. $\varphi' = \neg\varphi'_1$).

Suppose finally that φ is of the form $Q_K x \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$. By induction, there exist formulas φ'_δ over Σ' such that, for each δ, $(t, \lambda_v) \models \varphi'_\delta$ if and only if $(h(t), \lambda_v) \models \varphi_\delta$ for any tree $t \in \Sigma' M_k$, node v in t and mapping $\lambda\colon Y \to \mathsf{NV}(t)$. It follows that the characteristic tree determined by t, λ and $\langle\varphi'_\delta\rangle_{\delta\in\Delta}$, and the characteristic tree determined by $h(t)$, λ and $\langle\varphi_\delta\rangle_{\delta\in\Delta}$ coincide. As a result, we have $(t, \lambda) \models \varphi'$ if and only if $(h(t), \lambda) \models \varphi$. □

Left quotients We now assume that any left quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$. Let k_1, k_2, ℓ be non-negative integers and let $k = k_1+\ell+k_2$. Let also φ be a rank k $\mathbf{Lind}(\mathcal{K})$-formula over Σ with free variables in a finite set Y, and let $U = \mathsf{str}(u, \mu)$ be a Z-structure of rank $k_1 + 1 + k_2$ for some $Z \subseteq Y$. (Without loss of generality, we may assume that $u \neq \mathbf{1}$.) Let $X = Y \setminus Z$. We prove by structural induction on φ that there exists a rank ℓ $\mathbf{Lind}(\mathcal{K})$-formula φ' over Σ, with free variables in X and such that, for every tree $t \in \Sigma M_\ell$ and every mapping $\lambda\colon X \to \mathsf{NV}(t)$ (see Figure 4), we have

$$(t, \lambda) \models \varphi' \iff (U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2), \lambda) \models \varphi.$$

If φ is a formula without free variables ($X = Y = Z = \emptyset$, $U = u \in \Sigma M_{k_1+1+k_2}$), this shows that $L_{\varphi'} = (u, k_1, k_2)^{-1} L_\varphi$, and hence that $\mathcal{L}ind(\mathcal{K})$ is closed under left quotients.

We now proceed with the proof. We first observe that $\mathsf{NV}(t)$ may be viewed as a subset of $\mathsf{NV}(U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2))$: more precisely, the latter set is equal to the disjoint union of $\mathsf{NV}(t)$ and $\mathsf{NV}(U) = \mathsf{NV}(u)$.

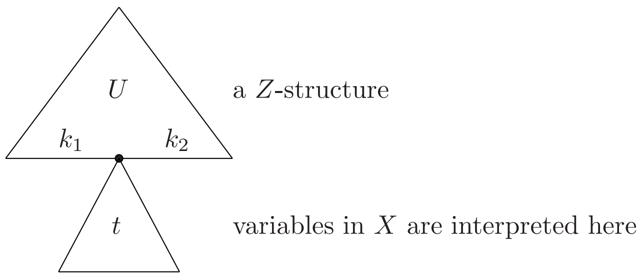

Figure 4: $S = U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2)$

If φ is equal to $P_\sigma(x)$, we let $\varphi' = \varphi$ if $x \notin Z$, and $\varphi' = \mathsf{true}$ (resp. false) if $x \in Z$ and x occurs at a node of U for which the first component of the label is (resp. is not) σ. That is, if $x \in Z$ and U satisfies (resp. does not satisfy) φ.

Let now $\varphi = \mathsf{left}_j(x)$ (resp. $\mathsf{right}_j(x)$, $\mathsf{max}_{i,j}(x)$) with $1 \leq j \leq k$. If $x \notin Z$, we let $\varphi' = \mathsf{left}_{j-k_1}(x)$ if $k_1 \leq j \leq k_1+\ell$ (resp. $\mathsf{right}_{j-k_1}(x)$ if $k_1 < j \leq k_1+\ell+1$, $\mathsf{max}_{i,j-k_1}(x)$ if $k_1 < j \leq k_1 + \ell$), and $\varphi' = \mathsf{false}$ otherwise. If $x \in Z$, then it does not depend on t and λ whether $(U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2), \lambda) \models \varphi$ or not, and we let $\varphi' = \mathsf{true}$ or false accordingly.

If $\varphi = \mathsf{root}(x)$ and $x \notin Z$, we let $\varphi' = \mathsf{false}$. If $x \in Z$, we let $\varphi' = \mathsf{true}$ or false depending on whether $U \models \varphi$.

Now consider the case where $\varphi = (x < y)$. If $x, y \notin Z$, we let $\varphi' = \varphi$. If $x \notin Z$ and $y \in Z$, we let $\varphi' = \mathsf{false}$. If $x \in Z$ but $y \notin Z$, we let $\varphi' = \mathsf{true}$ or false depending whether the node of U where x occurs in an ancestor of the $(k_1 + 1)$-st variable leaf. Finally, if $x, y \in Z$, we let $\varphi' = \mathsf{true}$ or false depending whether $U \models \varphi$.

The last case of an atomic formula occurs if φ is of the form $\mathsf{Succ}_i(x, y)$. If $x, y \notin Z$, we let $\varphi' = \varphi$. If $x, y \in Z$, we let $\varphi' = \mathsf{true}$ or false, depending on whether $U \models \varphi$. If $x \notin Z$ and $y \in Z$, we let $\varphi' = \mathsf{false}$. Finally, if $x \in Z$ and $y \notin Z$, let w be the node of U where x occurs. If the i-th successor of w is the $(k_1 + 1)$-st variable leaf, we let $\varphi' = \mathsf{root}(y)$; otherwise we let $\varphi' = \mathsf{false}$.

As usual, if $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), then we let $\varphi' = \varphi_1' \vee \varphi_2'$ (resp. $\varphi' = \neg\varphi_1'$).

We now consider the case where $\varphi = Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$. We may assume that $x \notin Y$. Let $S = U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2)$, let $s = u \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2)$ and let $\lambda\colon X \to \mathsf{NV}(t)$. Let $\bar{s}_\lambda$ be the characteristic tree determined by s, $[\lambda; \mu]$ and $\langle \varphi_\delta \rangle_{\delta \in \Delta}$. Then, for any $v \in \mathsf{NV}(s)$ and for any δ, we have

$$\begin{aligned}(\bar{s}_\lambda, [x \mapsto v]) \models P_\delta(x) &\iff (s, [\lambda; \mu; x \mapsto v]) \models \varphi_\delta \\ &\iff (S, [\lambda; x \mapsto v]) \models \varphi_\delta.\end{aligned}$$

Moreover,

$$(s, [\lambda; \mu]) \models \varphi \iff (S, \lambda) \models \varphi \iff \bar{s}_\lambda \in K.$$

For each $\delta \in \Delta$, let φ_δ' be the formula associated with φ_δ and U by the induction hypothesis. Let $\bar{t}_\lambda$ be the characteristic tree determined by t, λ and $\langle \varphi_\delta' \rangle_{\delta \in \Delta}$. Then, for any node $v \in \mathsf{NV}(t)$, we have

$$\begin{aligned}(S, [\lambda; x \mapsto v]) \models \varphi_\delta &\iff (t, [\lambda; x \mapsto v]) \models \varphi_\delta' \\ &\iff (\bar{t}_\lambda, [x \mapsto v]) \models P_\delta(x),\end{aligned}$$

and hence $\bar{s}_\lambda$ is of the form $\bar{s}_\lambda = \hat{u} \cdot (\mathbf{k}_1 \oplus \bar{t}_\lambda \oplus \mathbf{k}_2)$ for some tree $\hat{u}$ which differs from u only in the labeling of the nodes in $\mathsf{NV}(u)$.

For each $v \in \mathsf{NV}(u)$, we let $U^{(v)}$ be the $(Z \cup \{x\})$-structure obtained from U by adding x to the second component of the label of v. Then, for each $\delta \in \Delta$, we let $\psi_{\delta,v}$ be the formula associated with φ_δ and $U^{(v)}$ by the induction hypothesis. Then we have

$$\begin{aligned}(t, \lambda) \models \psi_{\delta,v} &\iff (U^{(v)} \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2), \lambda) \models \varphi_\delta \\ &\iff (S, [\lambda; x \mapsto v]) \models \varphi_\delta \\ &\iff (\bar{s}_\lambda, [x \mapsto v]) \models P_\delta(x).\end{aligned}$$

Now, for each mapping $\alpha\colon \mathsf{NV}(u) \to \Delta$, let $\hat{u}_\alpha$ be the tree obtained from u by relabeling each node $v \in \mathsf{NV}(u)$ with $\alpha(v)$. Let also ψ_α be the conjunction of the $\psi_{\alpha(v),v}$ when v runs over $\mathsf{NV}(u)$. Then

$$(t, \lambda) \models \psi_\alpha \iff \bar{s}_\lambda = \hat{u}_\alpha \cdot (\mathbf{k}_1 \oplus \bar{t}_\lambda \oplus \mathbf{k}_2).$$

Finally, let

$$\varphi'' = \bigvee_\alpha \Big(\psi_\alpha \wedge Q_{(\hat{u}_\alpha,k_1,k_2)^{-1}K}\langle \varphi'_\delta \rangle_\delta\Big),$$

where the disjunction runs over all mappings $\alpha\colon \mathsf{NV}(u) \to \Delta$. Then the above discussion establishes that (t, λ) satisfies φ'' if and only if $(U \cdot (\mathbf{k}_1 \oplus t \oplus \mathbf{k}_2), \lambda)$ satisfies φ. Moreover, since each $(\hat{u}_\alpha, k_1, k_2)^{-1}K$ is in $\mathcal{L}ind(\mathcal{K})$, the formula φ'' is a $\mathbf{Lind}(\mathcal{L}ind(\mathcal{K}))$-formula, and by Theorem 2.15, φ'' is equivalent to some $\mathbf{Lind}(\mathcal{K})$-formula φ', which concludes this proof. □

Right quotients The proof concerning the closure under right quotients is similar. We assume that every right quotient of a language in $\mathcal{K}$ belongs to $\mathcal{L}ind(\mathcal{K})$. Let $k \geq 0$ and let φ be a rank k $\mathbf{Lind}(\mathcal{K})$-formula over Σ with free variables in a finite set Y. Let $n \geq 1$ and $Z \subseteq Y$, and let $U = U_1 \oplus \cdots \oplus U_n \in \Sigma_Z M_{n,k}$ where each U_i is a Z_i-structure of rank k_i, $U_i = \mathsf{str}(u_i, \mu_i)$, $k = \sum_i k_i$ and the Z_i form a partition of Z. Let $u = \oplus_i u_i$, $\mu = [\mu_1, \ldots, \mu_n]$ and $X = Y \setminus Z$.

We show by structural induction on φ that there exists a rank n $\mathbf{Lind}(\mathcal{K})$-formula φ' with free variables in X such that, for every tree $t \in \Sigma M_n$ and every mapping $\lambda\colon X \to \mathsf{NV}(t)$ (see Figure 5), we have

$$(t, \lambda) \models \varphi' \iff (t \cdot U, \lambda) \models \varphi.$$

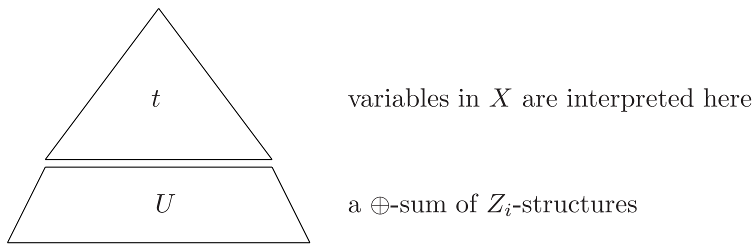

Figure 5: $S = t \cdot U$

We first consider the case of atomic formulas. If $\varphi = P_\sigma(x)$, we let $\varphi' = \varphi$ if $x \notin Z$ and $\varphi' = \mathsf{true}$ or false if $x \in Z$, depending on whether U satisfies φ.

If $\varphi = \mathsf{root}(x)$, we let $\varphi' = \varphi$ if $x \notin Z$, and false if $x \in Z$.

If $\varphi = \mathsf{max}_{i,j}(x)$ and $x \notin Z$, we let $\varphi' = \varphi$ if $j = k_1 + \cdots + k_{h-1} + 1$ for some h such that $U_h = \mathbf{1}$, and $\varphi' = \mathsf{false}$ otherwise. If $x \in Z$ and $k_1 + \cdots + k_{h-1} \leq j \leq k_1 + \cdots + k_h$, we let $\varphi' = \mathsf{true}$ or false depending on whether $U_h \models \mathsf{max}_{i,j-(k_1+\cdots+k_{h-1})}(x)$.

Suppose now that $\varphi = (x < y)$. If $x, y \notin Z$, we let $\varphi' = \varphi$. If $x, y \in Z$, we let $\varphi' = \mathsf{true}$ or false depending on whether one of the U_j satisfies φ. If $x \in Z$ and $y \notin Z$, we let $\varphi' = \mathsf{false}$. Finally, if $x \notin Z$ and $y \in Z$, let $1 \leq j \leq n$ be such that $y \in Z_j$ (i.e. y occurs in U_j). Then we let $\varphi' = \bigvee_{i<j} \mathsf{left}_i(x) \wedge \bigvee_{j<h} \mathsf{right}_h(x)$.

The situation is similar if $\varphi = \mathsf{Succ}_i(x, y)$. If $x, y \notin Z$, we let $\varphi' = \varphi$. If $x, y \in Z$, we let $\varphi' = \mathsf{true}$ or false depending on whether one of the U_j satisfies φ. If $x \in Z$ and $y \notin Z$, we let $\varphi' = \mathsf{false}$. Finally, if $x \notin Z$ and $y \in Z$, let j be such that $y \in Z_j$. If y does not occur at the root of U_j, we let $\varphi' = \mathsf{false}$. If y does occur at the root of U_j, we let $\varphi' = \mathsf{max}_{i,j}(x)$.

Finally, suppose that $\varphi = \mathsf{left}_j(x)$ (resp. $\mathsf{right}_j(x)$). If $x \in Z$, let i be such that $x \in Z_i$. Then we let $\varphi' = \mathsf{true}$ or false according to whether $k_1 + \cdots + k_{i-1} \leq j \leq k_1 + \cdots + k_i$ and U_i satisfies $\mathsf{left}_{j-(k_1+\cdots+k_{i-1})}(x)$ (resp. $\mathsf{right}_{j-(k_1+\cdots+k_{i-1})}(x)$). If $x \notin Z$, we let $\varphi' = \mathsf{left}_h(x)$ if $j = \sum_{i \leq h} k_i$ (resp. $j = 1 + \sum_{i \leq h} k_i$) for some h, and $\varphi' = \mathsf{false}$ if j is not of that form.

If $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), then we let $\varphi' = \varphi'_1 \vee \varphi'_2$ (resp. $\varphi' = \neg\varphi'_1$), and we now assume that $\varphi = Q_K x \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$, with $x \notin Y$. Let $S = t \cdot U$, let $s = t \cdot u$ and let $\lambda \colon X \to \mathsf{NV}(t)$. Let $\bar{s}_\lambda$ be the characteristic tree determined by s, $[\lambda; \mu]$ and $\langle \varphi_\delta \rangle_{\delta \in \Delta}$. For each $\delta \in \Delta$, let φ'_δ be the formula associated with φ_δ and U by the induction hypothesis, and let $\bar{t}_\lambda$ be the characteristic tree determined by t, λ and $\langle \varphi'_\delta \rangle_{\delta \in \Delta}$. Then the tree $\bar{s}_\lambda$ is of the form $\bar{s}_\lambda = \bar{t}_\lambda \cdot \hat{u}$ for some tree $\hat{u}$ which differs from u only in the labeling of the nodes in $\mathsf{NV}(u)$.

We continue as in the left quotient case. For each $v \in \mathsf{NV}(u)$, we let $U^{(v)}$ be the structure obtained from U by adding x to the second component of the label of v and for each $\delta \in \Delta$, we let $\psi_{\delta,v}$ be the formula associated with φ_δ and $U^{(v)}$ by the induction hypothesis. As above, we verify that if $\hat{u}_\alpha$ is the relabeling of u determined by the mapping $\alpha \colon \mathsf{NV}(u) \longrightarrow \Delta$, and if ψ_α is the conjunction of the $\psi_{\alpha(v),v}$ (over the nodes $v \in NV(u)$), then

$$(t, \lambda) \models \psi_\alpha \iff \bar{s}_\lambda = \bar{t}_\lambda \cdot \hat{u}_\alpha.$$

We then let

$$\varphi'' = \bigvee_\alpha \Big(\psi_\alpha \wedge Q_{K\hat{u}_\alpha^{-1}} \langle \varphi'_\delta \rangle_\delta \Big),$$

where the disjunction runs over all mappings $\alpha \colon \mathsf{NV}(u) \to \Delta$, and we note that (t, λ) satisfies φ'' if and only if $(t \cdot U, \lambda)$ satisfies φ. Since each $K\hat{u}_\alpha^{-1}$ is in $\mathcal{L}ind(\mathcal{K})$, the formula φ'' is a $\mathbf{Lind}(\mathcal{L}ind(\mathcal{K}))$-formula, and hence is equivalent to a $\mathbf{Lind}(\mathcal{K})$-formula φ', which concludes the proof. $\square$

2.4 Logics admitting relativization

We say that a fragment $\mathbf{L}$ of $\mathbf{Lind}$ *admits relativization* if Properties $R1$ and $R2$ below hold.

Property $R1$ For all integers $k_1, k_2 \geq 0$ and $k \geq k_1 + k_2$, for each $\mathbf{L}$-sentence φ of rank $k_1 + 1 + k_2$ over an alphabet Σ and for each first-order variable x

without occurrence in φ, there exists an **L**-formula $\varphi[\not\geq x]$ of rank k in the free variable x with the following property. For each tree $t \in \Sigma M_k$ and for each node $v \in \mathsf{NV}(t)$, $(t, x \mapsto v)$ satisfies $\varphi[\not\geq x]$ if and only if

- if s is the subtree of t with root v, then t is of the form $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ (see Figure 6), and
- $r \models \varphi$.

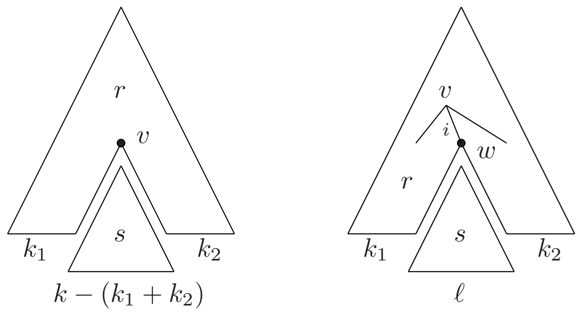

Figure 6: The factorizations of t in Properties $R1$ and $R2$ respectively

Property $R2$ For all ranked alphabet Σ, integer $i \geq 1$ less than or equal to the maximal rank of a letter in Σ and integers $k_1, k_2, \ell \geq 0$, for each rank ℓ **L**-sentence φ and for each first-order variable x without occurrence in φ, there exists an **L**-formula $\varphi[\geq xi]$ of rank $k_1 + \ell + k_2$ over Σ in the free variable x with the following property. For each tree $t \in \Sigma M_{k_1+\ell+k_2}$ and for each node $v \in \mathsf{NV}(t)$, $(t, x \mapsto v)$ satisfies $\varphi[\geq xi]$ if and only if

- the rank of v is greater than or equal to i, and its i-th child, w, has rank ℓ
- if s is the subtree of t with root w, then t is of the form $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$, and $s \models \varphi$ (see Figure 6).

Proposition 2.18 *Let $\mathcal{K}$ be a class of tree languages containing $\mathcal{K}_\exists$ and closed under the following operations. Let $k_1, k_2, \ell \geq 0$, let Δ be a ranked alphabet and let E be a disjoint ranked alphabet such that $\mathsf{card}(E_n) = 1$ if $\Delta_n \neq \emptyset$, and $\mathsf{card}(E_n) = 0$ otherwise: if $K \subseteq \Delta M_{k_1+1+k_2}$ belongs to $\mathcal{K}$, then $K \cdot (\mathbf{k}_1 \oplus EM_\ell \oplus \mathbf{k}_2) \in \mathcal{L}ind(\mathcal{K})$; if $K \subseteq \Delta M_\ell$ belongs to $\mathcal{K}$, then $EM_{k_1+1+k_2} \cdot (\mathbf{k}_1 \oplus K \oplus \mathbf{k}_2) \in \mathcal{L}ind(\mathcal{K})$.*

Then $\mathbf{Lind}(\mathcal{K})$ *admits relativization.*

Proof. We first consider Property $R1$. Let $k_1, k_2 \geq 0$, let $k \geq k_1 + k_2$ and let φ be a $\mathbf{Lind}(\mathcal{K})$ formula over Σ, of rank $k_1 + 1 + k_2$, without any occurrence of x. We show by structural induction on φ that there exists a rank k $\mathbf{Lind}(\mathcal{K})$-formula $\varphi[\not\geq x]$ where x is a free variable and such that, for any tree $t \in \Sigma M_k$, the following holds: if $v \in \mathsf{NV}(t)$ and $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ with the tree s rooted at v, and if $\lambda\colon Y \to \mathsf{NV}(r)$ is an interpretation (where Y is a set containing the free variables of φ and not containing x), then

$$(t, [\lambda; x \mapsto v]) \models \varphi[\not\geq x] \iff (r, \lambda) \models \varphi.$$

If $\varphi = \mathsf{left}_j(y)$ with $j > k_1$, we let $\varphi[\not\geq x] = \mathsf{left}_{j+\ell}(y)$, where $\ell = k - (k_1 + k_2)$. If $\varphi = \mathsf{right}_j(y)$ with $j > k_1 + 1$, we let $\varphi[\not\geq x] = \mathsf{right}_{j+\ell}(y)$. If $\varphi = \max_{i,j}(y)$ with $j > k_1 + 1$, we let $\varphi[\not\geq x] = \mathsf{max}_{i,j+\ell-1}(y)$. And if $\varphi = \max_{i,k_1+1}(y)$, we let $\varphi[\not\geq x] = \mathsf{Succ}_i(y, x)$.

For all other atomic formulas, we let $\varphi[\not\geq x] = \varphi$. It is elementary to verify that these choices guarantee the expected equivalence. Similarly, if $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), we let $\varphi[\not\geq x] = \varphi_1[\not\geq x] \vee \varphi_2[\not\geq x]$ (resp. $\varphi[\not\geq x] = \neg\varphi_1[\not\geq x]$).

Let us now assume that $\varphi = Q_K y \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$ where $K \subseteq \Delta M_{k_1+1+k_2}$ is in $\mathcal{K}$, $y \notin Y \cup \{x\}$ and the φ_δ are deterministic with respect to y. Let E be a ranked alphabet disjoint from Δ, with a single rank n element ε_n for each n such that $\Delta_n \neq \emptyset$; and let $\Delta' = \Delta \cup E$. Let $L = K \cdot (\mathbf{k}_1 \oplus EM_\ell \oplus \mathbf{k}_2)$; then $L \in \mathcal{L}ind(\mathcal{K})$ by assumption. For each $\delta \in \Delta$, we let $\psi_\delta = \neg(y \geq x) \wedge \varphi_\delta[\not\geq x]$; and we let $\psi_\varepsilon = (y \geq x)$ for each $\varepsilon \in E$.[3] We note that the ψ_δ have their free variables in $Y \cup \{x, y\}$. Using the induction hypothesis, one verifies that $\langle\psi_\delta\rangle_{\delta\in\Delta'}$ is deterministic with respect to y, and we let $\psi = Q_L y \cdot \langle\psi_\delta\rangle_{\delta\in\Delta'}$. Then ψ is a $\mathbf{Lind}(\mathcal{L}ind(\mathcal{K}))$-formula, and by Theorem 2.15, there exists an equivalent $\mathbf{Lind}(\mathcal{K})$-formula ψ'.

By the induction hypothesis, for every w in $\mathsf{NV}(r)$ and $\delta \in \Delta$, $(t, [\lambda; x \mapsto v, y \mapsto w]) \models \psi_\delta$ if and only if $(r, [\lambda, y \mapsto w]) \models \varphi_\delta$. Also, $(t, [\lambda; x \mapsto v, y \mapsto w]) \models (y > x)$ for all $w \in \mathsf{NV}(s)$. Thus, the characteristic tree determined by t, $[\lambda; x \mapsto v]$ and the ψ_δ is of the form $\hat{r} \cdot (\mathbf{k}_1 \oplus \hat{s} \oplus \mathbf{k}_2)$, where $\hat{r}$ is the characteristic tree determined by r, λ and the φ_δ, and where each $w \in \mathsf{NV}(\hat{s})$ is labeled in E. Thus, letting $\varphi[\not\geq x] = \psi'$, we have the desired equivalence.

Let us now consider Property $R2$. Let $i \geq 1$, $k_1, k_2, \ell \geq 0$, let $k = k_1 + \ell + k_2$ and let φ be a rank ℓ $\mathbf{Lind}(\mathcal{K})$-formula over Σ without any occurrence of x. We show by structural induction on φ that there exists a rank k $\mathbf{Lind}(\mathcal{K})$-formula $\varphi[\geq xi]$ where x is a free variable and such that, for any tree $t \in \Sigma M_k$, the following holds: if $v \in \mathsf{NV}(t)$, then

$$(t, [\lambda; x \mapsto v]) \models \varphi[\geq xi] \iff \begin{cases} v \text{ has rank at least } i, \\ t \text{ factors as } t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2) \\ (s, \lambda) \models \varphi, \end{cases}$$

where s is the subtree of t rooted at the i-th successor of v and $\lambda\colon Y \to \mathsf{NV}(s)$.

If $\varphi = \mathsf{left}_j(y)$ (resp. $\mathsf{right}_j(y)$, $\mathsf{max}_{h,j}(y)$), we let $\varphi[\geq xi] = \mathsf{left}_{k_1+j}(y)$ (resp. $\mathsf{right}_{k_1+j}(y)$, $\mathsf{max}_{h,k_1+j}(y)$). If $\varphi = \mathsf{root}(y)$, we let $\varphi[\geq xi] = \mathsf{Succ}_i(x, y)$. For all other atomic formulas, we let $\varphi[\geq xi] = \varphi$. If $\varphi = \varphi_1 \vee \varphi_2$ (resp. $\varphi = \neg\varphi_1$), we

[3] To justify this choice of ψ_δ, we need to verify that $y = x$ is expressible: it is equivalent to $\forall z \wedge_{i=1}^{n} \mathsf{Succ}_i(x, z) \leftrightarrow \mathsf{Succ}_i(y, z) \wedge \mathsf{Succ}_i(z, x) \leftrightarrow \mathsf{Succ}_i(z, y)$ where n denotes the maximal rank of a letter in Σ. The presence of a universal quantifier is acceptable since we have assumed that $\mathcal{K}$ contains $\mathcal{K}_\exists$.

take $\varphi[\geq xi] = \varphi_1[\geq xi] \vee \varphi_2[\geq xi]$ (resp. $\varphi[\geq xi] = \neg\varphi_1[\geq xi]$). Again, it is elementary to verify that these choices guarantee the expected equivalence.

Let us now assume that $\varphi = Q_K y \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$ where $K \subseteq \Delta M_\ell$ is in $\mathcal{K}$, $y \notin Y \cup \{x\}$ and the φ_δ are deterministic with respect to y. Let E and Δ' be as in the first part of the proof, and let $L = EM_{k_1+1+k_2} \cdot (\mathbf{k}_1 \oplus K \oplus \mathbf{k}_2)$; then $L \in \mathcal{L}ind(\mathcal{K})$ by assumption.

For each $n \geq 0$ such that $\Sigma_n \neq \emptyset$, let χ_n be the formula $\mathsf{Succ}_i(x,z) \wedge (z \leq y)$ (independent of n), and let $\chi = Q_{K_k(\exists)} z \cdot \langle\chi_n\rangle$. By assumption, χ is a $\mathbf{Lind}(\mathcal{K})$-formula. Moreover, $(t, [x \mapsto v; y \mapsto w])$ satisfies χ if and only if v has rank at least i and w is a descendant of the i-th child of v.

For each $\varepsilon \in E$, let $\psi_\varepsilon = \neg\chi$, and for each $\delta \in \Delta$, let $\psi_\delta = \varphi_\delta[\geq xi] \wedge \chi$. By induction, the ψ_δ ($\delta \in \Delta'$) are $\mathbf{Lind}(\mathcal{K})$-formulas with free variables in $Y \cup \{x, y\}$. Using the induction hypothesis again, one verifies that $\langle\psi_\delta\rangle_{\delta\in\Delta'}$ is deterministic with respect to y, and we let $\psi = Q_L y \cdot \langle\psi_\delta\rangle_{\delta\in\Delta'}$. Then ψ is a $\mathbf{Lind}(\mathcal{L}ind(\mathcal{K}))$-formula, and by Theorem 2.15, there exists an equivalent $\mathbf{Lind}(\mathcal{K})$-formula ψ'.

It follows as above that $(t, [\lambda; x \mapsto v]) \models \psi'$ if and only if the rank of v is at least i and t factors as $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ with $(s, \lambda) \models \varphi$, where s is the subtree of t with root v. □

This can be applied to the classes $\mathcal{K}_\exists$ and $\mathcal{K}_{\exists,\mathrm{mod}}$ discussed in Example 2.13.

Corollary 2.19 *The logics* $\mathbf{Lind}(\mathcal{K}_\exists)$ *and* $\mathbf{Lind}(\mathcal{K}_{\exists,\mathrm{mod}})$ *admit relativization.*

Proof. Let Δ be a ranked Boolean alphabet, let E be a disjoint ranked alphabet as in the statement of Proposition 2.18, and let $\Delta' = \Delta \cup E$. Denote by ε_n the element of rank n in E, if it exists. Let $k_1, k_2, \ell \geq 0$, and $k = k_1 + \ell + k_2$.

Let $K = K_{k_1+1+k_2}(\exists)$ and $L = K \cdot (\mathbf{k}_1 \oplus EM_\ell \oplus \mathbf{k}_2)$. Define, for each $m \geq 0$ such that $\Sigma_m \neq \emptyset$

$$\begin{aligned}
\varphi_m &= P_{1_m}(x),\\
\varphi &= Q_{K_k(\exists)} x \cdot \langle\varphi_n\rangle_n,\\
\chi_m &= (x \leq y) \wedge \neg P_{\varepsilon_m}(y),\\
\chi &= Q_{K_k(\exists)} y \cdot \langle\chi_n\rangle_n,\\
\omega_m &= \neg(x \leq y) \wedge P_{\varepsilon_m}(y),\\
\omega &= Q_{K_k(\exists)} y \cdot \langle\omega_n\rangle_n,\\
\psi_m &= \mathsf{left}_{k_1}(x) \wedge \mathsf{right}_{k_1+\ell+1}(x) \wedge \neg\chi \wedge \neg\omega \text{ and}\\
\psi &= Q_{K_k(\exists)} x \cdot \langle\psi_n\rangle_n.
\end{aligned}$$

Then a tree $t \in \Delta' M_k$ satisfies φ if and only if a letter of the form 1_n occurs at least once in t; $(t, [x \mapsto v])$ satisfies χ (resp. ω) if some descendant (resp. non-descendant) of v has its label in Δ (resp. in E); and t satisfies ψ if and only if t can be factored as $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ with all the nodes in $\mathsf{NV}(s)$ labeled in E and all the nodes in $\mathsf{NV}(r)$ labeled in Δ. It is immediate that L is defined by the $\mathbf{Lind}(\mathcal{K}_\exists)$-formula $\varphi \wedge \psi$.

Now let $K = K_\ell(\exists)$ and $L = EM_{k_1+1+k_2} \cdot (\mathbf{k}_1 \oplus K \oplus \mathbf{k}_2)$. Define, for each $m \geq 0$ such that $\Sigma_m \neq \emptyset$

$$\begin{aligned}
\chi_m &= \neg(x < y) \wedge \neg P_{\varepsilon_m}(y), \\
\chi &= Q_{K_k(\exists)} y \cdot \langle \chi_n \rangle_n, \\
\omega_m &= (x < y) \wedge P_{\varepsilon_m}(y), \\
\omega &= Q_{K_k(\exists)} y \cdot \langle \omega_n \rangle_n, \\
\psi_m &= \mathsf{left}_{k_1}(x) \wedge \mathsf{right}_{k_1+\ell+1}(x) \wedge \neg\chi \wedge \neg\omega \text{ and} \\
\psi &= Q_{K_k(\exists)} x \cdot \langle \psi_n \rangle_n.
\end{aligned}$$

Then if $t \in \Delta' M_k$, we have $(t, [x \mapsto v]) \models \omega$ (resp. χ) if some proper descendant (resp. non proper-descendant) of v has its label in E (resp. in Δ); and t satisfies ψ if and only if t can be factored as $t = r \cdot (\mathbf{k}_1 \oplus s \oplus \mathbf{k}_2)$ with all the nodes in $\mathsf{NV}(r)$ labeled in E and all the nodes in $\mathsf{NV}(s)$ labeled in E. It is immediate that L is defined by the $\mathbf{Lind}(\mathcal{K}_\exists)$-formula $\varphi \wedge \psi$.

The proof that $\mathbf{Lind}(\mathcal{K}_{\exists,\mathrm{mod}})$ admits relativization is similar. □

3 Algebraic characterization of logically defined tree languages

3.1 The block product of preclones

In this section, we introduce our main algebraic tool, the block product of preclones and of *pg*-pairs. This is a generalization of an operation on monoids that was introduced by Rhodes and Tilson [28], as a two-sided generalization of the more classical wreath product.

Let us first (attempt to) briefly summarize the spirit of the block product of monoids, which was introduced [28] in relation with the description of bimachines (Eilenberg [11]). Let T be a finite monoid and let $\tau\colon A^* \to T$ be a morphism. The associated bimachine $\mathcal{T}$ represents the simultaneous operations of left-to-right and right-to-left processing of a string in A^* by τ: if $a_1 \cdots a_n \in A^*$, the i-th component of this processing is the triple $(\tau(a_1 \cdots a_{i-1}), a_i, \tau(a_{i+1} \cdots a_n)) \in T \times A \times T$, and the output of $\mathcal{T}$ is the product of these components, namely the following string in $(T \times A \times T)^*$:

$$(1, a_1, \tau(a_2 \cdots a_n))(\tau(a_1), a_2, \tau(a_3 \cdots a_n)) \ \cdots \ (\tau(a_1 \cdots a_{n-1}), a_n, 1).$$

The idea of the block product is to capture the (cascade) product of this bimachine with an ordinary automaton, that is, to use the output of the bimachine as input for another automaton $\mathcal{S}$ operating on alphabet $T \times A \times T$. This translates to a monoid morphism $\sigma\colon (T \times A \times T)^* \to S$ into a finite monoid S (the transition monoid of $\mathcal{S}$) – which is entirely determined by the images of the triples $(t, a, t') \in T \times A \times T$. For each $a \in A$, let us denote by f_a the map $f_a(t, t') = \sigma(t, a, t')$. Then the composed machine output, on input $a_1 \cdots a_n$ is

$$f_{a_1}(1, \tau(a_2 \cdots a_n)) f_{a_2}(\tau(a_1), \tau(a_3 \cdots a_n)) \ \cdots \ f_{a_n}(\tau(a_1 \cdots a_{n-1}), 1).$$

Note that $f_a(t,t')$ is the σ-image (the S-value) of the effect of letter a in bimachine $\mathcal{T}$, when a is in a left-right context whose T-values are t and t'. The map f_a itself records the effect of letter a in all possible contexts.

The general definition of the block product of monoids is an abstraction of these ideas: $S \,\Box\, T$ is the set of pairs $(f,t) \in S^{T\times T} \times T$ and the product $(f_1,t_1)\cdots(f_n,t_n)$ is equal to $(g, t_1\cdots t_n)$, with

$$g(t,t') = f_1(1, t_2\cdots t_n) f_2(t_1, t_3\cdots t_n) \;\cdots\; f_{n-1}(t_1\cdots t_{n-2}, t_n) f_n(t_1\cdots t_{n-1}, 1).$$

This operation on monoids proved to be useful to decompose morphisms [28, 29, 30] and to explain the connection between first-order logic and aperiodic monoids (see [35]). We now extend these ideas to preclones. The resulting definition is more complex as our contexts are not just left-right pairs (see the definition of contexts in Section 1.3) and we need to take into account the rank of elements. In particular, this leads to the definition of a sequence of block products $S \,\Box_k\, T$ ($k \geq 0$).

Formally, let S and T be preclones. We define preclones $S \,\Box_k\, T$ for each $k \geq 0$. Recall (Section 1.3) that, for each $k, n \geq 0$, $I_{k,n}$ denotes the set of n-ary contexts in T_k. The set of rank n elements of $S \,\Box_k\, T$ is defined to be

$$(S \,\Box_k\, T)_n = S_n^{I_{k,n}} \times T_n, \quad n \geq 0.$$

The identity $\mathbf{1}$ is the pair $(F_{\mathbf{1}}, \mathbf{1})$, where $F_{\mathbf{1}}(C) = \mathbf{1}$, for all $C \in I_{k,1}$. As for the composition operation, let $(F,f) \in (S \,\Box_k\, T)_n$, and let $(G_i, g_i) \in (S \,\Box_k\, T)_{m_i}$ for each $i \in [n]$. Let $g = g_1 \oplus \cdots \oplus g_n \in T_{n,m}$, where $m = \sum_{i=1}^n m_i$. Then we let

$$(F,f)\cdot((G_1,g_1)\oplus\cdots\oplus(G_n,g_n)) = (Q, f\cdot g),$$

an element of $(S \,\Box_k\, T)_m = S_m^{I_{k,m}} \times T_m$, where $Q\colon I_{k,m} \to S_m$ is described as follows.

For each $(u, k_1, v, k_2) \in I_{k,m}$, we have $v = v_1 \oplus \cdots \oplus v_m \in T_{m,\ell}$, where $\ell = k - (k_1 + k_2)$. Let $\bar{v}_1$ be the $\oplus$-sum of the first m_1 v_j's, $\bar{v}_1 = v_1 \oplus \cdots \oplus v_{m_1}$, let $\bar{v}_2$ be the $\oplus$-sum of the next m_2 v_j's, etc, until $\bar{v}_n = v_{m-m_n+1} \oplus \cdots \oplus v_m$ is the $\oplus$-sum of the last m_n v_j's, see Figure 7. In particular, $v = \bigoplus_{i=1}^n \bar{v}_i$. For each $i \in [n]$, let ℓ_i be the total rank of $\bar{v}_i$, so that $\bar{v}_i \in T_{m_i,\ell_i}$ and $\sum_i \ell_i = \ell$.

For each $i \in [n]$, we observe that $g_i \cdot \bar{v}_i \in T_{\ell_i}$, and we let

$$c_i = u \cdot \Big(\mathbf{k}_1 \oplus f\cdot(g_1\cdot\bar{v}_1 \oplus\cdots\oplus g_{i-1}\cdot\bar{v}_{i-1} \oplus \mathbf{1} \oplus g_{i+1}\cdot\bar{v}_{i+1} \oplus\cdots\oplus g_n\cdot\bar{v}_n) \oplus \mathbf{k}_2\Big).$$

We note that $u\cdot(\mathbf{k}_1 \oplus f\cdot g\cdot v \oplus \mathbf{k}_2) = c_i \cdot (\mathbf{p}_1 \oplus g_i\cdot\bar{v}_i \oplus \mathbf{p}_2)$, where $p_1 = k_1 + \sum_{j<i}\ell_j$ and $p_2 = \sum_{j>i}\ell_j + k_2$. Then c_i is an element of T with rank $p_1 + 1 + p_2 = k_1 + k_2 + \ell - \ell_i + 1 = k - \ell_i + 1$. (Of course, the integers p_1 and p_2 depend on i even though our notation does not show it.)

In particular, $C_i = (c_i, p_1, \bar{v}_i, p_2)$ is a context in I_{k,m_i}, see Figure 7. We are finally ready to define Q:

$$Q(u,k_1,v,k_2) = F(u,k_1,g\cdot v,k_2)\cdot(G_1(C_1)\oplus\cdots\oplus G_n(C_n)).$$

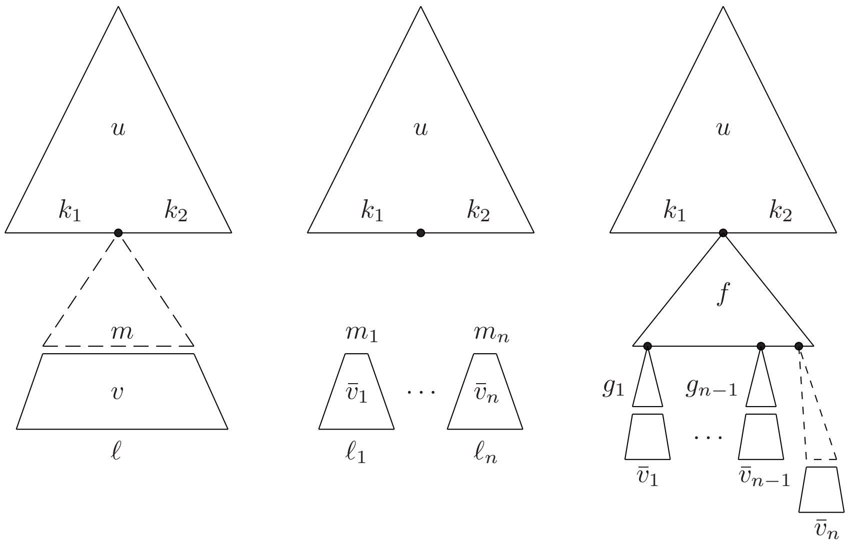

Figure 7: Two views of $(u, k_1, k_2, v) \in I_{k,m}$, and the context C_n

Lemma 3.1 *The above definition satisfies the axioms of preclones.*

Proof. Let us first verify the axioms concerning the identity element. Let $(G, g) \in (S \,\square_k\, T)_m$ and let $(Q, g) = (F_{\mathbf{1}}, \mathbf{1}) \cdot (G, g)$. Let $(u, k_1, v, k_2) \in I_{k,m}$. With reference to the notation in the definition above, we have $n = 1$ and $C_1 = (u, k_1, v, k_2)$. It follows that $Q = G$, so $(F_{\mathbf{1}}, \mathbf{1}) \cdot (G, g) = (G, g)$.

Let now $(F, f) \in (S \square_k T)_n$ and let $(Q, f) = (F, f) \cdot ((F_{\mathbf{1}}, \mathbf{1}) \oplus \cdots \oplus (F_{\mathbf{1}}, \mathbf{1}))$. Let $(u, k_1, v, k_2) \in I_{k,n}$. Then $Q(u, k_1, v, k_2) = F(u, k_1, v, k_2) \cdot (F_{\mathbf{1}}(C_1) \oplus \cdots \oplus F_{\mathbf{1}}(C_n))$ for some $C_1, \ldots, C_n$, and hence $Q(u, k_1, v, k_2) = F(u, k_1, v, k_2) \cdot (\mathbf{1} \oplus \cdots \oplus \mathbf{1}) = F(u, k_1, v, k_2)$. Thus $(F, f) \cdot ((F_{\mathbf{1}}, \mathbf{1}) \oplus \cdots \oplus (F_{\mathbf{1}}, \mathbf{1})) = (F, f)$.

Next let $(F, f) \in (S \,\square_k\, T)_n$; for $i \in [n]$ let $(G_i, g_i) \in (S \,\square_k\, T)_{m_i}$ and let $m = \sum_{i \in [n]} m_i$; for $j \in [m]$, let $(H_j, h_j) \in (S \,\square_k\, T)_{p_j}$ and let $p = \sum_{j \in [m]} p_j$.

Let $g = \oplus_{i \in [n]} g_i$ and let $h = \oplus_{j \in [m]} h_j$. We also denote by $\bar{h}_1$ the $\oplus$-sum of the first m_1 h_j's, $\bar{h}_2$ the $\oplus$-sum of the next m_2 h_j's, etc, to $\bar{h}_n$ the $\oplus$-sum of the last m_n h_j's, so that $h = \oplus_{i \in [n]} \bar{h}_i$. The rank of $\bar{h}_i$ is $\sum_{j=m_1+\cdots+m_{i-1}+1}^{m_1+\cdots+m_i} p_j$.

We need to consider p-ary contexts in T_k: let $(u, k_1, v, k_2) \in I_{k,p}$ be such a context. Then $v = v_1 \oplus \cdots \oplus v_p \in T_{p,\ell}$ with $\ell = k - k_1 - k_2$. Let $\bar{v}_1$ denote the $\oplus$-sum of the first p_1 v_i's, $\bar{v}_2$ the $\oplus$-sum of the next p_2 v_i's, etc to $\bar{v}_m$ the $\oplus$-sum of the last p_m v_i's. For $i \in [n]$, we also denote by $\bar{\bar{v}}_i$ the $\oplus$-sum of the $\bar{v}_j$ where h_j is part of the summation defining $\bar{h}_i$. That is, $\bar{\bar{v}}_1 = \bar{v}_1 \oplus \cdots \oplus \bar{v}_{m_1}, \ldots,$ $\bar{\bar{v}}_n = \bar{v}_{m-m_n+1} \oplus \cdots \oplus \bar{v}_m$.

We first consider the product

$$
\begin{aligned}
&\Big((F, f) \cdot \big((G_1, g_1) \oplus \cdots \oplus (G_n, g_n)\big)\Big) \cdot \big((H_1, h_1) \oplus \cdots \oplus (H_m, h_m)\big) \\
= \quad &(Q, f \cdot g) \cdot \big((H_1, h_1) \oplus \cdots \oplus (H_m, h_m)\big)
\end{aligned}
$$

$$= \quad (R, f \cdot g \cdot h).$$

Then we have $R(u, k_1, v, k_2) = Q(u, k_1, h \cdot v, k_2) \cdot (H_1(B_1) \oplus \cdots \oplus H_m(B_m))$, with $B_j = (b_j, p'_1, \bar{v}_j, p'_2)$ where b_j $(j \in [m])$ is

$$u \cdot (\mathbf{k}_1 \oplus f \cdot g \cdot (h_1 \cdot \bar{v}_1 \oplus \cdots \oplus h_{j-1} \cdot \bar{v}_{j-1} \oplus \mathbf{1} \oplus h_{j+1} \cdot \bar{v}_{j+1} \oplus \cdots \oplus h_m \cdot \bar{v}_m) \oplus \mathbf{k}_2),$$

$p'_1 = k_1 + \sum_{s=1}^{j-1} \mathsf{rank}(\bar{v}_s)$ and $p'_2 = k_2 + \sum_{s=j+1}^{m} \mathsf{rank}(\bar{v}_s)$, so that

$$u \cdot (\mathbf{k}_1 \oplus f \cdot g \cdot h \cdot v \oplus \mathbf{k}_2) = b_j \cdot (\mathbf{p}'_1 \oplus h_j \cdot \bar{v}_j \oplus \mathbf{p}'_2).$$

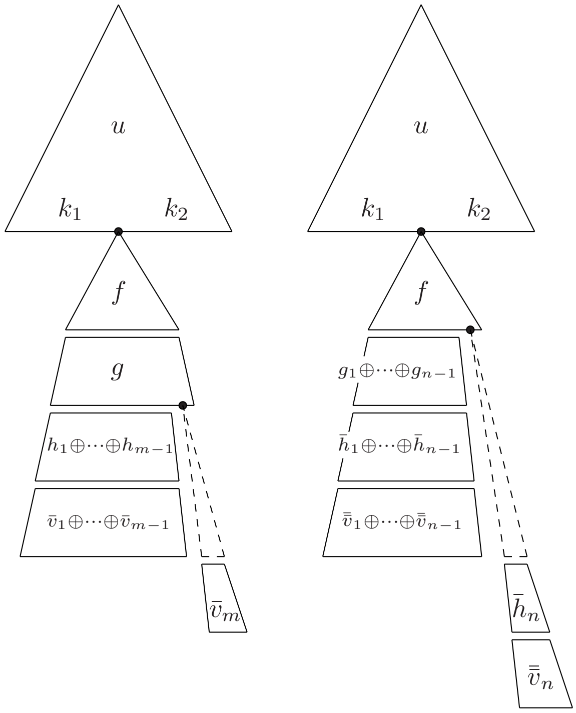

Figure 8: The contexts B_m, C_n and D_n

Moreover, $Q(u, k_1, h \cdot v, k_2) = F(u, k_1, g \cdot h \cdot v, k_2) \cdot (G_1(C_1) \oplus \cdots \oplus G_n(C_n))$, where C_i $(i \in [n])$ is the context $(c_i, r_1, \bar{h}_i \cdot \bar{\bar{v}}_i, r_2)$ with c_i equal to

$$u \cdot \big(\mathbf{k}_1 \oplus f \cdot (g_1 \cdot \bar{h}_1 \cdot \bar{\bar{v}}_1 \oplus \cdots \oplus g_{i-1} \cdot \bar{h}_{i-1} \cdot \bar{\bar{v}}_{i-1} \oplus \mathbf{1} \oplus g_{i+1} \cdot \bar{h}_{i+1} \cdot \bar{\bar{v}}_{i+1} \oplus \cdots \oplus g_n \cdot \bar{h}_n \cdot \bar{\bar{v}}_n) \oplus \mathbf{k}_2\big),$$

$r_1 = k_1 + \sum_{s=1}^{i-1} \mathsf{rank}(\bar{\bar{v}}_s)$, $r_2 = k_2 + \sum_{s=i+1}^{n} \mathsf{rank}(\bar{\bar{v}}_s)$, so that

$$u \cdot (\mathbf{k}_1 \oplus f \cdot g \cdot h \cdot v \oplus \mathbf{k}_2) = c_i \cdot (\mathbf{r}_1 \oplus g_i \cdot \bar{h}_i \cdot \bar{\bar{v}}_i \oplus \mathbf{r}_2).$$

See Figure 8. Thus

$$R(u,k_1,v,k_2) = F(u,k_1,g\cdot h\cdot v,k_2)\cdot\Big(\bigoplus_{i=1}^{n} G_i(C_i)\Big)\cdot\Big(\bigoplus_{j=1}^{m} H_j(B_j)\Big).$$

We compare this result with the product

$$\begin{aligned}&(F,f)\cdot\Big(\big((G_1,g_1)\oplus\cdots\oplus(G_n,g_n)\big)\cdot\big((H_1,h_1)\oplus\cdots\oplus(H_m,h_m)\big)\Big)\\ =\ &(F,f)\cdot\big((Q'_1,g_1\cdot\bar h_1)\oplus\cdots\oplus(Q'_n,g_n\cdot\bar h_n)\big)\\ =\ &(R',f\cdot g\cdot h).\end{aligned}$$

Then we have $R'(u,k_1,v,k_2) = F(u,k_1,g\cdot h\cdot v,k_2)\cdot(Q'_1(D_1)\oplus\cdots\oplus Q'_n(D_n))$, where D_i ($i\in[n]$) is the context $(c_i,r_1,\bar{\bar v}_i,r_2)$, where c_i,r_1,r_2 are defined above.

Next we compute $Q'_1(D_1)$: we have

$$Q'_1(D_1) = Q'_1(c_1,r_1,\bar{\bar v}_1,r_2) = G(c_1,r_1,\bar h_1\cdot\bar{\bar v}_1,r_2)\cdot(H_1(E_1)\oplus\cdots\oplus H_{m_1}(E_{m_1}))$$

where E_j ($j\in[m_1]$) is the context $(e_j,r'_{1,j},\bar v_j,r'_{2,j})$,

$$\begin{aligned}e_j\ =\ &u\cdot(\mathbf{k}_1\oplus f\cdot(g_1\cdot(h_1\cdot\bar v_1\oplus\cdots\oplus h_{j-1}\cdot\bar v_{j-1}\oplus\mathbf{1}\\ &\oplus h_{j+1}\cdot\bar v_{j+1}\oplus\cdots\oplus h_{m_1}\cdot\bar v_{m_1})\oplus g_2\cdot\bar h_2\cdot\bar{\bar v}_2\oplus\cdots\oplus g_n\cdot\bar h_n\cdot\bar{\bar v}_n)\oplus\mathbf{k}_2)\end{aligned}$$

and $r'_{1,j}$ and $r'_{2,j}$ are appropriate integers so that $c_1\cdot(\mathbf{r}_1\oplus g_1\cdot\bar h_1\cdot\bar{\bar v}_1\oplus\mathbf{r}_2) = e_j\cdot(\mathbf{r}_{1,j}\oplus h_j\cdot\bar v_j\oplus\mathbf{r}_{2,j})$, see Figure 8.

We observe now that $e_j = b_j$ and $E_j = B_j$ for $j\in[m_1]$. So we have $Q'_1(D_1) = G_1(C_1)\cdot\bigoplus_{j=1}^{m_1} H_j(B_j)$.

Similarly, for each $i\in[n]$, we have

$$Q'_i(D_i) = G_i(C_i)\cdot\bigoplus_{j=1}^{m_i} H_{m_1+\cdots+m_{i-1}+j}(B_{m_1+\cdots+m_{i-1}+j}),$$

and we have verified that $R = R'$. □

We also define block products of *pg*-pairs. If (S,A) and (T,B) are *pg*-pairs and $k\ge 0$, we define $(S,A)\,\Box_k\,(T,B)$ to be the sub-*pg*-pair of $S\,\Box_k\,T$ generated by those pairs (F,g) such that for some $n\ge 0$, $g\in B_n$ and $F(c)\in A_n$ for each $c\in I_{k,n}$.

Let (S,A) and (T,B) be *pg*-pairs and let $\alpha\colon AM\to S$ and $\beta\colon BM\to T$ be the natural morphisms, so that $\alpha(a)=a$ and $\beta(b)=b$ for all $a\in A$ and $b\in B$. Let $(U,\Sigma) = (S,A)\,\Box_k\,(T,B)$, and let $\varphi\colon\Sigma M\to U\subseteq S\,\Box_k\,T$ be the natural morphism. By definition, each $\sigma\in\Sigma_n$ ($n\ge 0$) is a pair $\sigma = (F_\sigma,b_\sigma)$ with $b_\sigma\in B_n$ and $F_\sigma\in A_n^{I_{k,n}}$. Let $\pi\colon\Sigma M\to BM$ be the morphism induced by the second component projection from Σ to B, and let $\tau=\beta\circ\pi\colon\Sigma M\to T$, see Figure 9. We now describe a way of computing $\varphi(t)$ for a tree $t\in\Sigma M_n$, say $\varphi(t) = (Q_t,\tau(t))$.

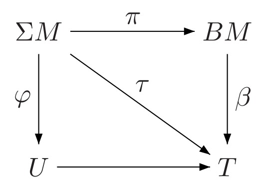

Figure 9: The morphisms φ, π and β

Let $D = (u, k_1, v, k_2) \in I_{k,n}$. We define the tree $\bar{t}_D$ by relabeling the nodes of t in $\mathsf{NV}(t)$ with elements of A as follows. Let x be a node of t of rank $m \geq 0$, let g be the subtree of t whose root is x, and let $h_1, \ldots, h_m$ be the subtrees whose roots are the children of x, see Figure 10. Let $h = h_1 \oplus \cdots \oplus h_m$, let $\sigma \in \Sigma_m$ be the label of x in t and let $r_2 \geq 0$ be such that $g = \sigma \cdot h \in \Sigma M_{r_2}$ and $h \in \Sigma M_{m,r_2}$.

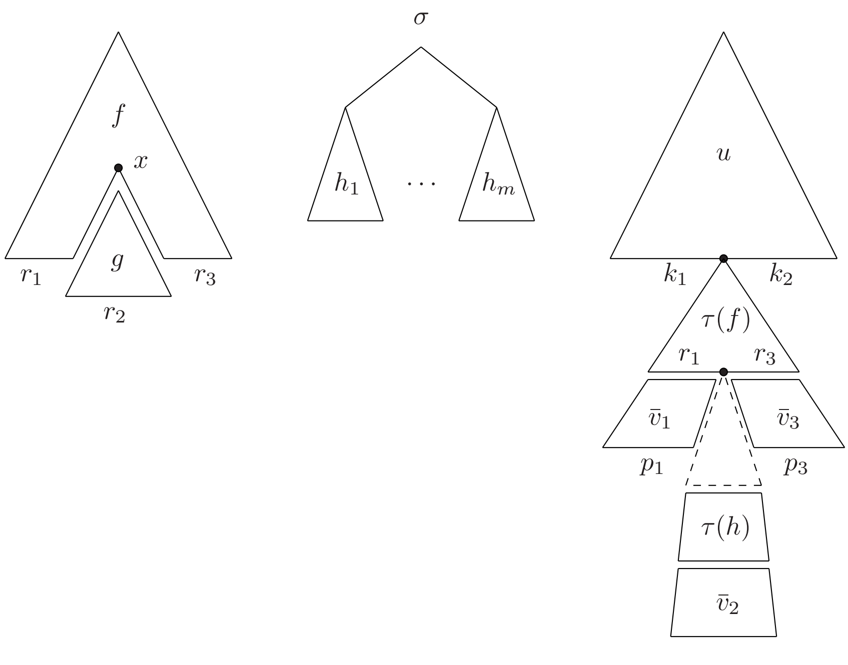

Figure 10: The trees t and $g = \sigma \cdot h$, and the context C

Let us write $t = f \cdot (\mathbf{r}_1 \oplus g \oplus \mathbf{r}_3)$, where r_1 and r_3 are integers such that the node x is now labeled by a variable in f (that is, $\mathsf{left}_{r_1}(x)$ and $\mathsf{right}_{n-r_3+1}(x)$ in t, and $n = r_1 + r_2 + r_3$). Let $\bar{v}_1$ be the $\oplus$-sum of the first r_1 v_j's, $\bar{v}_2$ the $\oplus$-sum of the next r_2 v_j's and $\bar{v}_3$ the $\oplus$-sum of the last r_3 v_j's. Then we have $\bar{v}_1 \in T_{r_1,p_1}$, $\bar{v}_2 \in T_{r_2,p_2}$ and $\bar{v}_3 \in T_{r_3,p_3}$ for some $p_1, p_2, p_3 \geq 0$ (and $k = k_1 + p_1 + p_2 + p_3 + k_2$). Let then $c = u \cdot (\mathbf{k}_1 \oplus \tau(f) \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_3) \oplus \mathbf{k}_2)$, so that $C = (c, k_1 + p_1, \tau(h) \cdot \bar{v}_2, p_3 + k_2) \in I_{k,m}$, see Figure 10. We finally label the node x in $\bar{t}_D$ by $F_\sigma(C)$.

The resulting tree $\bar{t}_D$ is an element of AM_n. We now show the following fact.

Fact 3.2 *With the notation above, $\varphi(t) = (Q_t, \tau(t))$ where $Q_t(D) = \alpha(\bar{t}_D)$ for each context $D \in I_{k,n}$.*

Proof. The proof is by structural induction on t. If $t = \mathbf{1}$, then $\bar{t}_D = \mathbf{1}$ for each D, and $\varphi(t) = (F_{\mathbf{1}}, \mathbf{1})$, so the announced result holds.

If t consists of a single node, then $t = \sigma \in \Sigma_0$ and $\varphi(t) = (F_\sigma, b_\sigma)$. Now let $D = (u, k_1, \mathbf{0}, k_2) \in I_{k,0}$. With the notation above, we have $g = \sigma = t$, $h = \mathbf{0}$, $f = \mathbf{1}$, and $p_i = r_j = 0$. In particular, $c = u$. It follows that $C = D$ and $\bar{t}_D = F_\sigma(D)$. Moreover, since $F_\sigma(D) \in A$, we have $\alpha(\bar{t}_D) = F_\sigma(D)$. This concludes the verification of the equality for one-node trees.

Let us now assume that $t \in \Sigma M_n$ $(n \geq 0)$ has more than one node, let $\sigma \in \Sigma_m$ be the label of the root of t, and let $s^{(1)}, \ldots, s^{(m)}$ be the subtrees of t attached to the children of the root. Let also $s = s^{(1)} \oplus \cdots \oplus s^{(m)}$, so that $t = \sigma \cdot s$. Let $D = (u, k_1, w, k_2) \in I_{k,n}$. By induction, we have

$$Q_t(D) = F_\sigma(u, k_1, \tau(s) \cdot w, k_2) \cdot \Big(\alpha(\bar{s}^{(1)}_{C_1}) \oplus \cdots \oplus \alpha(\bar{s}^{(m)}_{C_m})\Big),$$

where $w = \bar{w}_1 \oplus \cdots \oplus \bar{w}_n$, $C_i = (c_i, q_1, \bar{w}_i, q_2)$,

$$c_i = u \cdot (\mathbf{k}_1 \oplus \tau(\sigma \cdot (s^{(1)} \oplus \cdots \oplus s^{(i-1)} \oplus \mathbf{1} \oplus s^{(i+1)} \oplus \cdots \oplus s^{(m)})) \oplus \mathbf{k}_2)$$

and q_1 and q_2 are appropriate integers (which depend on i) such that

$$u \cdot (\mathbf{k}_1 \oplus \tau(t) \cdot w \oplus \mathbf{k}_2) = c_i \cdot (\mathbf{q}_1 \oplus \tau(s^{(i)}) \cdot \bar{w}_i \oplus \mathbf{q}_2).$$

We compare this value with $\alpha(\bar{t}_D)$. If a is the label of the root of $\bar{t}_D$ and if $d_1, \ldots, d_m$ are the subtrees of $\bar{t}_D$ attached to the children of the root, then $\alpha(\bar{t}_D) = \alpha(a) \cdot \bigoplus_i \alpha(d_i)$. We first discuss the value of a. With reference to the notation in the definition of the labels of $\bar{t}_D$ above, since $t = \sigma \cdot s$, the integers p_1, p_3, r_1, r_3 are all equal to 0 and $\bar{v}_1 = \bar{v}_3 = \mathbf{0}$. In particular, $a = F_\sigma(u, k_1, \tau(s) \cdot w, k_2)$. Thus $\alpha(a) = \alpha(F_\sigma(u, k_1, \tau(s) \cdot w, k_2)) = F_\sigma(u, k_1, \tau(s) \cdot w, k_2)$.

To conclude, we need only to verify that $d_i = \bar{s}^{(i)}_{C_i}$ for each $i \in [m]$, that is, each node x of t in $\mathsf{NV}(s^{(i)})$, has the same label in d_i and in $\bar{s}^{(i)}_{C_i}$. But it is easy to see that the label of x in both $\bar{t}_D$ and $\bar{s}^{(i)}_{C_i}$ is of the form $F_\rho(C)$ where $\rho \in \Sigma$ is the label of x in $s^{(i)}$ and C is appropriate. □

3.2 Closed pseudovarieties

We say that a pseudovariety $\mathbf{V}$ of preclones is *closed* if every block product $S \Box_k T$ with $S, T \in \mathbf{V}$ and $k \geq 0$ belongs to $\mathbf{V}$. Closed pseudovarieties of *pg*-pairs are defined similarly. Since the intersection of a family of closed pseudovarieties is closed, there exists a least closed pseudovariety containing any given class $\mathbf{K}$ of finitary preclones (resp. finitary *pg*-pairs).

We now give a technical result on closed pseudovarieties, that will be used in the proof of our main result. We consider the situation where S, T, T' are preclones and T is a sub-preclone of T'. Then the elements of $S \square_k T'$ whose second component belongs to T, form a sub-preclone of $S \square_k T'$ which we denote by $S \square_k^{T'} T$.

Proposition 3.3 *Let* $\mathbf{V}$ *be a closed pseudovariety of preclones. Let* $S, T \in \mathbf{V}$ *and let* T' *be a finitary preclone such that* T *is a sub-preclone of* T'*. For each* $k \geq 0$*, the product* $S \square_k^{T'} T$ *belongs to* $\mathbf{V}$*.*

Before we prove Proposition 3.3, we verify a technical lemma. In this lemma, we use the notation in the proposition. In particular, we need to consider contexts in both T_k and T'_k. We denote by $I_{k,n}$ (resp. $I'_{k,n}$) the set of n-ary contexts in T_k (resp. T'_k).

Lemma 3.4 *Let* S*,* T*,* T' *and* k *be as in Proposition 3.3 and let* $C \in I'_{k,n}$*. There exists a morphism* $\alpha^C\colon S \square_k^{T'} T \to S \square_n T$ *such that, if* $(F, f) \in (S \square_k^{T'} T)_n$*, then* $\alpha^C(F, f)$ *is of the form* $\alpha^C(F, f) = (F^C, f)$ *with* $F^C(\mathbf{1}, 0, \mathbf{n}, 0) = F(C)$*.*

Proof. Let $C = (u, k_1, v, k_2) \in I'_{k,n}$ and let $(F, f) \in (S \square_k^{T'} T)_m$, $m \geq 0$. We first define a mapping $F^C\colon I_{n,m} \to S_m$. Let $D = (r, p_1, s, p_2) \in I_{n,m}$. By definition, $p_1 + p_2 \leq n$, and we let $\bar{v}_1$ be the $\oplus$-sum of the first p_1 v_i's, $\bar{v}_2$ be the $\oplus$-sum of the last p_2 v_i's, and $\bar{v}$ be the $\oplus$-sum of the middle $n - p_1 - p_2$ v_i's. In particular, there exist integers q_1, q, q_2 such that $\bar{v}_1 \in T_{p_1,q_1}$, $\bar{v} \in T_{n-p_1-p_2,q}$, $\bar{v}_2 \in T_{p_2,q_2}$ and $k_1 + p_1 + q_1 + q + q_2 + p_2 + k_2 = k$, see Figure 11. We let

$$F^C(D) = F\big(u \cdot (\mathbf{k}_1 \oplus r \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_2) \oplus \mathbf{k}_2), k_1 + q_1, s \cdot \bar{v}, q_2 + k_2\big).$$

The verification that $F^C(\mathbf{1}, 0, \mathbf{n}, 0) = F(C)$ is straightforward, and we need to show that $\alpha^C\colon (F, f) \mapsto (F^C, f)$ defines a morphism of preclones.

Let $(F, f) \in (S \square_k^{T'} T)_m$ and let $(G_i, g_i) \in (S \square_k^{T'} T)_{h_i}$ ($i \in [m]$). For convenience, we let g be the $\oplus$-sum of the g_i, so $g \in T_{m,h}$ with $h = \sum_i h_i$. Let $(Q, f \cdot g) = (F, f) \cdot \bigoplus_i (G_i, g_i)$, so that $\alpha^C((F, f) \cdot \bigoplus_i (G_i, g_i)) = (Q^C, f \cdot g)$. Moreover, let $(R, f \cdot g) = (F^C, f) \cdot \bigoplus_i (G_i^C, g_i)$. We need to verify that $Q^C = R$.

Let $D = (r, p_1, s, p_2) \in I_{h,n}$. For each $i \in [m]$, let $\bar{s}_i$ be the appropriate $\oplus$-sum of s_j's such that $g \cdot s = g_1 \cdot \bar{s}_1 \oplus \cdots \oplus g_m \cdot \bar{s}_m$. Then $R(D) = F^C(r, p_1, g \cdot s, p_2) \cdot \bigoplus_i G_i^C(D_i)$, where $D_i = (r_i, \ell_1, \bar{s}_i, \ell_2)$, r_i is

$$r \cdot (\mathbf{p}_1 \oplus f \cdot (g_1 \cdot \bar{s}_1 \oplus \cdots \oplus g_{i-1} \cdot \bar{s}_{i-1} \oplus \mathbf{1} \oplus g_{i+1} \cdot \bar{s}_{i+1} \oplus \cdots \oplus g_m \cdot \bar{s}_m) \oplus \mathbf{p}_2),$$

and ℓ_1, ℓ_2 are such that $r \cdot (\mathbf{p}_1 \oplus f \cdot g \cdot s \oplus \mathbf{p}_2) = r_i \cdot (\ell_1 \oplus g_i \cdot \bar{s}_i \oplus \ell_2)$.

Thus $R(D)$ is the composition of

$$F\big(u \cdot (\mathbf{k}_1 \oplus r \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_2) \oplus \mathbf{k}_2), k_1 + p_1, g \cdot s \cdot \bar{v}, p_2 + k_2\big)$$

with

$$\bigoplus_{i=1}^{m} G_i\big(u \cdot (\mathbf{k}_1 \oplus r_i \cdot (\bar{v}_1^i \oplus \mathbf{1} \oplus \bar{v}_2^i) \oplus \mathbf{k}_2), k_1 + \ell_1, g \cdot s \cdot \bar{v}^i, \ell_2 + k_2\big),$$

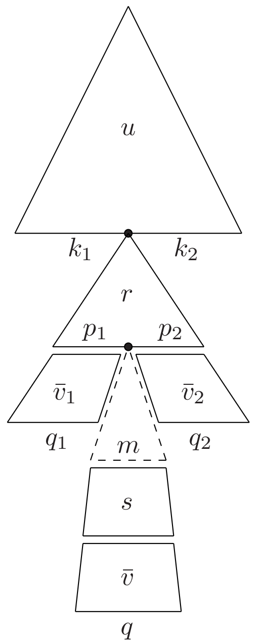

Figure 11: $F^C(D)$ is the image by F of the context represented here

where $v = \bar{v}_1^i \oplus \bar{v}^i \oplus \bar{v}_2^i$ is the appropriate grouping. Observe that

$$u \cdot (\mathbf{k}_1 \oplus r_i \cdot (\bar{v}_1^i \oplus \mathbf{1} \oplus \bar{v}_2^i) \oplus \mathbf{k}_2)$$

is equal to

$$u \cdot (\mathbf{k}_1 \oplus \mathbf{p}_1 \oplus f \cdot (g_1 \cdot \bar{s}_1 \oplus \cdots \oplus g_{i-1} \cdot \bar{s}_{i-1} \oplus \mathbf{1} \oplus g_{i+1} \cdot \bar{s}_{i+1} \oplus \cdots \oplus g_m \cdot \bar{s}_m) \oplus \mathbf{p}_2 \oplus \mathbf{k}_2).$$

Now, $Q^C(D) = Q(u \cdot (\mathbf{k}_1 \oplus r \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_2) \oplus \mathbf{k}_2), k_1 + p_1, s \cdot \bar{v}, p_2 + k_2)$. By definition of the block product, this is equal to the composition of

$$F\big(u \cdot (\mathbf{k}_1 \oplus r \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_2) \oplus \mathbf{k}_2), k_1 + p_1, g \cdot s \cdot \bar{v}, p_2 + k_2\big)$$

with

$$\bigoplus_{i=1}^{m} G_i(r_i', z_1, \bar{s}_i \cdot \bar{v}_i', z_2),$$

where $\bar{v} = \bigoplus_i \bar{v}_i'$ is the appropriate grouping,

$$\begin{aligned} r_i' \quad = \quad & u \cdot (\mathbf{k}_1 \oplus r \cdot (\bar{v}_1 \oplus f \cdot (g_1 \cdot \bar{s}_1 \cdot \bar{\bar{v}}_1 \oplus \cdots \oplus g_{i-1} \cdot \bar{s}_{i-1} \cdot \bar{\bar{v}}_{i-1} \oplus \mathbf{1} \\ & \oplus g_{i+1} \cdot \bar{s}_{i+1} \cdot \bar{\bar{v}}_{i+1} \oplus \cdots \oplus g_m \cdot \bar{s}_m \cdot \bar{\bar{v}}_m) \oplus \bar{v}_2) \oplus \mathbf{k}_2) \end{aligned}$$

where $v = \bar{\bar{v}}_1 \oplus \cdots \oplus \bar{\bar{v}}_m$ and z_1 and z_2 are appropriate integers.

It is now a straightforward verification that $R(D) = Q^C(D)$, which concludes the proof. □

Proof of Proposition 3.3. By Proposition 1.5, it suffices to verify that distinct elements of equal rank in $S \,\Box_k^{T'}\, T$ can be separated by a morphism into an element of $\mathbf{V}$.

So we consider (F_1, f_1) and (F_2, f_2), rank n elements of $S \,\Box_k^{T'}\, T$. If $f_1 \neq f_2$, the second component projection is a morphism into $T \in \mathbf{V}$ which separates the two elements. If $f_1 = f_2$, then $F_1 \neq F_2$ and we let $C \in I'_{k,n}$ such that $F_1(C) \neq F_2(C)$. Then the morphism α^C in Lemma 3.4 is a morphism into $S \,\Box_n\, T \in \mathbf{V}$ which separates the two elements. $\Box$

If (S, A) and (T', B') are *pg*-pairs and (T, B) is a sub-*pg*-pair of (T', B'), recall that the block product $(S, A) \Box_k (T', B')$ is generated by a ranked alphabet Σ' such that $\Sigma'_m = A_m^{I'_{k,m}} \times B'_m$. We let $(S, A) \,\Box_k^{(T',B')}\, (T, B)$ be the sub-*pg*-pair of $(S, A) \Box_k (T', B')$ generated by the subset Σ of Σ' such that $\Sigma_m = A_m^{I'_{k,m}} \times B_m$ for each m.

Proposition 3.5 *Let $\mathbf{V}$ be a closed pseudovariety of pg-pairs. Let (S, A) and (T, B) be pg-pairs in $\mathbf{V}$ and let (T', B') be a finitary pg-pair such that (T, B) is a sub-pg-pair of (T', B'). For each $k \geq 0$, the product $(S, A) \,\Box_k^{(T',B')}\, (T, B)$ belongs to $\mathbf{V}$.*

Proof. We note that the morphism α^C in the proof of Lemma 3.4, maps each generator of $(S, A) \,\Box_k^{(T',B')}\, (T, B)$ to a generator of $(S, A) \,\Box_n\, (T, B)$, so α^C is also a morphism of *pg*-pairs between these block products.

The same scheme as in the proof of Proposition 3.3 can then be applied, using Proposition 1.6 instead of Proposition 1.5. $\Box$

We conclude with a result on full pseudovarieties (see Section 1.4).

Proposition 3.6 *Let $\mathbf{W}$ be a pseudovariety of preclones, let $\mathbf{V} = \mathsf{pgp}(\mathbf{W})$ and let $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$ be the closure of $\mathbf{W}$ and $\mathbf{V}$ respectively. Then $\mathbf{V}$ and $\widehat{\mathbf{V}}$ are full and $\widehat{\mathbf{V}} = \mathsf{pgp}(\widehat{\mathbf{W}})$.*

Proof. In view of Proposition 1.9, it suffices to show that $\widehat{\mathbf{V}} = \mathsf{pgp}(\widehat{\mathbf{W}})$. We first verify that if $(S, A) \in \widehat{\mathbf{V}}$, then $S \in \widehat{\mathbf{W}}$, by induction on the construction of (S, A) from elements of $\mathbf{V}$ by means of block products. If $(S, A) \in \mathbf{V}$, then $S \in \mathbf{W}$ by definition and hence, $S \in \widehat{\mathbf{W}}$. Now suppose that $(S, A) < (S^{(1)}, A^{(1)}) \,\Box_k\, (S^{(2)}, A^{(2)})$ and $S^{(1)}, S^{(2)} \in \widehat{\mathbf{W}}$. By definition of the block product of *pg*-pairs, $S < S^{(1)} \,\Box_k\, S^{(2)}$ and hence $S \in \widehat{\mathbf{W}}$.

Now we show that if (S, A) is a finitary *pg*-pair with $S \in \widehat{\mathbf{W}}$, then $(S, A) \in \widehat{\mathbf{V}}$. The proof is by induction on the construction of S from elements of $\mathbf{W}$ by means of block products. If $S \in \mathbf{W}$, then $(S, A) \in \mathsf{pgp}(\mathbf{W}) = \mathbf{V}$ by definition. Now suppose that $S = S^{(1)} \,\Box_k\, S^{(2)}$ and $\mathsf{pgp}(S^{(i)}) \subseteq \widehat{\mathbf{V}}$. Let $B^{(2)}$ be the projection of A onto $S^{(2)}$. Each element of A is of the form (F, b) with $b \in B^{(2)}$. Let $B^{(1)}$ be the union of the ranges of the first components of elements of A. Then, if $T^{(i)}$ is the sub-preclone of $S^{(i)}$ generated by $B^{(i)}$, we have $(S, A) \subseteq (T^{(1)}, A^{(1)}) \,\Box_k\, (T^{(2)}, A^{(2)})$. It follows that $(S, A) \in \widehat{\mathbf{V}}$. $\Box$

3.3 Characterizing $\mathcal{L}ind(\mathcal{K})$

Our main result is:

Theorem 3.7 *Let $\mathcal{K}$ be a class of recognizable tree languages such that each 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 $\mathcal{L}ind(\mathcal{K})$ if and only if its syntactic pg-pair belongs to the least closed pseudovariety of pg-pairs containing the syntactic pg-pairs of the languages in $\mathcal{K}$.*

The proof of Theorem 3.7 is based on Propositions 3.8 and 3.9 below. Applications are considered in the next section.

Proposition 3.8 *Let $\mathcal{K}$ be a class of recognizable tree languages such that* $\mathbf{Lind}(\mathcal{K})$ *admits relativization and let (S, A) and (T, B) be pg-pairs such that every language recognizable by (S, A) or (T, B) belongs to $\mathcal{L}ind(\mathcal{K})$. Then every language recognizable by a block product of (S, A) and (T, B) also belongs to $\mathcal{L}ind(\mathcal{K})$.*

Proof. Let $k \geq 0$, $(U, \Sigma) = (S, A) \square_k (T, B)$, and $\varphi: (\Sigma M, \Sigma) \to (U, \Sigma)$ be the morphism induced by the identity map of Σ. Let L be a tree language recognized by a morphism $\varphi': (\Sigma' M, \Sigma') \to (U, \Sigma)$. Since φ is onto and $(\Sigma' M, \Sigma')$ is free, there exists a morphism $\psi: (\Sigma' M, \Sigma') \to (\Sigma M, \Sigma)$ such that $\varphi' = \varphi \circ \psi$. In particular, $L = {\varphi'}^{-1}(\varphi'(L)) = \psi^{-1}(\varphi^{-1}(\varphi'(L)))$. In view of Theorem 2.17, it suffices to show that every language recognized by φ lies in $\mathcal{L}ind(\mathcal{K})$. This in turn reduces to showing that $\varphi^{-1}(F, g) \in \mathcal{L}ind(\mathcal{K})$ for each $(F, g) \in U$.

We use the information obtained in Fact 3.2 on the computation of $\varphi(t)$. As in that statement, we let $\alpha: AM \to S$ and $\beta: BM \to T$ be the natural morphisms, we let $\pi: \Sigma M \to BM$ be the morphism induced by the second coordinate projection from Σ to B, and we let $\tau = \beta \circ \pi: \Sigma M \to T$. For each $\sigma \in \Sigma$, we let $\varphi(\sigma) = (F_\sigma, b_\sigma)$.

Let $(F, g) \in U_n$. For each $t \in \Sigma M_n$, we have $\varphi(t) = (Q_t, \tau(t))$, where $Q_t \in S_n^{I_{n,k}}$ is described in Fact 3.2. We note that $\tau^{-1}(g)$ is recognized by (T, B), and hence is in $\mathcal{L}ind(\mathcal{K})$. We denote by χ_g a rank n $\mathbf{Lind}(\mathcal{K})$-sentence defining $\tau^{-1}(g)$.

Recall (from Fact 3.2) that $Q_t(D) = \alpha(\bar{t}_D)$ for all $D \in I_{k,n}$. For each $s \in S$, the tree language $K_s = \alpha^{-1}(s)$ is recognized by (S, A), and hence $K_s \in \mathcal{L}ind(\mathcal{K})$. It now suffices to show that, for each $s \in S_n$ and $D \in I_{k,n}$, there exists a rank n $\mathbf{Lind}(\mathcal{K})$-sentence $\psi_{s,D}$ defining the language

$$\{t \in \Sigma M_n \mid \bar{t}_D \in K_s\}.$$

Indeed, since $I_{k,n}$ is finite, it will follow that $\varphi^{-1}(F, g)$ is defined by the conjunction of χ_g and the $\psi_{s,D}$ ($D \in I_{k,n}$ and $F(D) = s$). We construct the sentence $\psi_{s,D}$ in the form $\psi_{s,D} = Q_{K_s} z \cdot \langle \psi_a \rangle_{a \in A}$ (where ψ_a is a rank n $\mathbf{Lind}(\mathcal{K})$-formula on Σ depending on a and D). (The formula $\psi_{s,D}$ is actually a $\mathbf{Lind}(\mathcal{L}ind(\mathcal{K}))$-formula but this is sufficient for our purpose in view of Theorem 2.15 (3).)

Let $a \in A_m$ and $D = (u, k_1, v, k_2) \in I_{k,n}$. For each $0 \leq i < j \leq n+1$, let $\bar{v}_1 = \oplus_{q=1}^{i} v_q$, $\bar{v}_2 = \oplus_{q=i+1}^{j-1} v_q$ and $\bar{v}_3 = \oplus_{q=j}^{n} v_q$; and let p_1, p_2, p_3 be the ranks of $\bar{v}_1$, $\bar{v}_2$ and $\bar{v}_3$ respectively. For each such i, j, for $m \geq 0$, $\sigma \in \Sigma_m$, $c \in T_{k_1+p_1+1+p_3+k_2}$, $c_1 \oplus \cdots \oplus c_m \in T_{m,p_2}$, we let

$$\psi' = P_\sigma(z) \wedge \mathsf{left}_i(z) \wedge \mathsf{right}_j(z) \wedge \chi_c[\not\geq z] \wedge \bigwedge_{\ell \in [m]} \chi_{c_\ell}[\geq x\ell]$$

and we let $\psi_a(z)$ be the (finite) disjunction of the $\psi'(z)$ when

$$F_\sigma(u \cdot (\mathbf{k}_1 \oplus c \cdot (\bar{v}_1 \oplus \mathbf{1} \oplus \bar{v}_3) \oplus \mathbf{k}_2), k_1 + p_1, (c_1 \oplus \cdots \oplus c_m) \cdot \bar{v}_2, p_3 + k_2) = a.$$

It is elementary to verify that $(t, z \mapsto x) \models \psi_a$ if and only if node x is labeled a in $\bar{t}_D$. Thus t satisfies $\psi_{s,D}$ if and only if $\bar{t}_D \in K_s$, if and only if $Q(t) = \alpha(\bar{t}_D) \in \alpha(K_s) = \{s\}$, which concludes the proof. □

Proposition 3.9 *Let Y be a set of first order variables and let $y \notin Y$. Let $\langle\varphi_\delta\rangle_{\delta\in\Delta}$ be a family of rank k* **Lind***-formulas over Σ, with free variables in $Y \cup \{y\}$, deterministic with respect to y, let $K \subseteq \Delta M_k$ be a tree language and let $\varphi = Q_K y \cdot \langle\varphi_\delta\rangle_{\delta\in\Delta}$.*

Let (S, A) be a pg-pair recognizing K, and let (T, B) be a pg-pair recognizing simultaneously the languages $L_{\varphi_\delta} \subseteq \Sigma_{Y\cup\{y\}} M_k$ ($\delta \in \Delta$). Then the language L_φ (a subset of $\Sigma_Y M_k$) is recognized by $(S, A) \,\square_k\, (T, B)$.

Proof. Let $\kappa\colon (\Delta M, \Delta) \to (S, A)$ be an onto morphism recognizing K, and let $\tau\colon (\Sigma_{Y\cup\{y\}} M, \Sigma_{Y\cup\{y\}}) \to (T, B)$ be an onto morphism recognizing each of the L_{φ_δ} ($\delta \in \Delta$).

We observe the following: if $t \in \Sigma_{Y\cup\{y\}} M_k$ is a $(Y \cup \{y\})$-structure and y occurs in the label of a rank n node, then there exists a unique $\delta \in \Delta_n$ such that $\tau(t) \in \tau(L_{\varphi_\delta})$. Indeed, the determinism of $\langle\varphi_\delta\rangle_\delta$ with respect to y shows that t lies in exactly one of the L_{φ_δ} ($\delta \in \Delta_n$): by hypothesis, $\tau^{-1}\big(\tau(L_{\varphi_\varepsilon})\big) = L_{\varphi_\varepsilon}$ for each ε, so $\tau(t) \in \tau(L_{\varphi_\varepsilon})$ and $\varepsilon \in \Delta_n$ implies $\varepsilon = \delta$.

We consider the block product of *pg*-pairs $(S, A) \,\square_k\, (T, B)$. For each $\sigma \in \Sigma_n$ and $Z \subseteq Y$ (so that $(\sigma, Z) \in (\Sigma_Y)_n$), we let $\gamma(\sigma, Z) = (F_{\sigma,Z}, \tau(\sigma, Z))$, where $F_{\sigma,Z}$ is defined as follows. Let $(u, k_1, v, k_2) \in I_{k,n}$. If $u\cdot(\mathbf{k}_1 \oplus \tau(\sigma, Z\cup\{y\})\cdot v \oplus \mathbf{k}_2)$ is the τ-image of some $(Y \cup \{y\})$-structure, then we let $F_{\sigma,Z}(u, k_1, v, k_2) = \kappa(\delta)$ where $\delta \in \Delta_n$ is uniquely determined by the property $\tau(u\cdot(\mathbf{k}_1 \oplus (\sigma, Z\cup\{y\})\cdot v \oplus \mathbf{k}_2)) \in \tau(L_{\varphi_\delta})$. Otherwise, we choose $F_{\sigma,Z}(u, k_1, v, k_2)$ arbitrarily in A_n. Note that $\gamma(\sigma, Z)$ lies in the generator set of the *pg*-pair $(S, A) \,\square_k\, (T, B)$.

Let now $t \in \Sigma_Y M_k$ and let $D = (\mathbf{1}, 0, \mathbf{k}, 0) \in I_{k,k}$. Then $\gamma(t) = (Q, \tau(t))$, and $Q(D) = \alpha(\bar{t}_D)$, where $\alpha\colon AM \to S$ is the natural morphism and $\bar{t}_D$ is described in Fact 3.2. In particular, let $x \in \mathsf{NV}(t)$ be a rank n node, labeled by (σ, Z) in t and let t factor as $t = f \cdot (\mathbf{r}_1 \oplus (\sigma, Z) \cdot h \oplus \mathbf{r}_3)$ where f, r_1 and r_3 are such that $(\sigma, Z)\cdot h$ is the subtree of t rooted at node x. The label of x in $\bar{t}_D$ is equal to $F_{\sigma,Z}(\tau(f), r_1, \tau(h), r_2)$, for the computation of which we need to consider the tree $\tau(f \cdot (\mathbf{r}_1 \oplus (\sigma, Z \cup \{y\}) \cdot h \oplus \mathbf{r}_3))$, that is, $\tau(t')$, where t' is equal to t with the label of x changed to $(\sigma, Z \cup \{y\})$. Note that t' is a $(Y \cup \{y\})$-structure, so x is labeled by $\kappa(\delta)$ $(\delta \in \Delta_n)$ in $\bar{t}_D$ if and only if $\tau(t') \in L_{\varphi_\delta}$.

Going back to the definition of the interpretation of Lindström quantifiers, this shows that $t \in L_\varphi$ if and only if $\bar{t}_D \in K$. As a result, $L_\varphi = \gamma^{-1}(P)$ where P consists of the pairs (F, f) such that $F(\mathbf{1}, 0, \mathbf{k}, 0) \in \alpha(K)$, and hence L_φ is recognized by $(S, A) \Box_k (T, B)$. $\Box$

We are now ready to complete the proof of Theorem 3.7.

Proof of Theorem 3.7. Let $\mathbf{K}$ be the class of syntactic *pg*-pairs of the elements of $\mathcal{K}$, let $\mathbf{V}$ be the pseudovariety of *pg*-pairs generated by $\mathbf{K}$, and let $\widehat{\mathbf{V}}$ be the least closed pseudovariety containing $\mathbf{V}$. We first show that if L is a tree language with syntactic *pg*-pair $(S, A) \in \widehat{\mathbf{V}}$, then $L \in \mathcal{L}ind(\mathcal{K})$. In view of Proposition 1.7, (S, A) can be obtained from elements of $\mathbf{K}$ by a succession of operations consisting of taking either a sub-*pg*-pair, a quotient, a direct product or a block product. We let $\sharp(S, A)$ be the least number of such operations, and we proceed by induction on $\sharp(S, A)$.

If $\sharp(S, A) = 0$, then $(S, A) \in \mathbf{K}$, that is, (S, A) is the range of the syntactic morphism $\varphi\colon (\Sigma M, \Sigma) \to (S, A)$ of a language $K \subseteq \Sigma M_k$ in $\mathcal{K}$. We want to show that every language recognized by (S, A) is in $\mathcal{L}ind(\mathcal{K})$. As in the first lines of the proof of Proposition 3.8, this reduces to showing that for each $s \in S$, we have $\varphi^{-1}(s) \in \mathcal{L}ind(\mathcal{K})$. Now we deduce from Remark 1.10 that

$$\varphi^{-1}(s) \;=\; \bigcap \big((u, k_1, k_2)^{-1} K\big) v^{-1} \setminus \bigcup \big((u, k_1, k_2)^{-1} K\big) v^{-1},$$

where the intersection runs over all n-ary contexts (u, k_1, v, k_2) over ΣM_k such that $\big((u, k_1, k_2)^{-1} K\big) v^{-1}$ meets $\varphi^{-1}(s)$, and the union over the n-ary contexts that do not. Moreover, by Remark 1.10 again, this union and this intersection are finite. It follows from Theorem 2.17 that $\varphi^{-1}(s)$, and hence any language recognized by (S, A) lies in $\mathcal{L}ind(\mathcal{K})$.

We now suppose that $\sharp(S, A) > 0$. If (S, A) is a sub-*pg*-pair or a quotient of a *pg*-pair $(T, B) \in \widehat{\mathbf{V}}$ with $\sharp(T, B) < \sharp(S, A)$, Proposition 1.3 establishes that L is also recognized by (T, B), so every such language is in $\mathcal{L}ind(\mathcal{K})$ by induction hypothesis. If (S, A) is the direct product of *pg*-pairs (T, B) and (T', B') with lesser $\sharp$-values, then by a standard argument, every language recognized by (S, A) is a finite union of intersections of the form $L \cap L'$, where L is recognized by (T, B) and L' by (T', B'). In particular, such a language is in $\mathcal{L}ind(\mathcal{K})$ by Theorem 2.17. If on the other hand, (S, A) divides a block product of *pg*-pairs with lesser $\sharp$-values, the inductive step follows directly from Proposition 3.8.

This concludes the proof that every tree language recognized by a *pg*-pair in $\widehat{\mathbf{V}}$ is in $\mathcal{L}ind(\mathcal{K})$. We now turn to the converse, namely showing that any tree language defined by a $\mathbf{Lind}(\mathcal{K})$-sentence has its syntactic *pg*-pair in $\widehat{\mathbf{V}}$.

This is implied by the following, more precise statement: if φ is a rank k $\mathbf{Lind}(\mathcal{K})$-formula with free variables in a finite set Y, then L_φ is recognized by a morphism $\alpha\colon (\Sigma_Y M, \Sigma_Y) \to (S, A)$ such that $\mathcal{I}\mathrm{m}_\emptyset(\alpha) \in \widehat{\mathbf{V}}$, where $\mathcal{I}\mathrm{m}_\emptyset(\alpha)$

denotes the sub-*pg*-pair of (S, A) generated by $\alpha(\Sigma)$ (recall that Σ is identified with the subset $\Sigma \times \{\emptyset\}$ of Σ_Y).

We prove this statement by structural induction on φ. If φ is an atomic formula and α is the syntactic morphism of L_φ, then $\mathcal{I}\mathrm{m}_\emptyset(\alpha)$ is trivial by Example 2.14, and hence lies in $\widehat{\mathbf{V}}$. If $\varphi = \varphi_1 \vee \varphi_2$, then by induction hypothesis, there exist morphisms $\alpha_i\colon (\Sigma_Y M, \Sigma_Y) \to (S_i, A_i)$ recognizing L_{φ_i} with $\mathcal{I}\mathrm{m}_\emptyset(\alpha_i) \in \widehat{\mathbf{V}}$, $i = 1, 2$. It is immediate that L_φ is recognizable by the target tupling $\alpha = (\alpha_1, \alpha_2)\colon (\Sigma_Y M, \Sigma_Y) \to ((S_1, A_1) \times (S_2, A_2))$, and that $\mathcal{I}\mathrm{m}_\emptyset(\alpha)$ is a sub-*pg*-pair of the direct product $\mathcal{I}\mathrm{m}_\emptyset(\alpha_1) \times \mathcal{I}\mathrm{m}_\emptyset(\alpha_2)$. Since $\widehat{\mathbf{V}}$ is a pseudovariety, it follows that $\mathcal{I}\mathrm{m}_\emptyset(\alpha)$ is in $\widehat{\mathbf{V}}$. The case where φ is of the form $\varphi = \neg\varphi_1$, is also easily treated: any morphism recognizing L_{φ_1} also recognizes its complement, namely L_φ.

Finally, suppose that φ is of the form $\varphi = Q_K y \cdot \langle \varphi_\delta \rangle_{\delta \in \Delta}$, where $K \subseteq \Delta M_k$ is recognized by some $(S, A) \in \mathbf{K}$. Without loss of generality, we may assume that $y \notin Y$. By induction, each L_{φ_δ} ($\delta \in \Delta$) is recognized by a morphism β_δ such that $\mathcal{I}\mathrm{m}_\emptyset(\beta_\delta) \in \widehat{\mathbf{V}}$. Taking the target tupling of the β_δ, we construct a morphism $\beta\colon (\Sigma_{Y \cup \{y\}} M, \Sigma_{Y \cup \{y\}}) \to (T', B')$ recognizing each L_{φ_δ} and such that $\mathcal{I}\mathrm{m}_\emptyset(\beta) \in \widehat{\mathbf{V}}$ (since $\mathcal{I}\mathrm{m}_\emptyset(\beta)$ is a sub-*pg*-pair of the direct product of the $\mathcal{I}\mathrm{m}_\emptyset(\beta_\delta)$). By Proposition 3.9 (and its proof), we find that L_φ is recognized by a morphism $\gamma\colon (\Sigma_Y M, \Sigma_Y) \to (S, A) \square_k (T', B')$, where the composition $\pi \circ \gamma\colon (\Sigma_Y M, \Sigma_Y) \to (T', B')$ agrees with β (here π is the second component projection). In particular, $\mathcal{I}\mathrm{m}_\emptyset(\pi \circ \gamma)$ is contained in $\mathcal{I}\mathrm{m}_\emptyset(\beta)$. Thus $\mathcal{I}\mathrm{m}_\emptyset(\gamma)$ is a sub-*pg*-pair of $(S, A) \square_k^{(T', B')} \mathcal{I}\mathrm{m}_\emptyset(\beta)$, and hence $\mathcal{I}\mathrm{m}_\emptyset(\gamma) \in \widehat{\mathbf{V}}$ by Proposition 3.5. □

3.4 Applications

The first result follows directly from Theorem 3.7 and Proposition 3.6.

Theorem 3.10 *Let $\mathcal{K}$ be a class of recognizable tree languages such that each quotient of a language of $\mathcal{K}$ is in $\mathcal{L}ind(\mathcal{K})$ and such that* $\mathbf{Lind}(\mathcal{K})$ *admits relativization. Let $\mathbf{V}$ be the least pseudovariety of pg-pairs containing the syntactic pg-pairs of elements of $\mathcal{K}$ and let $\widehat{\mathbf{V}}$ be the least closed pseudovariety containing $\mathbf{V}$. The following holds.*

- *$\mathcal{L}ind(\mathcal{K})$ is a literal variety of recognizable tree languages, associated with the pseudovariety $\widehat{\mathbf{V}}$ in the Eilenberg correspondence (Theorem 1.11).*
- *If $\mathcal{K}$ is the class of languages recognized by a class $\mathbf{L}$ of preclones, then $\mathcal{L}ind(\mathcal{K})$ is a variety of recognizable tree languages. Moreover, if $\mathbf{W}$ is the pseudovariety of preclones generated by $\mathbf{L}$ and $\widehat{\mathbf{W}}$ is the least closed pseudovariety of preclones containing $\mathbf{W}$, then $\widehat{\mathbf{W}}$ is the pseudovariety of preclones associated with $\mathcal{L}ind(\mathcal{K})$.*

Proof. Theorem 3.7 shows that $\mathcal{L}ind(\mathcal{K})$ consists of recognizable languages and Theorem 2.17 then shows that it is a literal variety. Let $\mathbf{X}$ be the pseudovariety of *pg*-pairs associated with $\mathcal{L}ind(\mathcal{K})$ by Theorem 1.11. Then $\mathbf{X}$ and $\widehat{\mathbf{V}}$ contain the same syntactic *pg*-pairs by Theorem 3.7, and Proposition 1.8 shows that this implies $\mathbf{X} = \widehat{\mathbf{V}}$. This concludes the proof of the first statement.

We now suppose that $\mathcal{K}$ is the class of tree languages recognized by the preclones in a class $\mathbf{L}$. By definition, $\mathbf{V} = \langle \mathsf{pgp}(\mathbf{L}) \rangle$ and Proposition 1.9 shows that $\mathbf{V}$ is full and

$$\mathbf{V} = \mathsf{pgp}(\langle \mathsf{precl}(\mathbf{V}) \rangle) = \mathsf{pgp}(\langle \mathbf{L} \rangle) = \mathsf{pgp}(\mathbf{W}).$$

Proposition 3.6 then shows that $\widehat{\mathbf{V}}$ is full and $\widehat{\mathbf{V}} = \mathsf{pgp}(\widehat{\mathbf{W}})$. Thus $\mathcal{L}ind(\mathcal{K})$ is a variety of tree languages (Corollary 1.12) and the corresponding pseudovariety of preclones is $\mathsf{precl}(\widehat{\mathbf{V}}) = \widehat{\mathbf{W}}$. □

The following statement is an important consequence of Theorem 3.10, which motivated this work.

Corollary 3.11 $\mathcal{L}ind(\mathcal{K}_\exists)$ *(that is, the class of* FO*-definable tree languages) is a variety of tree languages and the corresponding variety of preclones is the least pseudovariety containing* $T_\exists$ *and closed under block product.*

Proof. Note that, by Example 2.13, $\mathcal{L}ind(\mathcal{K}_\exists)$ is the class of FO-definable tree languages. Recall that $\mathcal{K}_\exists$ consists of the language $K_k(\exists) \subseteq \Delta M_k$, where Δ is a ranked Boolean alphabet.

Now let $\mathcal{V}_\exists$ be the variety of tree languages corresponding to the pseudovariety $\langle T_\exists \rangle$ generated by $T_\exists$. According to Example 1.13, a language $L \subseteq \Sigma M_k$ is in $\mathcal{V}_\exists$ if and only if it is a Boolean combination of languages of the form $\Sigma' M_k$, $\Sigma' \subseteq \Sigma$. Now the complement of $\Sigma' M_k$ in ΣM_k is the language of trees that contain at least a letter outside Σ': therefore this complement is the inverse image of $K_k(\exists)$ in the literal morphism from ΣM to ΔM that maps Σ' to $\{0_n \mid n \geq 0\}$ and $\Sigma \setminus \Sigma'$ to $\{1_n \mid n \geq 0\}$. In view of the closure properties of $\mathcal{L}ind(\mathcal{K}_\exists)$ (Theorem 2.17), it follows that $\mathcal{K}_\exists \subseteq \mathcal{V}_\exists \subseteq \mathcal{L}ind(\mathcal{K}_\exists)$ and hence $\mathcal{L}ind(\mathcal{V}_\exists) = \mathcal{L}ind(\mathcal{K}_\exists)$ by Theorem 2.15.

By definition, $\mathcal{V}_\exists$ is a variety and as such, it is closed under taking quotients. The corresponding logic $\mathbf{Lind}(\mathcal{V}_\exists)$ is equivalent to $\mathbf{Lind}(\mathcal{K}_\exists)$ (since $\mathcal{L}ind(\mathcal{V}_\exists) = \mathcal{L}ind(\mathcal{K}_\exists)$) and hence it admits relativization by Corollary 2.19. Thus we can apply Theorem 3.10 to conclude the proof. □

A similar reasoning, using both $T_\exists$ and the preclones T_p (see Section 1.2 and Examples 1.4, 2.10, 2.13 and Corollary 2.19), yields the following result.

Corollary 3.12 *The class of* (FO+ MOD)*-definable tree languages is a variety of tree languages and the corresponding variety of preclones is the least pseudovariety containing* $T_\exists$ *and the* T_p *(*$p \geq 2$*) and closed under block product.*

Conclusion

We reduced the characterization of the expressive power of certain naturally defined logics on trees, a chief example of which is given by first-order sentences, to an algebraic problem. This algebraic problem is set in a new algebraic framework, that of preclones, which the authors introduced in [18] precisely for the purpose of discussing tree languages. It is worth stating again that the notion of algebraic recognizability resulting from this new framework coincides with the usual one: we simply gave ourselves a richer algebraic set-up to classify recognizable tree languages.

Our result does not yield (yet?) 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. Our key algebraic tool is the block product: this product was introduced by Rhodes and Tilson [28] for monoids, to investigate the lattice of pseudovarieties of monoids and its application to the theory of formal languages (of finite words), and we adapted its definition for preclones. The use of wreath products instead of block products (the wreath product can be seen as a one-sided restriction of the block product) can yield algebraic characterizations for other natural classes of recognizable tree languages, see [14].

Our approach also raises a number of questions. At a technical level first: it was shown in [28] that for monoids, the block product can be expressed in terms of a double semidirect product, a two-sided generalization of the semidirect product. It might be convenient to have such a notion for preclones as well, and to derive analogues of the wreath product principle and the block product principle (general descriptions of the languages recognized by a wreath product or a block product). This might yield, as in the finite word case, the characterization of the recognizing power of the block product of two varieties, the characterization of logical hierarchies within FO, etc.

At a more general level, we observe that in the word language case, the decidability of first-order definability does not stem from the analogue of our main result, namely the fact that a language is FO-definable if and only if its syntactic monoid is in the least pseudovariety containing $\{0, 1\}$ and closed under block product. It follows rather from the characterization of that class of monoids as the aperiodic monoids, see the theorems of McNaughton and Papert on the equivalence of FO-definability and star-freeness, and of Schützenberger on the equivalence between star-freeness and aperiodicity. This characterization makes use in an essential way of the notion of star-freeness and of the structure theory of finite monoids. The question is therefore whether we can find a useful analogue of star-freeness for tree languages. There were attempts in this direction ([19, 27, 26]) that established that the more natural notions of star-freeness for trees do not coincide with FO-definability. Are we missing on an important concept? Taking the question from a different angle, can we directly develop the relevant fragment of a structure theory of finitary preclones, to prove decidability of FO-definability?

Another, more general remark is the following. We are convinced that the algebraic concept of preclones is well suited for the study and the classification of recognizable languages of finite ranked trees. However, we are conscious that beyond its qualities (the first of which is to allow results such as those proved in this paper), our algebraic framework is cumbersome, and perhaps intimidating. We argued in [18] that several known results on the characterization of particular classes of tree languages can be expressed in a natural way in the language of preclones, — but there might be an equivalent, yet lighter algebraic set-up.

This remark is related with another question. Other algebraic frameworks have been investigated in the literature since our results were announced in 2003 [17], in the (considerable) interval it took for this paper to be written and refereed. One of the more promising and elegant is the notion of *forest algebras*, introduced by Bojańczyk and Walukiewicz [8], initially to discuss languages of unranked, unordered trees. These authors also achieved the algebraic characterization of certain logically defined tree languages. More recent papers record interesting results using forest algebras, also to discuss ranked or ordered trees (e.g. Bojańczyk, Ségoufin, Straubing, Walukiewicz [4, 5, 6]), and it is tempting to wonder whether this algebraic approach and ours could be unified, since forest algebras may be seen as an unsorted version of preclones.

Possibly as a longer term project, one should consider the following. From the point of view of applications (in the field of verification, the investigation of distributed computation models, of game theory, etc), being able to handle languages of infinite (ranked ordered) trees is important. Discussing languages of infinite words as well as languages of finite words, was a topic of interest from the very beginnings of automata theory (Büchi), automata models were proposed quite early on, but the development of an algebraic model to handle them (namely the notion of ω-semigroups) was very slow in coming, and was matured only in the late 1980s (through work of Arnold, Perrin, Pin, Wilke, etc, see [21]). Can an analogous extension be developed for our preclones? One key technical tool in dealing with recognizable languages of infinite words is Ramsey's theorem, and the authors are unfortunately not aware of a relevant analogue of this theorem for trees.

Acknowledgements The authors wish to thank Szabolcs Iván for his careful reading of preliminary versions of this work, which helped eliminate many mistakes.

References

[1] J. Almeida. On pseudovarieties, varieties of languages, filters of congruences, pseudoidentities and related topics, *Algebra Universalis* **27** (1990) 333-350.

[2] J. Almeida, S. Margolis, B. Steinberg, M. Volkov. Representation theory of finite semigroups, semigroup radicals and formal language theory, `arXiv:math/0702400v1[math.GR]`

[3] M. Benedikt, L. Ségoufin. Regular tree languages definable in FO and in FO_{mod}, to appear.

[4] M. Bojańczyk. Two-way unary temporal logic over trees, in *LICS 2007*, IEEE, 121-130.

[5] M. Bojańczyk, L. Ségoufin. Tree Languages Defined in First-Order Logic with One Quantifier Alternation, in (L. Aceto, I. Damgrard, L.A. Goldberg, M.M. Halldórsson, A. Ingólfsdóttir, I. Walukiewicz eds.) *ICALP 2008, part II*, Lect. Notes in Computer Science **5126**, Springer, 2008, 233-245.

[6] M. Bojańczyk, L. Ségoufin, H. Straubing. Piecewise Testable Tree Languages, in *LICS 2008*, IEEE, 442-451.

[7] M. Bojańczyk, I. Walukiewicz. Characterizing EF and EX tree logics, *Theoret. Computer Science* **358** (2006) 255-272.

[8] M. Bojańczyk, I. Walukiewicz. Forest algebras, in (J. Flum, E. Graedel, T. Wilke eds.) *Logic and Automata*, Texts in Logic and Games, Amsterdam University Press, 2007.

[9] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi. *Tree Automata Techniques and Applications*, `http://www.grappa.univ-lille3.fr/tata`.

[10] H.-D. Ebbinghaus, J. Flum. *Finite Model Theory*, Springer, 1995.

[11] S. Eilenberg. *Automata, Languages, and Machines*, vol. A and B, Academic Press, 1974 and 1976.

[12] Z. Ésik. A variety theorem for trees and theories, *Publicationes Mathematicae* **54** (1999) 711-762.

[13] Z. Ésik. Characterizing CTL-like logics on finite trees, *Theoret. Computer Science* **356** (2006) 136-152.

[14] Z. Ésik, Sz. Iván. Some varieties of finite tree automata related to restricted temporal logics, *Fundamenta Informaticæ* **82** (2008) 79-103.

[15] Z. Ésik, Sz. Iván. Products of tree automata with an application to temporal logic, *Fundamenta Informaticæ* **82** (2008) 61-78.

[16] Z. Ésik, K. G. Larsen. Regular languages definable by Lindström quantifiers, *Theoretical Informatics and Applications* **37** (2003) 179-241.

[17] Z. Ésik, P. Weil. On logically defined recognizable tree languages, in *Proc. FST TCS 2003* (P. K. Pandya, J. Radhakrishnan eds.), Lect. Notes in Computer Science **2914**, Springer, 2003, 195-207.

[18] Z. Ésik, P. Weil. Algebraic characterization of regular tree languages, *Theoretical Computer Science* **340** (2005) 291-321.

[19] U. Heuter. First-order properties of trees, star-free expressions, and aperiodicity, *Theoretical Informatics and Applications* **25** (1991) 125-146.

[20] P. Lindström. First order predicate logic with generalized quantifiers. *Theoria* **32** (1966) 186-195.

[21] D. Perrin, J.-E. Pin. *Infinite words*, Pure and Applied Mathematics vol. **141**, Elsevier (2004).

[22] J.-E. Pin. *Variétés de langages formels*, Masson, Paris (1984). English translation: *Varieties of formal languages*, Plenum, New-York (1986).

[23] J.-E. Pin. Logic, Semigroups and Automata on Words, *Annals of Mathematics and Artificial Intelligence* **16** (1996) 343-384.

[24] J.-E. Pin, H. Straubing. Some results on C-varieties, *Theoret. Informatics Appl.* **39** (2005) 239-262.

[25] A. Podelski. A monoid approach to tree automata, in: *Tree automata and languages* (M. Nivat, A. Podelski, eds.), North Holland, 1992, 41-56.

[26] A. Potthoff. First order logic on finite trees, in: *TAPSOFT '95* (P.D. Mosses, M. Nielsen, M.I. Schwartzbach eds.), Lect. Notes in Computer Science **915**, Springer, 1995, 125-139.

[27] A. Potthoff, W. Thomas. Regular tree languages without unary symbols are star-free, in (Z. Ésik ed.) *FCT 93*, Lect. Notes in Computer Science **710**, Springer, 1993, 396-405.

[28] J. Rhodes, B. Tilson. The kernel of monoid morphisms, *J. Pure and Appl. Alg.* **62** (1989) 227-268.

[29] J. Rhodes, P. Weil. Decomposition techniques for finite semigroups using categories, I *Journal of Pure and Applied Algebra* **62** (1989) 269-284.

[30] J. Rhodes, P. Weil. Decomposition techniques for finite semigroups using categories, II, *Journal of Pure and Applied Algebra* **62** (1989) 285-312.

[31] S. Salehi. Varieties of tree languages definable by syntactic monoids, *Acta Cybernetica* **17** (2005), 21-41.

[32] S. Salehi, M. Steinby. Tree algebras and varieties of tree languages, *Theoret. Comput. Sci.* **377** (2007) 1-24.

[33] M. Steinby. A theory of tree language varieties, in: *Tree automata and languages* (M. Nivat, A. Podelski, eds.), North Holland, 1992, 57-81.

[34] M. Steinby. General varieties of tree languages, *Theoret. Comput. Sci.* **205** (1998) 1-43.

[35] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*, Birkhaüser, 1994.

[36] H. Straubing. On logical descriptions of regular languages, in *LATIN 2002*, Lect. Notes in Computer Science **2286**, Springer, 2002, 528-538.

[37] W. Thomas. Languages, automata and logic, in (G. Rozenberg, A. Salomaa, eds) *Handbook of Formal Languages*, volume III, pp. 389-455, Springer, 1997.

[38] P. Weil. Closure of varieties under products with counter, *Journal of Computing and System Science* **45** (1992) 316-339.

[39] P. Weil. Algebraic recognizability of languages, in: *MFCS 2004* (J. Fiala, V. Koubek, J. Kratochvíl eds.), Lect. Notes in Computer Science **3153**, Springer, 2004, 149-175.

[40] Th. Wilke. An algebraic characterization of frontier testable tree languages, *Theoret. Comput. Sci.* **154** (1996) 85-106.