

Algebraic recognizability of tree languages

Pascal Weil
(joint work with Z. Ésik)

LaBRI, CNRS and Université Bordeaux-1

Aachen, 16 November 2006

Recognizable languages

Preclones: an algebraic framework

Logically defined tree languages

Recognizable tree languages

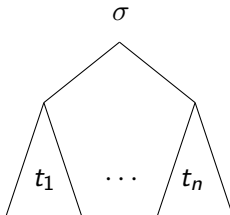
- ▶ Finite, ranked, ordered trees

Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children

Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children



Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children

If L is a set of Σ -trees, TFAE (Doner, Thatcher, Wright, 1960s):

Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children

If L is a set of Σ -trees, TFAE (Doner, Thatcher, Wright, 1960s):

- ▶ recognizability by a bottom-up tree automaton

Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children

If L is a set of Σ -trees, TFAE (Doner, Thatcher, Wright, 1960s):

- ▶ recognizability by a bottom-up tree automaton
- ▶ recognizability by a deterministic bottom-up tree automaton

Recognizable tree languages

- ▶ Finite, ranked, ordered trees
- ▶ $\Sigma = (\Sigma_n)_n$, a finite ranked alphabet. In a Σ -tree, nodes are labeled by letters in Σ , a node labeled by $\sigma \in \Sigma_n$ has n children

If L is a set of Σ -trees, TFAE (Doner, Thatcher, Wright, 1960s):

- ▶ recognizability by a bottom-up tree automaton
- ▶ recognizability by a deterministic bottom-up tree automaton
- ▶ MSO-definability (atoms: $P_\sigma(x)$ ($\sigma \in \Sigma$), $x < y$, $\text{Succ}_i(x, y)$)

The algebraic approach for word languages

- ▶ Recognizable word languages are also recognized by finite monoids

The algebraic approach for word languages

- ▶ Recognizable word languages are also recognized by finite monoids
- ▶ **Application:** L is FO-definable (resp. LTL-definable) iff L is recognized by an aperiodic monoid

The algebraic approach for word languages

- ▶ Recognizable word languages are also recognized by finite monoids
- ▶ **Application:** L is FO-definable (resp. LTL-definable) iff L is recognized by an aperiodic monoid
- ▶ **Application:** FO- and LTL-definability are decidable

The algebraic approach for word languages

- ▶ Recognizable word languages are also recognized by finite monoids
- ▶ **Application:** L is FO-definable (resp. LTL-definable) iff L is recognized by an aperiodic monoid
- ▶ **Application:** FO- and LTL-definability are decidable
- ▶ Eilenberg's theory of varieties: fine classification of recognizable languages, piecewise or local testability, FO₂- and FO + MOD-definability, levels in the FO-hierarchy, etc

The algebraic approach for word languages

- ▶ Recognizable word languages are also recognized by finite monoids
- ▶ **Application:** L is FO-definable (resp. LTL-definable) iff L is recognized by an aperiodic monoid
- ▶ **Application:** FO- and LTL-definability are decidable
- ▶ Eilenberg's theory of varieties: fine classification of recognizable languages, piecewise or local testability, FO_2 - and $\text{FO} + \text{MOD}$ -definability, levels in the FO-hierarchy, etc
- ▶ **In contrast**, the decidability of $\text{FO}[<]$ is open for trees (see [Benedikt and Ségoufin, 2005] for an algebraic characterization and decidability of $\text{FO}[\text{Succ}]$)

My objective today

- ▶ To present an algebraic framework for the description of recognizable tree languages,

My objective today

- ▶ To present an algebraic framework for the description of recognizable tree languages,
- ▶ to explain how FO-definable tree languages are characterized in this framework.

My objective today

- ▶ To present an algebraic framework for the description of recognizable tree languages,
- ▶ to explain how FO-definable tree languages are characterized in this framework.
- ▶ The problem of deciding FO-definability remains open. . .

Algebraic views of the minimal automaton (1/3)

- ▶ L a recognizable language of Σ -trees, \mathcal{A} its minimal automaton with state set Q

Algebraic views of the minimal automaton (1/3)

- ▶ L a recognizable language of Σ -trees, \mathcal{A} its minimal automaton with state set Q
- ▶ ΣM_0 , the set of Σ -trees is a Σ -algebra

$$\sigma \cdot (t_1, \dots, t_n) = \begin{array}{c} \sigma \\ \diagdown \quad \diagup \\ \triangle_{t_1} \quad \dots \quad \triangle_{t_n} \end{array}$$

Algebraic views of the minimal automaton (1/3)

- ▶ L a recognizable language of Σ -trees, \mathcal{A} its minimal automaton with state set Q
- ▶ ΣM_0 , the set of Σ -trees is a Σ -algebra

$$\sigma \cdot (t_1, \dots, t_n) =$$

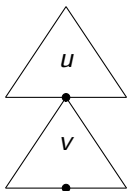
- ▶ Induces a notion of syntactic algebra: the syntactic Σ -algebra of L is in fact \mathcal{A} .

Algebraic views of the minimal automaton (2/3)

- ▶ ΣM_1 , the set of Σ -trees with a special, unlabeled leaf. They can be composed (a monoid), and composed with elements of ΣM_0 .

Algebraic views of the minimal automaton (2/3)

- ▶ ΣM_1 , the set of Σ -trees with a special, unlabeled leaf. They can be composed (a monoid), and composed with elements of ΣM_0 .

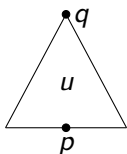


Algebraic views of the minimal automaton (2/3)

- ▶ ΣM_1 , the set of Σ -trees with a special, unlabeled leaf. They can be composed (a monoid), and composed with elements of ΣM_0 .

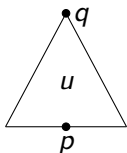
- ▶ Induces a notion of *syntactic monoid* of L : If $x, y \in \Sigma M_1$, say $x \sim_L y$ if $\forall u \in \Sigma M_1, v \in \Sigma M_0, u \cdot x \cdot v \in L$ if and only if $u \cdot y \cdot v \in L$.

Algebraic views of the minimal automaton (2/3)



- ▶ Induces a notion of *syntactic monoid* of L : If $x, y \in \Sigma M_1$, say $x \sim_L y$ if $\forall u \in \Sigma M_1, v \in \Sigma M_0, u \cdot x \cdot v \in L$ if and only if $u \cdot y \cdot v \in L$.
- ▶ Each $u \in \Sigma M_1$ induces a map $Q \rightarrow Q, p \mapsto q$.

Algebraic views of the minimal automaton (2/3)



- ▶ Induces a notion of *syntactic monoid* of L : If $x, y \in \Sigma M_1$, say $x \sim_L y$ if $\forall u \in \Sigma M_1, v \in \Sigma M_0, u \cdot x \cdot v \in L$ if and only if $u \cdot y \cdot v \in L$.
- ▶ Each $u \in \Sigma M_1$ induces a map $Q \rightarrow Q, p \mapsto q$.
- ▶ This is not just *in analogy* with the word case, it is a *generalization*: an ordinary alphabet is a ranked alphabet with all letters of rank 1.

Algebraic views of the minimal automaton (3/3)

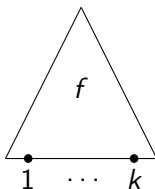
► *Let us continue!*

Algebraic views of the minimal automaton (3/3)

- ▶ *Let us continue!*
- ▶ ΣM_k , the set of Σ -trees with k special leaves, labeled $1, \dots, k$ from left to right.

Algebraic views of the minimal automaton (3/3)

- ▶ *Let us continue!*
- ▶ ΣM_k , the set of Σ -trees with k special leaves, labeled $1, \dots, k$ from left to right.

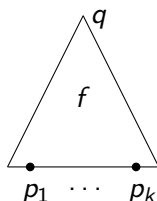


Algebraic views of the minimal automaton (3/3)

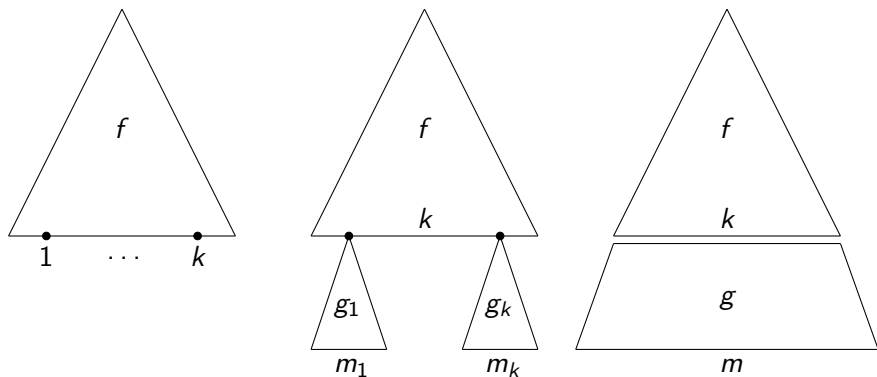
- ▶ Each such k -ary Σ -tree defines a map $Q^k \rightarrow Q$,
 $(p_1, \dots, p_k) \mapsto q$

Algebraic views of the minimal automaton (3/3)

- ▶ Each such k -ary Σ -tree defines a map $Q^k \rightarrow Q$,
 $(p_1, \dots, p_k) \mapsto q$



$\Sigma M = (\Sigma M_k)_{k \geq 0}$ is an algebra for composition.



So is $\mathbb{T}(Q) = (\mathbb{T}_k(Q))_k$, where $\mathbb{T}_k(Q)$ is the set of k -ary transformations of Q .

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)
- ▶ composition: $f \in S_k, g_1 \in S_{m_1}, \dots, g_n \in S_{m_n},$

$$f \cdot (g_1 \oplus \dots \oplus g_n) \in S_m$$

with $m = m_1 + \dots + m_n$

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)
- ▶ composition: $f \in S_k, g_1 \in S_{m_1}, \dots, g_n \in S_{m_n},$

$$f \cdot (g_1 \oplus \dots \oplus g_n) \in S_m$$

with $m = m_1 + \dots + m_n$

- ▶ unit element $\mathbf{1} \in S_1$

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)
- ▶ composition: $f \in S_k, g_1 \in S_{m_1}, \dots, g_n \in S_{m_n},$

$$f \cdot (g_1 \oplus \dots \oplus g_n) \in S_m$$

with $m = m_1 + \dots + m_n$

- ▶ unit element $\mathbf{1} \in S_1$
- ▶ associativity axioms

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)
- ▶ composition: $f \in S_k, g_1 \in S_{m_1}, \dots, g_n \in S_{m_n},$

$$f \cdot (g_1 \oplus \dots \oplus g_n) \in S_m$$

with $m = m_1 + \dots + m_n$

- ▶ unit element $\mathbf{1} \in S_1$
- ▶ associativity axioms
- ▶ **Note:** S_1 is a monoid

Preclones = abstraction of ΣM and $\mathbb{T}(Q)$

- ▶ $S = (S_k)_k$ (infinitely many *sorts* of elements)
- ▶ composition: $f \in S_k, g_1 \in S_{m_1}, \dots, g_n \in S_{m_n},$

$$f \cdot (g_1 \oplus \dots \oplus g_n) \in S_m$$

with $m = m_1 + \dots + m_n$

- ▶ unit element $\mathbf{1} \in S_1$
- ▶ associativity axioms
- ▶ **Note:** S_1 is a monoid
- ▶ S is *finitary* if each S_k is finite

Recognizability by a finitary preclone

- ▶ $\varphi: \Sigma M \rightarrow S$, morphism into a finitary preclone (φ respects sorts and products)

Recognizability by a finitary preclone

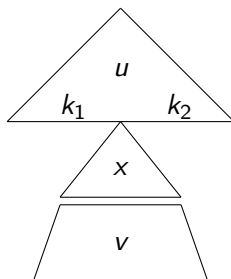
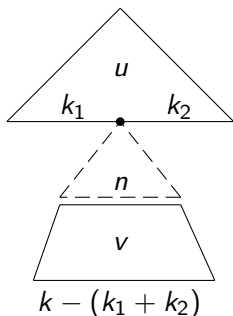
- ▶ $\varphi: \Sigma M \rightarrow S$, morphism into a finitary preclone (φ respects sorts and products)
- ▶ φ recognizes $L \in \Sigma M_k$ if $L = \varphi^{-1}\varphi(L)$

Recognizability by a finitary preclone

- ▶ $\varphi: \Sigma M \rightarrow S$, morphism into a finitary preclone (φ respects sorts and products)
- ▶ φ recognizes $L \in \Sigma M_k$ if $L = \varphi^{-1}\varphi(L)$
- ▶ recognizability of subsets of ΣM_0 (Σ -tree languages) is equivalent to the classical definition

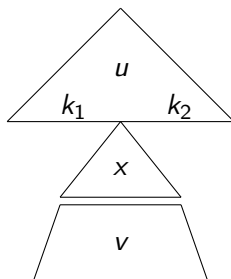
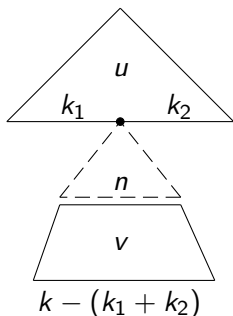
Syntactic preclone of $L \subseteq \Sigma M_k$ (1/2)

- an n -ary context in ΣM_k is a tuple (u, k_1, v, k_2) with $k_1, k_2 \geq 0$, $u \in \Sigma M_{k_1+1+k_2}$, $v = v_1 \oplus \dots \oplus v_n \in \Sigma_{n,\ell}$ and $k = k_1 + \ell + k_2$;



Syntactic preclone of $L \subseteq \Sigma M_k$ (1/2)

- ▶ an n -ary context in ΣM_k is a tuple (u, k_1, v, k_2) with $k_1, k_2 \geq 0$, $u \in \Sigma M_{k_1+1+k_2}$, $v = v_1 \oplus \cdots \oplus v_n \in \Sigma_{n,\ell}$ and $k = k_1 + \ell + k_2$;



- ▶ An L -context of $x \in \Sigma M_n$ if $u \cdot (k_1 \oplus x \cdot v \oplus k_2) \in L$.

Syntactic preclone of $L \subseteq \Sigma M_k$ (2/2)

- ▶ *Syntactic congruence* of L : if $x, y \in \Sigma_n$, say that $x \sim_L y$ if x and y have the same L -contexts

Syntactic preclone of $L \subseteq \Sigma M_k$ (2/2)

- ▶ *Syntactic congruence* of L : if $x, y \in \Sigma_n$, say that $x \sim_L y$ if x and y have the same L -contexts
- ▶ *Syntactic preclone*: $\text{Synt}(L) = \Sigma M / \sim_L$

Syntactic preclone of $L \subseteq \Sigma M_k$ (2/2)

- ▶ *Syntactic congruence* of L : if $x, y \in \Sigma_n$, say that $x \sim_L y$ if x and y have the same L -contexts
- ▶ *Syntactic preclone*: $\text{Synt}(L) = \Sigma M / \sim_L$
- ▶ In $\text{Synt}(L)$, sort 0 is the minimal automaton, sort 1 is the syntactic monoid, the other sorts carry further information

Varieties and Eilenberg's theorem

- ▶ One can define varieties of finitary preclones and varieties of recognizable tree languages, and state an Eilenberg-type theorem.

Varieties and Eilenberg's theorem

- ▶ One can define varieties of finitary preclones and varieties of recognizable tree languages, and state an Eilenberg-type theorem.
- ▶ Varieties of tree languages: closure under Boolean operations, inverse morphisms (between free preclones ΣM) and left and right quotients

Varieties and Eilenberg's theorem

- ▶ One can define varieties of finitary preclones and varieties of recognizable tree languages, and state an Eilenberg-type theorem.
- ▶ Varieties of tree languages: closure under Boolean operations, inverse morphisms (between free preclones ΣM) and left and right quotients
- ▶ Varieties of finitary preclones: closure under sub-preclone, morphic image, direct product + finitary inverse limits and finitary unions of ω -chains.

An example

- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.

An example

- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.
- ▶ T_{\exists} is generated by true_0 , $\text{or}_0 = \text{false}_0$ and or_2 .

An example

- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.
- ▶ T_{\exists} is generated by true_0 , $\text{or}_0 = \text{false}_0$ and or_2 .
- ▶ Let $\varphi: \Sigma M \rightarrow T_{\exists}$. Let $\Sigma^{(1)}$ be the set of letters that map to some true_n . Then $\varphi^{-1}(\text{true}_k)$ is the language of k -ary trees containing a label in $\Sigma^{(1)}$.

An example

- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.
- ▶ T_{\exists} is generated by true_0 , $\text{or}_0 = \text{false}_0$ and or_2 .
- ▶ Let $\varphi: \Sigma M \rightarrow T_{\exists}$. Let $\Sigma^{(1)}$ be the set of letters that map to some true_n . Then $\varphi^{-1}(\text{true}_k)$ is the language of k -ary trees containing a label in $\Sigma^{(1)}$.
- ▶ (This is a simple example of a Σ_1 -formula.)

An example

- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.
- ▶ T_{\exists} is generated by true_0 , $\text{or}_0 = \text{false}_0$ and or_2 .
- ▶ Let $\varphi: \Sigma M \rightarrow T_{\exists}$. Let $\Sigma^{(1)}$ be the set of letters that map to some true_n . Then $\varphi^{-1}(\text{true}_k)$ is the language of k -ary trees containing a label in $\Sigma^{(1)}$.
- ▶ (This is a simple example of a Σ_1 -formula.)
- ▶ The variety of tree languages associated with $\langle T_{\exists} \rangle$ consists of the Boolean closures of such languages.

An example

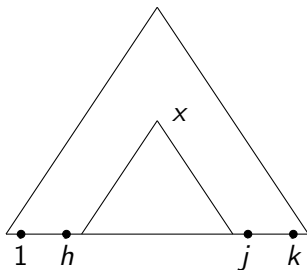
- ▶ Let $T_{\exists} \subseteq \mathbb{T}(\mathbb{B})$ (where $\mathbb{B} = \{\text{true}, \text{false}\}$) be given by $(T_{\exists})_n = \{\text{or}_n, \text{true}_n\}$.
- ▶ T_{\exists} is generated by true_0 , $\text{or}_0 = \text{false}_0$ and or_2 .
- ▶ Let $\varphi: \Sigma M \rightarrow T_{\exists}$. Let $\Sigma^{(1)}$ be the set of letters that map to some true_n . Then $\varphi^{-1}(\text{true}_k)$ is the language of k -ary trees containing a label in $\Sigma^{(1)}$.
- ▶ (This is a simple example of a Σ_1 -formula.)
- ▶ The variety of tree languages associated with $\langle T_{\exists} \rangle$ consists of the Boolean closures of such languages.
- ▶ For specialists: note that $(T_{\exists})_1 = \{\text{true}_1, \text{or}_1\}$ is the monoid $U_1 = \{0, 1\}$ and $\langle T_{\exists} \rangle$ is a generalization of \mathbf{J}_1 .

Logic on trees in ΣM_k

- ▶ Atomic formulas: $P_\sigma(x)$ ($\sigma \in \Sigma$), $x < y$, $\text{Succ}_i(x, y)$ ($i \leq \max_{\sigma \in \Sigma} \text{rank}(\sigma)$), and $\text{left}_h(x)$ and $\text{right}_j(x)$

Logic on trees in ΣM_k

- ▶ Atomic formulas: $P_\sigma(x)$ ($\sigma \in \Sigma$), $x < y$, $\text{Succ}_i(x, y)$ ($i \leq \max_{\sigma \in \Sigma} \text{rank}(\sigma)$), and $\text{left}_h(x)$ and $\text{right}_j(x)$



Logic on trees in ΣM_k

- ▶ Atomic formulas: $P_\sigma(x)$ ($\sigma \in \Sigma$), $x < y$, $\text{Succ}_i(x, y)$ ($i \leq \max_{\sigma \in \Sigma} \text{rank}(\sigma)$), and $\text{left}_h(x)$ and $\text{right}_j(x)$
- ▶ Formulas of rank k : atomic formulas with $h, j \leq k$, true, false, Boolean connectives, Lindstrom quantifiers

Logic on trees in ΣM_k

- ▶ Atomic formulas: $P_\sigma(x)$ ($\sigma \in \Sigma$), $x < y$, $\text{Succ}_i(x, y)$ ($i \leq \max_{\sigma \in \Sigma} \text{rank}(\sigma)$), and $\text{left}_h(x)$ and $\text{right}_j(x)$
- ▶ Formulas of rank k : atomic formulas with $h, j \leq k$, true, false, Boolean connectives, Lindstrom quantifiers
- ▶ *Lindstrom quantifiers* include existential or modular quantifiers, but also certain path quantifiers, modular quantifiers, etc

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,
- ▶ let φ_δ be formulas with free variables in $X \cup \{x\}$, such that for each $n \geq 0$ and every interpretation mapping x to a rank n vertex, exactly one of the formulas φ_δ ($\delta \in \Delta_n$) holds,

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,
- ▶ let φ_δ be formulas with free variables in $X \cup \{x\}$, such that for each $n \geq 0$ and every interpretation mapping x to a rank n vertex, exactly one of the formulas φ_δ ($\delta \in \Delta_n$) holds,
- ▶ and let $K \subseteq \Delta M_k$.

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,
- ▶ let φ_δ be formulas with free variables in $X \cup \{x\}$, such that for each $n \geq 0$ and every interpretation mapping x to a rank n vertex, exactly one of the formulas φ_δ ($\delta \in \Delta_n$) holds,
- ▶ and let $K \subseteq \Delta M_k$.
- ▶ Let $t \in \Sigma M_k$ and let $\lambda: X \rightarrow V(t)$.

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,
- ▶ let φ_δ be formulas with free variables in $X \cup \{x\}$, such that for each $n \geq 0$ and every interpretation mapping x to a rank n vertex, exactly one of the formulas φ_δ ($\delta \in \Delta_n$) holds,
- ▶ and let $K \subseteq \Delta M_k$.
- ▶ Let $t \in \Sigma M_k$ and let $\lambda: X \rightarrow V(t)$.
- ▶ Then $(t, \lambda) \models Q_{K^x} \cdot \langle \varphi_\delta \rangle_\delta$ if the tree \bar{t}_λ lies in K , where

Lindstrom quantifiers

- ▶ Let Δ be a finite ranked alphabet such that $\Delta_n \neq \emptyset$ whenever $\Sigma_n \neq \emptyset$,
- ▶ let φ_δ be formulas with free variables in $X \cup \{x\}$, such that for each $n \geq 0$ and every interpretation mapping x to a rank n vertex, exactly one of the formulas φ_δ ($\delta \in \Delta_n$) holds,
- ▶ and let $K \subseteq \Delta M_k$.
- ▶ Let $t \in \Sigma M_k$ and let $\lambda: X \rightarrow V(t)$.
- ▶ Then $(t, \lambda) \models Q_{K \times} \langle \varphi_\delta \rangle_\delta$ if the tree \bar{t}_λ lies in K , where
- ▶ \bar{t}_λ is a relabeling of t by letters from Δ : a rank n vertex v is labeled by **the** letter $\delta \in \Delta_n$ such that $(t, [\lambda; x \mapsto v]) \models \varphi_\delta$.

The logic $\mathbf{Lind}(\mathcal{K})$ and the variety $\mathcal{Lind}(\mathcal{K})$ of tree languages

- ▶ Allow only Lindstrom quantifiers Q_K with $K \in \mathcal{K}$: logic $\mathbf{Lind}(\mathcal{K})$

The logic $\mathbf{Lind}(\mathcal{K})$ and the variety $\mathcal{Lind}(\mathcal{K})$ of tree languages

- ▶ Allow only Lindstrom quantifiers Q_K with $K \in \mathcal{K}$: logic $\mathbf{Lind}(\mathcal{K})$
- ▶ $\mathcal{Lind}(\mathcal{K}) = \mathbf{Lind}(\mathcal{K})$ -definable languages

The logic $\mathbf{Lind}(\mathcal{K})$ and the variety $\mathcal{Lind}(\mathcal{K})$ of tree languages

- ▶ Allow only Lindstrom quantifiers Q_K with $K \in \mathcal{K}$: logic $\mathbf{Lind}(\mathcal{K})$
- ▶ $\mathcal{Lind}(\mathcal{K}) = \mathbf{Lind}(\mathcal{K})$ -definable languages
- ▶ if \mathcal{K} is the variety of languages associated with $\langle T_{\exists} \rangle$, then $\mathbf{Lind}(\mathcal{K})$ is FO.

A generalization of an operation on monoids

- ▶ Let S, T be preclones and $k \geq 0$. For each $n \geq 0$, Let $C_{k,n}$ be the set of n -ary contexts in T_k .

A generalization of an operation on monoids

- ▶ Let S, T be preclones and $k \geq 0$. For each $n \geq 0$, Let $C_{k,n}$ be the set of n -ary contexts in T_k .
- ▶ The $S \square_k T$ is the preclone such that

$$(S \square_k T)_n = S_n^{C_{k,n}} \times T_n$$

A generalization of an operation on monoids

- ▶ Let S, T be preclones and $k \geq 0$. For each $n \geq 0$, Let $C_{k,n}$ be the set of n -ary contexts in T_k .
- ▶ The $S \square_k T$ is the preclone such that

$$(S \square_k T)_n = S_n^{C_{k,n}} \times T_n$$

- ▶ with product as follows.

The composition operation in a block product (1/3)

- ▶ Let $(F, f) \in (S \square_k T)_n$ and for each $1 \leq i \leq n$, $(G_i, g_i) \in (S \square_k T)_{m_i}$. Let $m = \sum_i m_i$.

The composition operation in a block product (1/3)

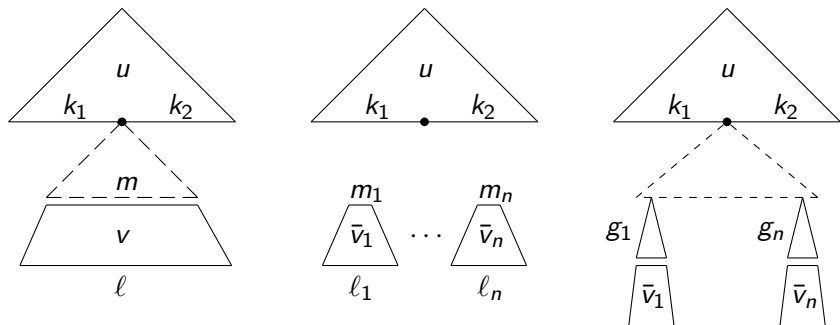
- ▶ Let $(F, f) \in (S \square_k T)_n$ and for each $1 \leq i \leq n$, $(G_i, g_i) \in (S \square_k T)_{m_i}$. Let $m = \sum_i m_i$.
- ▶ Then $(F, f) \cdot ((G_1, g_1) \oplus \cdots \oplus (G_n, g_n)) = (H, f \cdot (g_1 \oplus \cdots \oplus g_n))$ is the element of $(S \square_k T)_m$ such that

The composition operation in a block product (1/3)

- ▶ Let $(F, f) \in (S \square_k T)_n$ and for each $1 \leq i \leq n$, $(G_i, g_i) \in (S \square_k T)_{m_i}$. Let $m = \sum_i m_i$.
- ▶ Then $(F, f) \cdot ((G_1, g_1) \oplus \cdots \oplus (G_n, g_n)) = (H, f \cdot (g_1 \oplus \cdots \oplus g_n))$ is the element of $(S \square_k T)_m$ such that
- ▶ for $(u, k_1, v, k_2) \in C_{k,m}$,
$$H(u, k_1, v, k_2) = F(u, k_1, g \cdot v, k_2) \cdot \left(\bigoplus_{i=1}^n G_i(C_i) \right)$$

The composition operation in a block product (2/3)

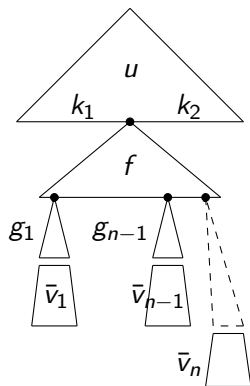
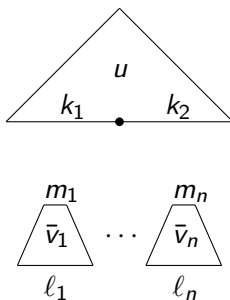
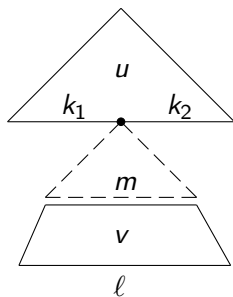
$$H(u, k_1, v, k_2) = F(u, k_1, g \cdot v, k_2) \cdot \left(\bigoplus_{i=1}^n G_i(C_i) \right)$$



$$(u, k_1, v, k_2) \in C_{k,m} \text{ and } (u, k_1, g \cdot v, k_2) \in C_{k,n}$$

The composition operation in a block product (3/3)

$$H(u, k_1, v, k_2) = F(u, k_1, g \cdot v, k_2) \cdot \left(\bigoplus_{i=1}^n G_i(C_i) \right)$$



$$(u, k_1, v, k_2) \in C_{k,m} \text{ and } C_n \in C_{k,m_n}$$

The main results (1/2)

- ▶ Let \mathcal{K} be a variety of tree languages with appropriate closure properties (technical: to allow relativization of the logic **Lind**(\mathcal{K}), etc).

The main results (1/2)

- ▶ Let \mathcal{K} be a variety of tree languages with appropriate closure properties (technical: to allow relativization of the logic **Lind**(\mathcal{K}), etc).
- ▶ If S and T recognize only languages in $\mathcal{L}ind(\mathcal{K})$, then so does $S \square_k T$.

The main results (1/2)

- ▶ Let \mathcal{K} be a variety of tree languages with appropriate closure properties (technical: to allow relativization of the logic **Lind**(\mathcal{K}), etc).
- ▶ If S and T recognize only languages in $\mathcal{L}ind(\mathcal{K})$, then so does $S \square_k T$.
- ▶ If S recognizes K and T recognizes the languages defined by the φ_δ , then $S \square_k T$ recognizes the language defined by $Q_{K^X} \cdot \langle \varphi_\delta \rangle_\delta$.

The main results (2/2)

- ▶ Let \mathbf{V} be the least variety of preclones containing $\text{Synt}(L)$ for $L \in \mathcal{K}$ and closed under block product. Then L is $\mathbf{Lind}(\mathcal{K})$ -definable if and only if $\text{Synt}(L) \in \mathbf{V}$.

The main results (2/2)

- ▶ Let \mathbf{V} be the least variety of preclones containing $\text{Synt}(L)$ for $L \in \mathcal{K}$ and closed under block product. Then L is $\mathbf{Lind}(\mathcal{K})$ -definable if and only if $\text{Synt}(L) \in \mathbf{V}$.
- ▶ FO-definable languages form a variety, which corresponds to the least closed variety containing T_{\exists} .

Conclusion on the results

- ▶ The first algebraic characterization of FO-definability.

Conclusion on the results

- ▶ The first algebraic characterization of FO-definability.
- ▶ FO-definable languages form a variety in a natural sense (not surprising).

Conclusion on the results

- ▶ The first algebraic characterization of FO-definability.
- ▶ FO-definable languages form a variety in a natural sense (not surprising).
- ▶ The result does not yield a decision procedure.

Conclusion on the results

- ▶ The first algebraic characterization of FO-definability.
- ▶ FO-definable languages form a variety in a natural sense (not surprising).
- ▶ The result does not yield a decision procedure.
- ▶ Neither does the analogous characterization for words, but the model of preclones gives a natural framework to express such a decidable characterization (as I expect there is one).

Conclusion on the model (1/2)

- ▶ Preclones have advantages: they generalize the word situation, as well as previous attempts such as the tree monoids,

Conclusion on the model (1/2)

- ▶ Preclones have advantages: they generalize the word situation, as well as previous attempts such as the tree monoids,
- ▶ they are algebraically robust (Eilenberg's theorem, universal algebra)

Conclusion on the model (1/2)

- ▶ Preclones have advantages: they generalize the word situation, as well as previous attempts such as the tree monoids,
- ▶ they are algebraically robust (Eilenberg's theorem, universal algebra)
- ▶ they are a natural by-product of the minimal automaton of a tree language,

Conclusion on the model (1/2)

- ▶ Preclones have advantages: they generalize the word situation, as well as previous attempts such as the tree monoids,
- ▶ they are algebraically robust (Eilenberg's theorem, universal algebra)
- ▶ they are a natural by-product of the minimal automaton of a tree language,
- ▶ they allow proving the characterization of FO-definability

Conclusion on the model (1/2)

- ▶ Preclones have advantages: they generalize the word situation, as well as previous attempts such as the tree monoids,
- ▶ they are algebraically robust (Eilenberg's theorem, universal algebra)
- ▶ they are a natural by-product of the minimal automaton of a tree language,
- ▶ they allow proving the characterization of FO-definability
- ▶ they allow a natural expression of other recent results (EF and EX-definability, Bojańczyk and Walukiewicz; FO[Succ]-definability, Benedikt and Ségoufin).

Conclusion on the model (2/2)

- ▶ Preclones have a big disadvantage: technically cumbersome.

Conclusion on the model (2/2)

- ▶ Preclones have a big disadvantage: technically cumbersome.
- ▶ Current hope: unification with the model of forest algebras, developed by Bojańczyk and Walukiewicz for unranked trees.

Thank you for your attention!