

FOREST ALGEBRAS

Igor Walukiewicz
joint work with Mikolaj Bojanczyk and Howard Straubing

CNRS, LaBRI Bordeaux

Bordeaux, November 2007

THE GOAL

UNDERSTAND LOGICS ON (FINITE) TREES.

SOME MORE PROMINENT TREE LOGICS

- CTL, CTL*, μ -calculus,
- FOL, MSOL, chain logic, anti-chain logic.

WE UNDERSTAND RELATIONS BETWEEN THESE LOGICS

μ -calculus = MSOL = anti-chain logic

\cup

\cup

\cup

CTL* = FOL \subsetneq chain logic

\cup

CTL

The case of
binary trees

THE GOAL

UNDERSTAND LOGICS ON (FINITE) TREES.

SOME MORE PROMINENT TREE LOGICS

- CTL, CTL*, μ -calculus,
- FOL, MSOL, chain logic, anti-chain logic.

WE UNDERSTAND RELATIONS BETWEEN THESE LOGICS

μ -calculus	\subseteq	MSOL	\supseteq	anti-chain logic
\cup		\cup		\nsubseteq
CTL*	\subseteq	FOL	\subseteq	chain logic
\cup				
CTL				

The case of unranked, unordered trees

THE GOAL

UNDERSTAND LOGICS ON (FINITE) TREES.

SOME MORE PROMINENT TREE LOGICS

- CTL, CTL*, μ -calculus,
- FOL, MSOL, chain logic, anti-chain logic.

WE UNDERSTAND RELATIONS BETWEEN THESE LOGICS

μ -calculus	\subseteq	MSOL	\supseteq	anti-chain logic
\cup		\cup		\nsubseteq
CTL*	\subseteq	FOL	\subseteq	chain logic
\cup				
CTL				

The case of unranked, unordered trees

PROBLEM

Given a property, decide in which logic it can be expressed.

AN EXAMPLE OF A SURPRISING EXPRESSIVE POWER

EVEN DEPTH PROPERTY (POTTHOFF)

- Consider the property of binary trees saying that all leaves in the forest are at even depth.
- This property can be expressed in $\text{FO}[\leq]$ (with the descendant order).

MIXED PARITY NODES

A **mixed parity node** has a path to a leaf of even length, and another one of odd length

THE CASE WHEN THERE ARE NO MIXED PARITY NODES

HOW TO CHECK IF THERE ARE MIXED PARITY NODES

AN EXAMPLE OF A LANGUAGE NOT IN FIRST-ORDER

POSITIVE BOOLEAN EXPRESSIONS

Let $A = \{\vee, \wedge, 0, 1\}$, and consider the set of trees that evaluate to 1 (and are well-formed).

THEOREM (HEUTER, POTTHOFF)

The above language is not definable in chain logic. (This language is aperiodic.)

THEOREM (POTTHOFF, THOMAS)

The above language is definable in antichain logic.

HOW IT IS DONE FOR WORDS

DEFINITION (RECOGNITION)

A language L is recognized by a monoid S if there are $\alpha : \Sigma^* \rightarrow S$ and $F \subseteq S$ such that $\alpha^{-1}(F) = L$.

DEFINITION (SYNTACTIC MONOID FOR L)

- Define $v_1 \sim_L v_2$ iff for all $u, w \in \Sigma^*$: $uv_1w \in L$ iff $uv_2w \in L$.
- This is an equivalence relation so we can take $\langle \Sigma^* / \sim_L, \cdot, \varepsilon \rangle$.

DEFINITION (APERIODICITY)

A monoid $\langle S, \cdot \rangle$ is **aperiodic** iff there is n such that $s^n = s^{n+1}$ for all $s \in S$.

THEOREM (SCHÜTZENBERGER, MCNAUGHTON & PAPERT)

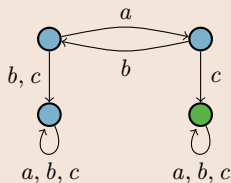
A language is FOL definable iff its syntactic monoid is aperiodic.

A NAIVE EXAMPLE COMPARING AUTOMATA AND ALGEBRA

A FOL DEFINABLE LANGUAGE

Let $L = (ab)^* c \Sigma^*$. It is first-order definable.

THE MINIMAL AUTOMATON FOR L



REMARKS

- The aperiodicity property is not that visible from the structure of the automaton.
- Every property of syntactic algebra is also a property of the minimal automaton. The converse is not true.
- This is good, as long as we know that properties of interest are properties of syntactic algebras.

HOW TO DO EQUALLY WELL FOR TREES?

THE BIG PLAN

- Define an algebraic notion of recognition.
- This should give a notion of syntactic algebra (invariant, for a given language).
- Decide the properties of the language looking at the properties of its syntactic algebra.
 - It is convenient to look at equational properties.
 - If we have an Eilenberg like theorem we can deduce that there is an equational characterization without knowing what it is.

RELATED WORK

- [Nivat & Podelski'87] Looking only at the structure contexts with application. Not enough to characterize FOL.
- [Wilke'96] Three sorted structure: labels, trees, contexts.
- [Salehi & Steinby' 07] Variety theorem for the Wilke's setting.
- [Esik'99]
- [Esik & Weil' 05] Preclones. Algebra of all term, i.e., contexts of arbitrary arity.
- [Thomas'84] Regular expressions for trees.
- [Heuter'89] Regular expressions for FOL.
- [Potthoff'] Star free=regular. Aperiodic is more than star free.
- [Bojańczyk & W.'04] Characterization of EF.
- [Benedikt & Segufin'05-'07] Characterization of $\text{FO}[succ]$.
- [Bojańczyk '07] Characterization of $\text{EF} + F^{-1}$.

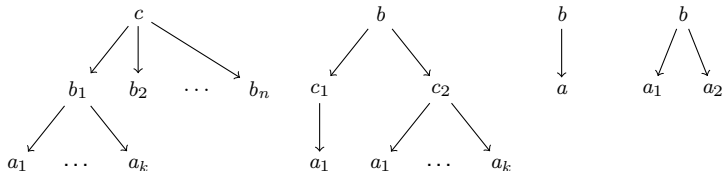
- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

PLAN

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

DEFINITION (TREES, FORESTS)

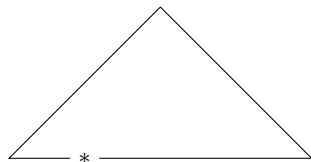
- A **A-tree** is a partial mapping $t : \mathbb{N}^* \rightarrow A$ with finite and prefix closed domain.
- **Forest** is a finite sequence of trees.



CONTEXT

DEFINITION (CONTEXTS)

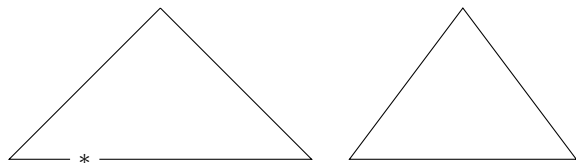
A A -context is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.



CONTEXT

DEFINITION (CONTEXTS)

A A -context is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.



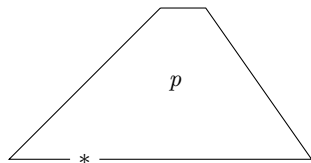
CONTEXT

DEFINITION (CONTEXTS)

A A -context is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.

We have two operations:

context substitution $p(t)$, and **context composition** $p \cdot q$.



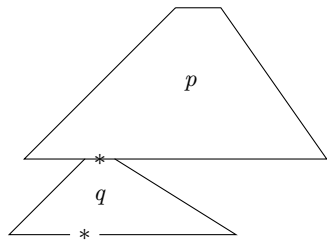
CONTEXT

DEFINITION (CONTEXTS)

A A -context is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.

We have two operations:

context substitution $p(t)$, and **context composition** $p \cdot q$.



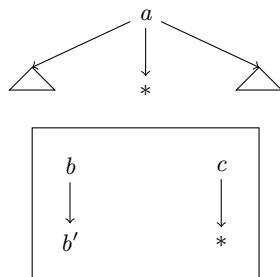
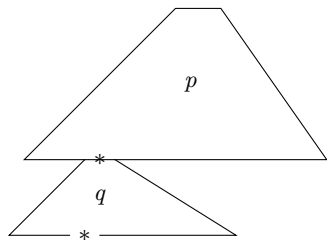
CONTEXT

DEFINITION (CONTEXTS)

A **A -context** is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.

We have two operations:

context substitution $p(t)$, and **context composition** $p \cdot q$.



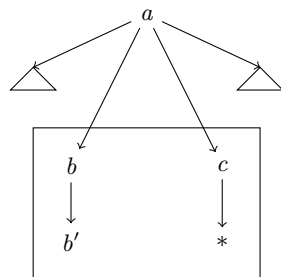
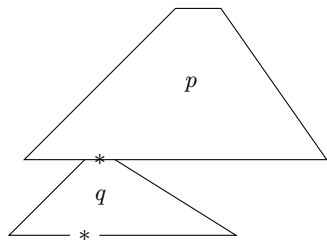
CONTEXT

DEFINITION (CONTEXTS)

A **A -context** is a $(A \cup \{*\})$ -forest, with $*$ occurring in exactly one leaf; called a **hole**.

We have two operations:

context substitution $p(t)$, and **context composition** $p \cdot q$.



FOREST AND CONTEXT MONOID

FOREST MONOID

Forest monoid consists of the set of A -forests, with concatenation operation (it is not commutative), and the empty tree.

CONTEXT MONOID

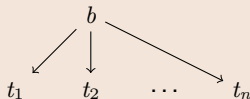
Context monoid consists of the set of A -contexts with context composition.

NOTATION

Forest concatenation will be denoted by $+$ and context composition by \cdot :

$$s + t, \quad p \cdot q$$

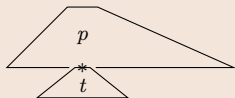
One letter trees and contexts are just denoted by letters : $b(t_1 + t_2 + \dots + t_n)$



ACTIONS OF FORESTS AND CONTEXTS

ACTION OF CONTEXTS ON FORESTS

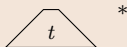
If p is a context and t is a forest then $p(t)$ is the tree obtained by the substitution of p in the hole of t .



$$\text{we have } (p \cdot q)(t) = p(q(t))$$

FROM FORESTS TO CONTEXTS.

If t is a forest then we have the context $in_l(t)$.



Similarly we have $in_r(t)$.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

ALGEBRAIC STRUCTURE

FOREST ALGEBRA $\langle H, V, act, in_l, in_r \rangle$

- Two monoids: H and V . We denote their operations by $+$ and \cdot , respectively.
- An action $act : H \times V \rightarrow H$. We write vh for $act(h, v)$.
- Two operations $in_l, in_r : H \rightarrow V$.
- Axioms:

ACTION $(v \cdot w)h = v(wh)$;

INSERTION $in_l(g)h = g + h$ and $in_r(g)h = h + g$;

FAITHFULNESS for every two distinct $v, w \in V$ there is $h \in H$ with $vh \neq wh$;

SOME CONVENTIONS

- H will be called **horizontal monoid**, and V **vertical monoid**.
- The neutral element of the horizontal monoid is **0** and the operation is $+$.
- The neutral element of the vertical monoid is **1** and the operation is \cdot .
- Action is written on the left: so $act(h, v)$ is written vh .

ALGEBRAIC STRUCTURE

FOREST ALGEBRA $\langle H, V, act, in_l, in_r \rangle$

- Two monoids: H and V . We denote their operations by $+$ and \cdot , respectively.
- An action $act : H \times V \rightarrow H$. We write vh for $act(h, v)$.
- Two operations $in_l, in_r : H \rightarrow V$.
- Axioms:

ACTION $(v \cdot w)h = v(wh)$;

INSERTION $in_l(g)h = g + h$ and $in_r(g)h = h + g$;

FAITHFULNESS for every two distinct $v, w \in V$ there is $h \in H$ with $vh \neq wh$;

EXAMPLE

- H any monoid, $V = H \rightarrow H$ with composition as multiplication.
- The action is application.
- The insertions are determined by the insertion axiom.

FOREST ALGEBRA $\langle H, V, act, in_l, in_r \rangle$

- Two monoids: H and V . We denote their operations by $+$ and \cdot , respectively.
- An action $act : H \times V \rightarrow H$. We write vh for $act(h, v)$.
- Two operations $in_l, in_r : H \rightarrow V$.
- Axioms:

ACTION $(v \cdot w)h = v(wh)$;

INSERTION $in_l(g)h = g + h$ and $in_r(g)h = h + g$;

FAITHFULNESS for every two distinct $v, w \in V$ there is $h \in H$ with $vh \neq wh$;

REMARKS

- Every element from H is of the form $v0$ for some $v \in V$: take $v = in_l h$.

$$v0 = (in_l h)0 = h + 0 = h .$$

- Mappings $in_l, in_r : H \rightarrow V$ are, injective, monoid homomorphisms:

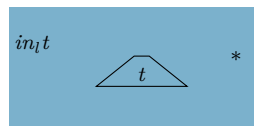
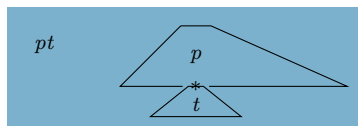
$$in_l(h_1 + h_2) = in_l(h_1)in_l(h_2) \quad \text{and} \quad in_l(0) = 1 .$$

FREE FOREST ALGEBRA

FREE FOREST ALGEBRA

For an alphabet A , the free forest algebra A^Δ is:

- The horizontal monoid is the set of forests over A .
- The vertical monoid is the set of contexts over A .
- The action is the substitution of forests in contexts.
- The in_l function takes a forest and transforms it into a context with a hole to the right of all the roots in the forest. Similarly for in_r but the hole is to the left of the roots.



MORPHISMS

MORPHISM $(\alpha, \beta) : (H, V) \rightarrow (G, V)$

It is a pair of monoid morphisms $(\alpha : H \rightarrow G, \beta : V \rightarrow W)$ with additional requirements ensuring that the operations are preserved:

$$\begin{aligned}\alpha(vh) &= \beta(v)\alpha(h) \\ \beta(in_l(h)) &= in_l(\alpha(h)) \quad \text{and} \quad \beta(in_r(h)) = in_r(\alpha(h))\end{aligned}$$

REMARK

The component α is determined by β via

$$\alpha(h) = \alpha(h + 0) = \alpha(in_l(h)0) = \beta(in_l(h))\alpha(0),$$

where $\alpha(0)$ must be the neutral element in G as α is a monoid morphism.

LEMMA (FREE ALGEBRA)

For every forest algebra (H, V) : every function $f : A \rightarrow V$ can be uniquely extended to a morphism $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ such that $\beta(a) = f(a)$ for every $a \in A$.

- Forests, contexts, and some operations.
- Forest algebras.
- **Recognition.**
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

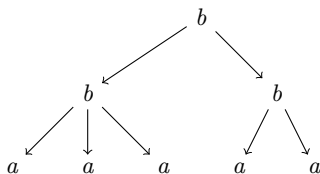
RECOGNITION

RECOGNIZING A LANGUAGE L OF A -FORESTS

L is **recognized** by a surjective **morphism** $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ if L is the inverse image $\alpha^{-1}(F)$ of some $F \subseteq H$. L is **recognizable** if it is recognized by a finite (H, V) .

EVEN NUMBER OF NODES

- Let L be the set of forest with even number of nodes.
- Consider the forest algebra where $H = V = \langle \{0, 1\}, 0, +(\text{mod } 2) \rangle$
- The action is also addition mod 2 and insertions are determined uniquely.
- The morphism maps a context to 0 if it has even number of nodes.
- The accepting set of L is 0.



$$\beta(a) = 1$$

$$\beta(b) = 1$$

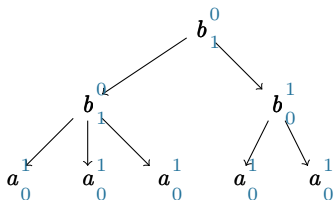
RECOGNITION

RECOGNIZING A LANGUAGE L OF A -FORESTS

L is **recognized** by a surjective **morphism** $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ if L is the inverse image $\alpha^{-1}(F)$ of some $F \subseteq H$. L is **recognizable** if it is recognized by a finite (H, V) .

EVEN NUMBER OF NODES

- Let L be the set of forest with even number of nodes.
- Consider the forest algebra where $H = V = \langle \{0, 1\}, 0, +(\text{mod } 2) \rangle$
- The action is also addition mod 2 and insertions are determined uniquely.
- The morphism maps a context to 0 if it has even number of nodes.
- The accepting set of L is 0.



$$\beta(a) = 1$$

$$\beta(b) = 1$$

RECOGNITION

RECOGNIZING A LANGUAGE L OF A -FORESTS

L is **recognized** by a surjective morphism $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ if L is the inverse image $\alpha^{-1}(F)$ of some $F \subseteq H$. L is **recognizable** if it is recognized by a finite (H, V) .

EVEN NUMBER OF NODES

- Let L be the set of forest with even number of nodes.
- Consider the forest algebra where $H = V = \langle \{0, 1\}, 0, +(\text{mod}2) \rangle$
- The action is also addition and insertions are determined uniquely.
- The morphism maps a context to 0 if it has even number of nodes.
- The accepting set of L is 0.

LABEL TESTABLE

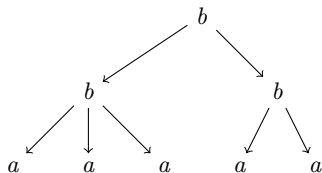
- L is **label-testable** if membership of t in L depends on the labels that occur in t .
- We set $H = V = \mathcal{P}(A)$ with union as operation.
- The action is also union, the insertions are determined.
- Recognizing morphism takes a context to the set of labels appearing in it.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

AUTOMATON FOR UNRANKED TREES (FIRST APPROXIMATION)

$$\mathcal{A} = \langle Q, A, \delta, F \subseteq Q \rangle$$

where $\delta : (A \times Q^* \rightarrow Q)$



AUTOMATA

AUTOMATON FOR UNRANKED TREES (FIRST APPROXIMATION)

$$\mathcal{A} = \langle Q, A, \delta, F \subseteq Q \rangle$$

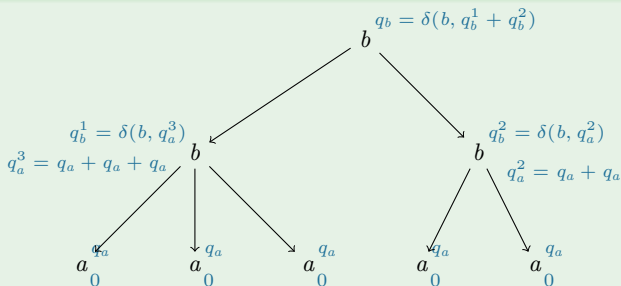
where $\delta : (A \times Q^* \rightarrow Q)$

DEFINITION (AUTOMATON FOR UNRANKED TREES)

$$\mathcal{A} = \langle (Q, 0, +), A, \delta, F \subseteq Q \rangle$$

where $\delta : (A \times Q \rightarrow Q)$

EXAMPLE



BETWEEN ALGEBRA AND AUTOMATA

ALGEBRA \rightarrow AUTOMATA

Take a morphism $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ and $F \subseteq H$.

We construct the automaton

$$\mathcal{A}^{(\alpha, \beta)} = \langle H, A, \delta, F \rangle \quad \text{where: } \delta(a, h) = \beta(a)h.$$

Claim: $L(\mathcal{A}^{(\alpha, \beta)}) = \alpha^{-1}(F)$.

AUTOMATA \rightarrow ALGEBRA

Take an automaton $\mathcal{A} = \langle (Q, +, 0), A, \delta : A \times Q \rightarrow Q, F \rangle$.

We construct the algebra (H, V) with

- $H = (Q, +, 0)$;
- $V : Q \rightarrow Q$ with the composition operation.
- act is the application $v(h)$.

Claim: Take the unique homomorphism $(\alpha_{\mathcal{A}}, \beta_{\mathcal{A}}) : A^\Delta \rightarrow (H, V)$ s.t.:

$$\beta_{\mathcal{A}}(a) = \delta(a) \quad \text{for all } a \in A.$$

We have $\alpha_{\mathcal{A}}(t) = t^{\mathcal{A}}$.

Remark: (H, V) so defined may be not faithful.

PLAN

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

SYNTACTIC ALGEBRA FOR FOREST LANGUAGES

Fix a language L of A -forests.

EQUIVALENCES

- Two A -forests s, t are L -equivalent if for every context p , either both or none of the forests ps, pt belong to L .
- Two A -contexts p, q are L -equivalent if for every forest t , the forests pt and qt are L -equivalent.

DEFINITION (SYNTACTIC FOREST ALGEBRA FOR L)

The **syntactic forest algebra** for L is the quotient of A^Δ with respect to L -equivalence. The **syntactic morphism** (α^L, β^L) assigns to every element of A^Δ its equivalence class.

FACT

A language L of A -forests is recognized by a the syntactic morphism (α^L, β^L) . Moreover, any morphism $(\alpha, \beta) : A^\Delta \rightarrow (H, V)$ that recognizes L can be extended by a morphism $(\alpha', \beta') : (H, V) \rightarrow (H^L, V^L)$ so that $\beta' \circ \beta = \beta^L$.

COMPUTING SYNTACTIC ALGEBRAS

If L is recognizable then the syntactic algebra is finite. Given an automaton for L , its syntactic algebra can be computed by a straightforward fixpoint algorithm.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

AN EXAMPLE: LABEL-TESTABLE LANGUAGES

THEOREM

A language is label testable iff its syntactic algebra satisfies the equations:

$$vv = v \quad vw = wv.$$

PROOF

- If L is label testable then its syntactic algebra satisfies the equations.
- If an algebra satisfies the equations then H is also idempotent and commutative.
- It also holds that $v(h) = h + v0$.
- This allows to show that every forest is equivalent to the one of the form:

$$a_1 0 + \cdots + a_n 0.$$

AN EXAMPLE: Σ_1 -LANGUAGES

Σ_1 FORMULAS

A Σ_1 formula is a formula of first-order logic, where only existential quantifiers appear in the quantifier prenex normal form. (There is the descendant order in the signature).

THEOREM

Let L be a forest language, and let (α, β) be its syntactic morphism. A language L is definable in Σ_1 if and only if $vh \in \alpha(L)$ implies $vwh \in \alpha(L)$.

PROOF

- \Rightarrow If $t \in L$ then any extension of t is in L .
- \Leftarrow For every $h \in H$ consider minimal size forests mapped to h . Every such forest is of bounded size.
- A forest is in L if it is an extension of a minimal forest that is mapped to $h \in \alpha(L)$.

AN EXAMPLE: EF-LOGIC

SYNTAX AND SEMANTICS

- If a is a letter, then a is a formula true in trees whose root label is a .
- EF formulas are closed under **boolean connectives**.
- If φ is an EF formula, then $EF\varphi$ is an EF formula true in trees having a *proper* subtree satisfying φ .

DEFINITION (DEFINABILITY IN EF)

A forest language L is definable in EF if for some $a \in A$ and for some φ of EF:

$$L = \{t : at \models \varphi\}.$$

THEOREM

A forest language is definable in EF if and only if its syntactic forest algebra satisfies the following equations

$$\begin{aligned}g + h &= h + g \\vh &= h + vh.\end{aligned}$$

EXAMPLE: $EF + F^{-1}$

DEFINITION ($EF + F^{-1}$)

To the definition of EF we add the clause:

- If φ is a formula then $F^{-1}\varphi$ is a formula true in a node of a tree if there is a proper ancestor satisfying φ .

THEOREM (BOJAŃCZYK)

Forest language is definable in $EF + F^{-1}$ iff its syntactic forest algebra satisfies the following:

$$\begin{aligned}h + h &= h & g + h &= h + g \\(vw)^\omega &= (vw)^\omega w(vw)^\omega \\(u_1 w_1)^\omega (u_2 w_2)^\omega &= (u_1 w_1)^\omega u_1 w_2 (u_2 w_2)^\omega && \text{for } u_1 \dashv u_2, w_1 \dashv w_2\end{aligned}$$

we need to assume here that $v_i, w_i \neq 1$.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

TOWARDS VARIETIES OF FOREST ALGEBRAS

FOREST ALGEBRA $\langle H, V, act, in_l, in_r \rangle$

- Two monoids: H and V . We denote their operations by $+$ and \cdot .
- An action $act : H \times V \rightarrow H$. We write vh for $act(h, v)$.
- Two operations $in_l, in_r : H \rightarrow V$.
- Axioms:

ACTION $(v \cdot w)h = v(wh)$;

INSERTION $in_l(g)h = g + h$ and $in_r(g)h = h + g$;

FAITHFULNESS for every two distinct $v, w \in V$ there is $h \in H$ with $vh \neq wh$;

REMARK

A subalgebra, quotient, or a homomorphic image of a forest algebra may not be a forest algebra (because of faithfulness.).

The solution is to take the faithful quotient of the result.

PSEUDOVARIETY

A **pseudovariety** of finite forest algebras is a collection \mathbb{V} of finite forest algebras with the following properties:

- \mathbb{V} is closed under **binary product**.
If $(H, V), (G, U) \in \mathbb{V}$ then $(H, V) \times (G, U) \in \mathbb{V}$.
- \mathbb{V} is closed under faithful quotients of **homomorphic images**.
If $(H, V) \in \mathbb{V}$ and (H', V') is its homomorphic image, then $\text{faith}(H', V') \in \mathbb{V}$.
- \mathbb{V} is closed under faithful quotients of **subalgebras**.
If $(H, V) \in \mathbb{V}$ and (H', V') is its subalgebra then $\text{faith}(H', V') \in \mathbb{V}$.

EXAMPLE

- Equation is a pair of terms in a signature of forest algebras (with variables of two types).
- An algebra satisfies an equation if for all valuations of variables the values of the two terms are the same.
- Everything definable by a finite set of equations is a variety.

EXAMPLES

THEOREM

A language is label testable iff its syntactic algebra satisfies the equations:

$$vv = v \quad vw = wv.$$

THEOREM

Let L be a forest language, and let (α, β) be its syntactic morphism. A language L is definable in Σ_1 if and only if $vh \in \alpha(L)$ implies $vwh \in \alpha(L)$.

THEOREM

A forest language is definable in EF if and only if its syntactic forest algebra satisfies the following equations

$$g + h = h + g$$
$$vh = h + vh .$$

PLAN

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

FIRST CORRESPONDENCE THEOREM

VARIETIES OF FOREST LANGUAGES

Let \mathbb{V} be a variety of algebras. For every finite alphabet A define

$$\mathcal{V}(A) = \{L \subseteq H_A : (H^L, V^L) \in \mathbb{V}\}.$$

We call \mathcal{V} the **variety of forest languages** associated to \mathbb{V} , and write

$$\mathbb{V} \mapsto \mathcal{V}.$$

THEOREM

The mapping $\mathbb{V} \mapsto \mathcal{V}$ is one-to-one.

SECOND CORRESPONDENCE THEOREM

A PREFIX OPERATOR

If L is a language of A -forests and p is a A -context then we define

$$p^{-1}L = \{t : pt \in L\}.$$

THEOREM

Let \mathcal{W} be an operator assigning to each finite alphabet A a family $\mathcal{W}(A)$ of A -languages. \mathcal{W} is a variety of languages if and only if the following three conditions hold:

- 1 for all finite alphabets A , $\mathcal{W}(A)$ is closed under boolean operations.
- 2 for all finite alphabets A , if $L \in \mathcal{W}(A)$ and p is an A -context then $p^{-1}L \in \mathcal{W}(A)$.
- 3 for all finite alphabets A and B , if $(\alpha, \beta) : A^\Delta \rightarrow B^\Delta$ is a homomorphism, and $L \in \mathcal{W}(B)$, then $\alpha^{-1}(L) \in \mathcal{W}(A)$.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- Related and further work.

APPLICATION: FIRST-ORDER LOGIC

FIRST-ORDER LOGIC OVER FOREST, $\text{FO}[\leq]$

- We have a unary predicate Q_a for every letter in the alphabet.
- We have \leq relation interpreted as a descendant relation in the tree.
- Example: $\exists x \forall y. (x \leq y)$ defines trees.

FACT

$\text{FO}[\leq]$ is a variety of forest languages.

FACT

Chain logic is a variety of forest languages.

REMARK

$\text{FO}[\textit{succ}]$ is not a variety of forest languages. Closure under inverse homomorphic images fails.

DEFINING VARIETIES BY EQUATIONS

- Let $\{l_i = r_i\}_{i>0}$ be an infinite set of equations.
- An algebra **ultimately satisfies** $\{l_i = r_i\}_{i>0}$ if it satisfies all but finitely many of these equations.
- A variety **ultimately defined by** $\{l_i = r_i\}_{i>0}$ is the set of finite algebras ultimately satisfying these equations.

THEOREM

If \mathbb{V} is a pseudovariety, then it is ultimately defined by some set of equations.

COROLLARY

There is a set of equations ultimately defining $\text{FO}[\leq]$.

- Forests, contexts, and some operations.
- Forest algebras.
- Recognition.
- Automata over forests.
- Syntactic forest algebras.
- Simple applications.
- Pseudovarieties.
- Eilenberg theorem for forest algebras.
- Applications.
- **Related and further work.**

TWO ORDERS

In unranked trees we have two orders:

- \leq_H the **horizontal order**, i.e., order between siblings.
- \leq_V the **vertical order**, i.e., ancestor-descendant order.

We write $FO[\leq_H, \leq_V]$ or $FO[\leq_V]$ to show which order is present.

UHV: Given a regular language L of unranked trees,
decide if L can be defined in $FO[\leq_H, \leq_V]$.



BHV: Given a regular language L of binary trees,
decide if L can be defined in $FO[\leq_H, \leq_V]$.



BV: Given a regular language L of binary trees,
decide if L can be defined in $FO[\leq_V]$.



UV: Given a regular language L of unranked trees,
decide if L can be defined in $FO[\leq_V]$.



CTL*: Given a regular language L of unranked trees,
decide if L can be defined in CTL*.

COMMENTS ON MISSING REDUCTIONS

FACT

The language “unranked trees with even number of nodes” is definable in $\text{MSOL}[\leq_H, \leq_V]$ but not in $\text{MSOL}[\leq_V]$.

FACT

The language “binary trees where all leaves have even depth” can be defined in $\text{FO}[\leq_H, \leq_V]$ but not in $\text{FO}[\leq_H]$.

RELATED WORK

- [Nivat & Podelski'87] Looking only at the structure contexts with application. Not enough to characterize FOL.
- [Wilke'96] Three sorted structure: labels, trees, contexts.
- [Salehi & Steinby' 07] Variety theorem for the Wilke's setting.
- [Esik'99]
- [Esik & Weil' 05] Preclones. Algebra of all term, i.e., contexts of arbitrary arity.
- [Thomas'84] Regular expressions for trees.
- [Heuter'89] Regular expressions for FOL.
- [Potthoff'] Star free=regular. Aperiodic is more than star free.
- [Bojańczyk & W.'04] Characterization of EF.
- [Benedikt & Segufin'05-07] Characterization of $\text{FO}[succ]$.
- [Bojańczyk '07] Characterization of $\text{EF} + F^{-1}$.

CONCLUSIONS AND SOME OTHER RESULTS

- An algebraic framework for recognizing forest languages. It is based on a new interpretation of a transformation semigroup.
- Basic results from the theory of semigroups carry over to forest algebras.
- The framework permits to express some characterizations in a simple way.
- “By design” there are some things not expressible in the framework. Ex. deterministic tree languages, or frontier languages.
- The variety theorem guarantees that there is an equational characterisation for some interesting logics.
- Considering unranked forests is interesting also for the case of binary forests.
- The notion of the wreath product of transformation semigroups generalizes naturally to forest algebras.
- As in the word case one gets the wreath product principle connecting the wreath product with the cascade product of automata, and the formula substitution.
- This gives characterizations of many known logics in terms of wreath products of simple varieties.