

Unranked Tree Algebra

Mikolaj Bojanczyk and Igor Walukiewicz

Warsaw University and LaBRI Bordeaux

Bonn meeting, June 7–9, 2006

The problem

Problem

Given a regular tree language decide if it is definable in FOL.

Language is given by a finite automaton (leaves to root)

$$\mathcal{A} = \langle Q, \Sigma, q_0, \delta : Q \times \Sigma \times Q \rightarrow Q, F \rangle$$

FOL over trees

$$P_a(x) \mid x \leq y \mid \neg\alpha \mid \alpha \wedge \beta \mid \exists x.\alpha$$

What kind of trees?

ranked	vs	unranked
ordered sons	vs	unordered sons
finite	vs	infinite

The problem

Problem

Given a regular tree language decide if it is definable in FOL.

Language is given by a finite automaton (leaves to root)

$$\mathcal{A} = \langle Q, \Sigma, q_0, \delta : Q \times \Sigma \times Q \rightarrow Q, F \rangle$$

FOL over trees

$$P_a(x) \mid x \leq y \mid \neg\alpha \mid \alpha \wedge \beta \mid \exists x.\alpha$$

What kind of trees?

ranked	vs	unranked
ordered sons	vs	unordered sons
finite	vs	infinite

The problem

Problem

Given a regular tree language decide if it is definable in FOL.

Language is given by a finite automaton (leaves to root)

$$\mathcal{A} = \langle Q, \Sigma, q_0, \delta : Q \times \Sigma \times Q \rightarrow Q, F \rangle$$

FOL over trees

$$P_a(x) \mid x \leq y \mid \neg\alpha \mid \alpha \wedge \beta \mid \exists x.\alpha$$

What kind of trees?

ranked	vs	unranked
ordered sons	vs	unordered sons
finite	vs	infinite

The problem

Problem

Given a regular tree language decide if it is definable in FOL.

Language is given by a finite automaton (leaves to root)

$$\mathcal{A} = \langle Q, \Sigma, q_0, \delta : Q \times \Sigma \times Q \rightarrow Q, F \rangle$$

FOL over trees

$$P_a(x) \mid x \leq y \mid \neg\alpha \mid \alpha \wedge \beta \mid \exists x.\alpha$$

What kind of trees?

ranked	vs	unranked
ordered sons	vs	unordered sons
finite	vs	infinite

In this talk

Finite unranked, ordered trees.

Logics over these trees

- EF (fragment of CTL)
- CTL*
- FOL
- PDL
- CL (chain logic: MSOL with quantification restricted to chains)

Goal

An algebraic characterization of all of these logics.
(Preferably a decidable one).

Recognizing words

Definition (Recognition)

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

Definition (Syntactic semigroup 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 $\langle \Sigma^+ / \sim_L, \cdot \rangle$.

Definition (Aperiodicity)

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

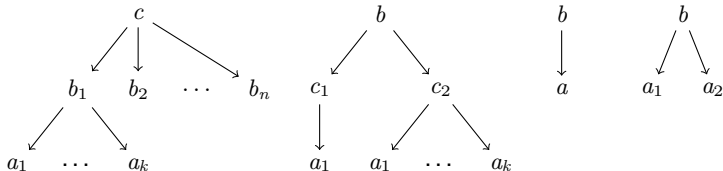
Theorem (Schützenberger, McNaughton & Papert)

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

Forests

Definition (Trees, Forests)

- A (A, B) -tree is a partial mapping $t : \mathbb{N}^* \rightarrow (A \cup B)$ with finite and prefix closed domain. We require that leaves have labels from A and internal nodes have labels from B .
- Forest is a finite sequence of trees.



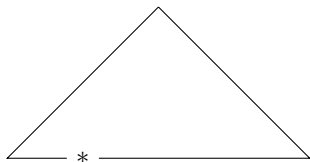
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and context composition $v_1 \cdot v_2$.



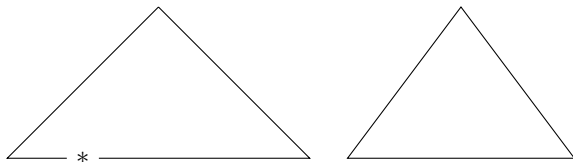
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and context composition $v_1 \cdot v_2$.



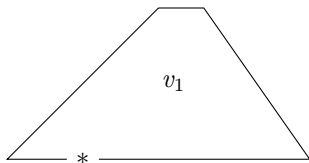
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and **context composition** $v_1 \cdot v_2$.



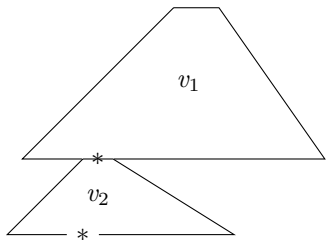
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and **context composition** $v_1 \cdot v_2$.



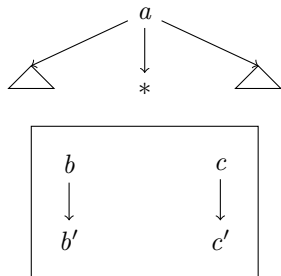
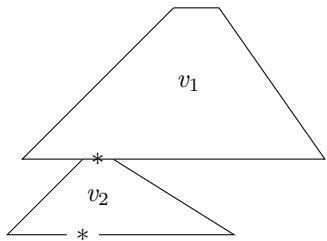
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and **context composition** $v_1 \cdot v_2$.



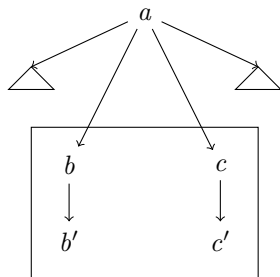
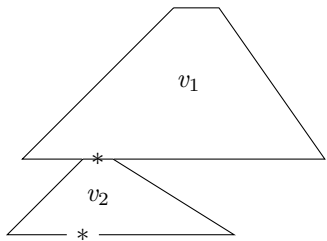
Context

Definition (Contexts)

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

We have two operations:

context substitution $v(t)$, and **context composition** $v_1 \cdot v_2$.



Forest and context semigroups

Forest semigroup

Forest semigroup consists of the set of (A, B) -forests, with concatenation operation (it is not commutative).

Context semigroup

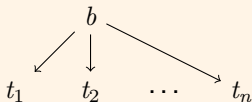
Context semigroup consists of the set of (A, B) -contexts with context composition.

Notation

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

$$t_1 + t_2, \quad v_1 \cdot v_2$$

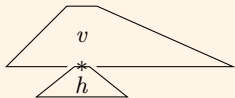
One letter trees and contexts are just denoted by letters : $b(t_1 + t_2 + \dots + t_n)$



Actions in forests

Action of contexts on forests

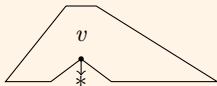
If v is a context and h is a forest then $v(h)$ is the tree obtained by the substitution of h in the hole of v .



we have $(v_1 \cdot v_2)(h) = v_1(v_2(h))$

Action of forests on contexts

If h is a forest and v a context then we have the context $in_l(h, v)$.



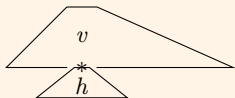
Similarly we have $in_r(h, v)$.

We also have $in^l(h, v)$ and $in^r(h, v)$ that insert h next to the roots.

Actions in forests

Action of contexts on forests

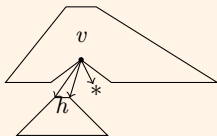
If v is a context and h is a forest then $v(h)$ is the tree obtained by the substitution of h in the hole of v .



we have $(v_1 \cdot v_2)(h) = v_1(v_2(h))$

Action of forests on contexts

If h is a forest and v a context then we have the context $in_l(h, v)$.



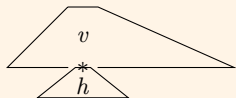
Similarly we have $in_r(h, v)$.

We also have $in^l(h, v)$ and $in^r(h, v)$ that insert h next to the roots.

Actions in forests

Action of contexts on forests

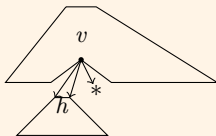
If v is a context and h is a forest then $v(h)$ is the tree obtained by the substitution of h in the hole of v .



we have $(v_1 \cdot v_2)(h) = v_1(v_2(h))$

Action of forests on contexts

If h is a forest and v a context then we have the context $in_l(h, v)$.



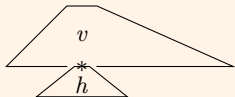
Similarly we have $in_r(h, v)$.

We also have $in^l(h, v)$ and $in^r(h, v)$ that insert h next to the roots.

Actions in forests

Action of contexts on forests

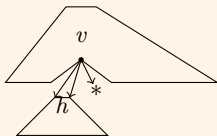
If v is a context and h is a forest then $v(h)$ is the tree obtained by the substitution of h in the hole of v .



we have $(v_1 \cdot v_2)(h) = v_1(v_2(h))$

Action of forests on contexts

If h is a forest and v a context then we have the context $in_l(h, v)$.



Similarly we have $in_r(h, v)$.

We also have $in^l(h, v)$ and $in^r(h, v)$ that insert h next to the roots.

Algebraic structure

Two semigroups and some actions

For two fixed alphabets A and B we have:

- A semigroup of forests with concatenation.
- A semigroup of contexts with context substitution.
- Action of contexts on forest $act(v, h)$: inserting a forest in a hole.
- Four actions of forest on contexts: putting forest on a side of a context or a hole.

$$in_l(h, v), \quad in_r(h, v), \quad in^l(h, v), \quad in^r(h, v)$$

Tree algebra

Definition (Tree pre-algebra)

(H, V, act) where H, V are semigroups and $act : V \times H \rightarrow H$ is a faithful action of V on H .

Remark: Faithful means that $act(v_1, \cdot) \neq act(v_2, \cdot)$ if $v_1 \neq v_2$.

Remark: Tree prealgebra is just a transition semigroup where the set acted upon is a semigroup.

Definition (Tree algebra)

Tree algebra is a tree prealgebra that is **saturated**: for every v and h there are contexts $in_l(h, v), in_r(h, v), in^l(h, v), in^r(h, v)$ satisfying:

$$in_l(h, v)(g) = v(h + g)$$

$$in_r(h, v)(g) = v(g + h)$$

$$in^l(h, v)(g) = h + v(g)$$

$$in^r(h, v)(g) = v(g) + h$$

The standard tree algebra

Definition (Free tree algebra)

The **free tree algebra** over alphabets A and B , $(A, B)^\Delta$, is the tree algebra $\langle (A, B) - \text{trees}, (A, B) - \text{contexts}, \text{act} \rangle$ where act is the action of inserting the tree into the hole of the context.

Remarks

- Free tree algebra is indeed a tree algebra.
- Free tree algebra is indeed free
 - every function $f : (A, B) \rightarrow (H, V)$ uniquely extends to a homomorphism $\bar{f} : (A, B)^\Delta \rightarrow (H, V)$.
(homomorphism of tree algebras is a function preserving all operations)

Recognition

Definition (Recognition)

A set L of (A, B) -forests is **recognized** by a morphism

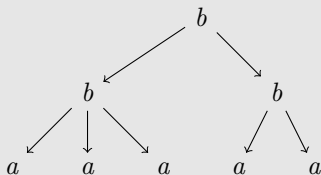
$$\alpha : (A, B)^\Delta \rightarrow (H, V)$$

if there is a set $F \subseteq H$ such that $\alpha^{-1}(F) = L$.

Example

Let L be the set of forests with even number of nodes.

We can recognize L with (H, V) where $H = V = \{0, 1\}$ and all operations are addition modulo 2.



$$\alpha(a) = 1$$

$$\alpha(b) = 1$$

Recognition

Definition (Recognition)

A set L of (A, B) -forests is **recognized** by a morphism

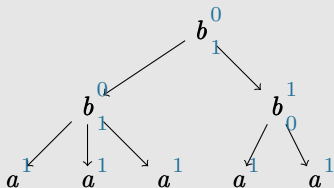
$$\alpha : (A, B)^\Delta \rightarrow (H, V)$$

if there is a set $F \subseteq H$ such that $\alpha^{-1}(F) = L$.

Example

Let L be the set of forests with even number of nodes.

We can recognize L with (H, V) where $H = V = \{0, 1\}$ and all operations are addition modulo 2.



$$\alpha(a) = 1$$

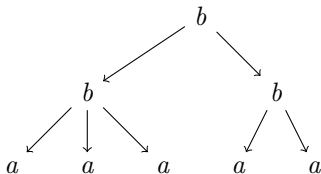
$$\alpha(b) = 1$$

Automata

Automaton for unranked trees (first approximation)

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

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



Automata

Automaton for unranked trees (first approximation)

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

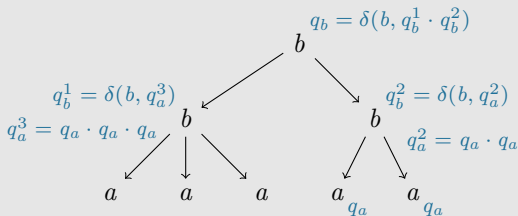
where $\delta : (A \rightarrow Q) \times (B \times Q^* \rightarrow Q)$

Definition (Automaton for unranked trees)

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

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

Example



Between algebra and automata

Algebra \rightarrow automata

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

We construct the automaton

$$\mathcal{A}^\alpha = \langle (Q, \cdot) = H, A, B, \delta, F \rangle$$

where: $\delta(a) = \alpha(a)$ and $\delta(b, h) = act(\alpha(b), h)$.

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

Automata \rightarrow algebra

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

We construct the algebra (H, V, act) with

- $H = (Q, \cdot)$;
- $V : Q \rightarrow Q$ with composition operation. ($\delta(b) : Q \rightarrow Q$).
- act is application $v(h)$.

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

$$\alpha_{\mathcal{A}}(a) = \delta(a) \text{ and } \alpha_{\mathcal{A}}(b) = \delta(b).$$

We have $\alpha_{\mathcal{A}}^{-1}(F) = L(\mathcal{A})$.

Syntactic tree algebra

Fix a language L of (A, B) -forests

Definition (Equivalences)

- Two nonempty (A, B) -forests g, h are L -equivalent if for every (perhaps empty) Σ -context v , either both or none of the forests $v(g), v(h)$ belong to L .
- Two nonempty (A, B) -contexts v, w are L -equivalent if for every nonempty Σ -forest h the forests $v(h), w(h)$ are L -equivalent.

Definition (Rational language)

A language L is rational if the above equivalences are finite.

Remark: It is enough that the horizontal one is finite.

Remark: The two equivalences are congruences on $(A, B)^\Delta$.

Syntactic tree algebra

Definition (Equivalences)

- Two nonempty (A, B) -forests g, h are L -equivalent if for every (perhaps empty) Σ -context v , either both or none of the forests $v(g), v(h)$ belong to L .
- Two nonempty (A, B) -contexts v, w are L -equivalent if for every nonempty Σ -forest h the forests $v(h), w(h)$ are L -equivalent.

Definition (Syntactic tree algebra for L)

Syntactic tree algebra for L is the quotient of the standard tree algebra $(A, B)^\Delta$ by the above relations.

Lemma

The syntactic tree algebra recognizes L and it is a quotient of any other tree algebra recognizing L .

Example: path languages

Path language

Language L is **path language** if $(t \in? L)$ depends only on the set of paths of t .

Example (“set” equations)

$$h + h = h \quad \text{and} \quad g + h = h + g \quad \text{for } g, h \in H$$

Membership in the language does not depend on the order nor on the multiplicity of subtrees.

Example (“thinning equation” equations)

$$v(g + h) = v(g) + v(h) \quad \text{for } v \in V, g, h \in H .$$

Lemma

Language is a path language iff its syntactic tree algebra satisfies the above equations.

Every tree is equivalent to a forest of its paths (without repetitions).

Example: EF

EF logic

- Eb is a formula true in forest with a root labelled b .
- The logic is closed under **boolean connectives**.
- If α is a EF-formula then $EF\alpha$ is an EF formula that is true in forests with a **proper subtree** satisfying α .

Theorem (Bojanczyk & W.)

A forest language is EF definable iff its syntactic algebra satisfies:

$$(vw)^\omega = (vw)^\omega v$$

$$v(h) = h + v(h)$$

$$g + h = h + g$$

$$h + h = h.$$

Cascade product of automata

Cascade product

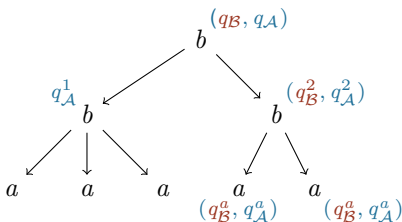
$$\mathcal{A} = \langle Q_{\mathcal{A}}, A, B, \delta_{\mathcal{A}}, F_{\mathcal{A}} \rangle \quad \mathcal{B} = \langle Q_{\mathcal{B}}, A, B \times Q_{\mathcal{A}}, \delta_{\mathcal{B}}, F_{\mathcal{B}} \rangle$$

The cascade product is

$$\mathcal{C} = \mathcal{B} \circ \mathcal{A} = \langle Q_{\mathcal{B}} \times Q_{\mathcal{A}}, A, B, \delta_{\mathcal{C}}, F_{\mathcal{C}} \rangle$$

where

- $F_{\mathcal{C}} = F_{\mathcal{B}} \times F_{\mathcal{A}}$.
- $\delta_{\mathcal{C}}(b, (q_{\mathcal{B}}, q_{\mathcal{A}})) = (\delta_{\mathcal{B}}((b, q_{\mathcal{A}}), q_{\mathcal{B}}), \delta_{\mathcal{A}}(b, q_{\mathcal{A}}))$.



$$q_{\mathcal{B}} = \delta_{\mathcal{B}}((b, q_{\mathcal{A}}^1 \cdot q_{\mathcal{A}}^2), q_{\mathcal{B}}^1 \cdot q_{\mathcal{B}}^2)$$

$$q_{\mathcal{B}}^2 = \delta_{\mathcal{B}}((b, q_{\mathcal{A}}^a \cdot q_{\mathcal{A}}^a), q_{\mathcal{B}}^a \cdot q_{\mathcal{B}}^a)$$

Wreath product

Definition (Wreath product)

- Take two tree algebras

$$\mathcal{B} = (H, V, \text{act}^{\mathcal{B}}) \quad \text{and} \quad \mathcal{A} = (G, W, \text{act}^{\mathcal{A}})$$

- The **wreath product** $\mathcal{C} = \mathcal{B} \circ \mathcal{A}$ is the tree algebra $(I, U, \text{act}^{\mathcal{C}})$
 - The horizontal semigroup I is the product semigroup $H \times G$.
 - The vertical semigroup U is $V^G \times W$ with multiplication:

$$(f, w) \circ_U (f', w') = (f'', w \circ_W w') \quad \text{where } f''(g) = f(w'(g)) \circ_V f'(g)$$

- The action $\text{act}^{\mathcal{C}}$ of U on I is:

$$\text{act}^{\mathcal{C}}((f, w), (h, g)) = (f(g)(h), w(g)) \quad \text{for } (f, w) \in V^G \times W, (h, g) \in H \times G.$$

Lemma

The wreath product of two tree algebras is a tree algebra.

Wreath product vs. cascade product

Wreath products recognize cascade products

- Take two automata \mathcal{A} , \mathcal{B} and their algebras $(H^{\mathcal{A}}, V^{\mathcal{A}})$, $(H^{\mathcal{B}}, V^{\mathcal{B}})$.
- Consider the cascade product $\mathcal{B} \circ \mathcal{A}$.
- The language $L(\mathcal{B} \circ \mathcal{A})$ is recognizable by $(H^{\mathcal{B}}, V^{\mathcal{B}}) \circ (H^{\mathcal{A}}, V^{\mathcal{A}})$.

Classes closed on wreath product

Definition

Let \mathbb{V}, \mathbb{W} be two classes of tree algebras. We put

$$\mathbb{W} \circ \mathbb{V} = \{\mathcal{B} \circ \mathcal{A} : \mathcal{B} \in \mathbb{W}, \mathcal{A} \in \mathbb{V}\}$$

$$\langle \mathbb{V} \rangle = \bigcup_{n \in \mathbb{N}} \mathbb{V}^n \quad \text{where } \mathbb{V}^n = \overbrace{\mathbb{V} \circ \dots \circ \mathbb{V}}^{n \text{ times}} .$$

We will be interested by $\langle \mathbb{V} \rangle$ for various \mathbb{V} defined equationally.

Basic classes of tree algebras

Idempotent if $s \cdot s = s$ for all $s \in S$.

Commutative if $s \cdot t = t \cdot s$ for all $s, t \in S$.

Aperiodic if there is $n \in \mathbb{N}$ such that $s^n = s^n \cdot s$ for all $s \in S$.

Definition (Equations on tree algebras)

Tree algebra (H, V) is

distributive $v(g \cdot h) = v(g) \cdot v(h)$

top $v(g) = v(h)$ for every $v \in V$ and $g, h \in H$.

Definition (Interesting classes of tree algebras)

- \mathbb{Y} distributive tree algebras where horizontal semigroup is commutative aperiodic and the vertical semigroup is aperiodic;
- \mathbb{Z} top tree algebras where horizontal semigroup is commutative aperiodic.

Characterization of FOL

Theorem (Bojanczyk & W.)

First-order logic $\equiv \langle \mathbb{Y} + \mathbb{Z} \rangle$.

Remark

The base classes allow to capture the operators and wreath product corresponds to substitution.

Remark

Similar characterizations are possible for other logics:

- CTL*, (the number of sons is irrelevant)
- Chain logic, (aperiodicity requirement is dropped)
- PDL, (the number of sons is irrelevant and aperiodicity dropped)

Varieties

Definition (Variety)

A **variety** of tree algebras is a class of tree algebras closed under subalgebras, quotients, and binary products.

Definition (Equations)

- An **equation** over (A, B) , $h_1 = h_2$, is a pair of horizontal elements of $(A, B)^\Delta$.
- An algebra (H, V) **satisfies the equation** if for all morphisms $\alpha : (A, B)^\Delta \rightarrow (H, V)$ we have $\alpha(h_1) = \alpha(h_2)$.
- An algebra (H, V) **ultimately satisfies** a set of equations $\mathcal{E} = \{e_0, e_1, \dots\}$ if it satisfies almost all equations.

Theorem (Variety theorem)

A class of algebras is a variety iff it is ultimately defined by a set of equations.

Varieties cont.

Definition (Language variety)

A **variety of forest languages** is class of forest languages that is closed under: boolean combinations, morphic preimages and cutting of contexts ($C^{-1}(L)$).

Fact

*FOL languages form a language variety.
Similarly for the other logics considered here.*

Fact

*A class of tree algebras associated to a variety of forest languages is a tree algebra variety.
A class of forest languages associated to a variety of tree algebras is a language variety.*

Corollary

There is a set of equations defining FOL.

Conclusions

- New algebraic framework for recognizing tree languages. It is based on a new interpretation of transformation semigroup.
- The framework permits to express nicely known characterizations.
- The characterization of FOL over trees refers to notions known from words “lifted” to trees.
- The characterization uses a kind of “wreath product principle”.

- We can obtain characterizations of CL, PDL, CTL* similar to that of FOL.
- These characterizations do not give decidability results.
- They suggest though that the case of unranked trees may be more natural than that of ranked trees (cf. the case of EF).
- Algebra is probably not necessary, but looks like a good way to understand what is happening.