Unranked Tree Algebra

Mikolaj Bojanczyk and Igor Walukiewicz

Warsaw University and LaBRI Bordeaux

ACI Cachan, March 13-14, 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)

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

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)

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

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)

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

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)

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

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

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

Contribution

An algebraic characterization of all of these logics.
(It does not give decidability though)

Recognizing words

Definition (Recognition)

A language L is recognized by a semigroup S if there are $h : \Sigma^* \to S$ 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 (Apperiodicity)

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

Definition (Contexts)

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

We have two operations:
context substitution $C[t]$, and context composition $C[D[]]$.

We have thus two semigroups:
forest with forest composition, and contexts with context composition.

Digression: Transformation semigroups

Definition (Semigorup)

A set with an associative operation $\langle S, \cdot \rangle$.

Definition (Transformation semigroup)

$\langle Q, S, act : S \times Q \to Q \rangle$ where Q is a set, S is a semigroup and act is an action:

$$act(s \cdot t, q) = act(s, act(t, q))$$

Example

- $\langle Q, PF(Q), \circ \rangle$ partial functions with composition.
- Take automaton $\mathcal{A} = \langle Q, \Sigma, \delta : Q \times \Sigma \to Q \rangle$. Define $\langle Q, \{\delta_w : w \in \Sigma^*\}, \cdot \rangle$.

Actions in forests

Example (Action of contexts on forests)

If v is a context and h is a fores then $v(h)$ is the substitution of h in the hole of v.

Example (Action of forests on contexts)

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

Tree algebra

Definition (Tree prealgebra)

(H, V, act) where H, V are semigroups and $act : V \times H \to H$ is an action of V on H.

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

Definition (Tree algebra)

(H, V, act, in_l, in_r); where (H, V, act) is a tree prealgebra and $in_l, in_r : H \times V \to V$ are two actions satisfying inserting conditions:

$$in_l(h, v)(g) = v(h \cdot g) \qquad in_r(h, v)(g) = v(g \cdot h)$$

The standard tree algebra

Definition (Standard tree algebra)

$$\mathit{Trees}(\Sigma) = (H^\Sigma, V^\Sigma, act^\Sigma, {in_l}^\Sigma, {in_r}^\Sigma)$$

- H^Σ is the set of forests over Σ with forest composition.
- V^Σ is the set of contexts over Σ with context composition.
- $act^\Sigma : V^\Sigma \times H^\Sigma \rightarrow H^\Sigma$ is the action on inserting a forest h into the hole of a context v.
- ${in_l}^\Sigma, {in_r}^\Sigma : H^\Sigma \times V^\Sigma \rightarrow V^\Sigma$ are the insertions of a forest h on the left (resp. right) side of the hole in v.

Recognition

Definition (Morphism)

A pair of functions $(\alpha, \beta) : (H, V, act, in_l, in_r) \rightarrow (G, W, act', in_l', in_r')$ where $\alpha : H \rightarrow G$, $\beta : V \rightarrow W$ and all operations are preserved.

Definition (Recognition)

A set L of forests is recognized by a morphism

$$(\alpha, \beta) : \mathit{Trees}(\Sigma) \rightarrow (H, V, act, in_l, in_r)$$

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, act, in_l, in_r) where $H = V = \{0, 1\}$ and all operations are addition modulo 2.

Syntactic tree algebra

Fix a language L of forests

Definition (Equivalences)

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

Definition (Regular language)

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

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

Remark: The two equivalences are congruences.

Syntactic tree algebra

Definition (Equivalences)

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

Definition (Syntactic tree algebra for L)

Syntactic tree algebra for L is the quotient of the standard tree algebra $\mathit{Trees}(\Sigma)$ by the above relation.

Lemma

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

Syntactic tree algebra

Definition (Equivalences)

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

Definition (Syntactic tree algebra for L)

Syntactic tree algebra for L is the quotient of the standard tree algebra $Trees(\Sigma)$ by the above relation.

Corollary

- *Regular* $\equiv$ *recognized.*
- *If some algebra recognizing* L *satisfies an equation then the syntactic algebra satisfies the equation.*

Example TJ_1

Example ("set" equations)

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

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

Example ("flatening" equations)

$$v(g \cdot h) = v(g) \cdot v(h) \qquad (v \circ w)(g) = v(h) \cdot w(h) \qquad \text{for } v, w \in V, g, h \in H \ .$$

Lemma

Language is label testable iff its syntactic tree algebra satisfies the above equations.

Wreath product

- Take two tree algebras
  $$\mathcal{B} = (H, V, act^{\mathcal{B}}, in_l{}^{\mathcal{B}}, in_r{}^{\mathcal{B}}) \text{ and } \mathcal{A} = (G, W, act^{\mathcal{A}}, in_l{}^{\mathcal{A}}, in_r{}^{\mathcal{A}})$$
- The wreath product $\mathcal{C} = \mathcal{B} \circ \mathcal{A}$ is the tree algebra $(I, U, act^{\mathcal{C}}, in_l{}^{\mathcal{C}}, in_r{}^{\mathcal{C}})$
  - The horizontal semigroup I is the product semigroup $H \times G$.
  - The vertical semigroup U is $V^G \times W$ with multiplication:
    $$(f, w) \circ_U (f', w') = (f'', w \circ_W w') \qquad \text{where } f''(g) = f(w'(g)) \circ_V f'(g)$$
  - The action $act^{\mathcal{C}}$ of U on I is:
    $$act^{\mathcal{C}}((f, w), (h, g)) = (f(g)(h), w(g)) \qquad \text{for } (f, w) \in V^G \times W, (h, g) \in H \times G.$$
  - The left insertion $in_l{}^{\mathcal{C}}$ of I on U is:
    $$in_l{}^{\mathcal{C}}((h, g), (f, w)) = (f', in_l{}^{\mathcal{A}}(g, w)) \qquad \text{where } f'(g') = in_l{}^{\mathcal{B}}(h, f(gg'))$$

Lemma

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

Wreath product (cont.)

Definition (Wreath product)

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

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

Example (Cartesian product)

The cartesian product $\mathcal{B} \times \mathcal{A}$ is a subalgebra of the wreath product $\mathcal{B} \circ \mathcal{A}$.

- The horizontal part is OK.
- The element (v, w) of the vertical part of $\mathcal{B} \times \mathcal{A}$ is represented by (f_v, w) of $\mathcal{B} \circ \mathcal{A}$; where f_v is the constant function with value v.

Wreath product (cont.)

Definition (Wreath product)

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

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

Example (Cartesian product)

The cartesian product $\mathcal{B} \times \mathcal{A}$ is a subalgebra of the wreath product $\mathcal{B} \circ \mathcal{A}$.

- The horizontal part is OK.
- The element (v, w) of the vertical part of $\mathcal{B} \times \mathcal{A}$ is represented by (f_v, w) of $\mathcal{B} \circ \mathcal{A}$; where f_v is the constant function with value v.

Classes closed on wreath product

Definition

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

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

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

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

Temporal logics over trees

- Logic UETL has two kinds of formulas: tree formulas and path formulas.
- The semantics of a tree formula is a set of trees.
- The semantics of a path formula is a set of pairs (tree,path).

Example

$[\mathsf{E}^2(\Sigma^* a \Sigma^*)]^* b$ is true in (t, π) if the leaf at the end of π has label b while all other nodes on the path have at least two independent descendants labelled a.

- The syntax of UETL is as follows:
  - Every letter a of the alphabet is a tree formula.
  - Tree formulas are closed under boolean operations.
  - If $k \in \mathcal{N}$ and ψ is a path formula then $\mathsf{E}^k \psi$ is a tree formula.
  - Every tree formula is a path formula.
  - Path formulas are closed under $\psi^*, \psi_1 + \psi_2, \neg\psi, \psi_1 \cdot \psi_2$.

Example

The formula $\mathsf{E}^2(a^* b(a+b)^*)$ is true in $\{a, b\}$-trees that have at least two incomparable b's. This property is not definable in PDL (nor in CTL*) over unranked trees. It is definable in first-order logic.

Other logics as fragments of UETL

Theorem

UETL	$\equiv$	*Chain Logic*
UETL without ψ^*	$\equiv$	*First-order logic.*
UETL without $\mathsf{E}^k\psi$ *for* $k > 1$	$\equiv$	*PDL.*
UETL without ψ^* *and* $\mathsf{E}^k\psi$ *for* $k > 1$	$\equiv$	*CTL***.*

We want to give algebraic characterizations of these logics.

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 $n \in \mathcal{N}$ such that $s^n = s^n \cdot s$ for all $s \in S$.

Definition (Distributive tree algebra)

Tree algebra (H, V, act, in_l, in_r) is distributive if

$$v(g \cdot h) = v(g) \cdot v(h) \qquad \text{for every } v \in V \text{ and } g, h \in H.$$

Definition (Interesting classes of distributive tree algebras)

- $\mathbb{X}$ horizontal semigroup is commutative aperiodic;
- $\mathbb{X}'$ horizontal semigroup is commutative idempotent;
- $\mathbb{Y}$ horizontal semigroup is commutative aperiodic and the vertical semigroup is aperiodic;
- $\mathbb{Y}'$ horizontal semigroup is commutative idempotent and the vertical semigroup is aperiodic;

Main theorem

Theorem (Main theorem)

Chain Logic	$\equiv$	$\langle \mathbb{X} \rangle$.
PDL	$\equiv$	$\langle \mathbb{X}' \rangle$.
First-order logic	$\equiv$	$\langle \mathbb{Y} \rangle$.
CTL*	$\equiv$	$\langle \mathbb{Y}' \rangle$.

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

Definition (Interesting classes of distributive tree algebras)

- $\mathbb{X}$ horizontal semigroup is commutative aperiodic;
- $\mathbb{X}'$ horizontal semigroup is commutative idempotent;
- $\mathbb{Y}$ horizontal semigroup is commutative aperiodic and the vertical semigroup is aperiodic;
- $\mathbb{Y}'$ horizontal semigroup is commutative idempotent and the vertical semigroup is aperiodic;

Main theorem

Theorem (Main theorem)

Chain Logic	$\equiv$	$\langle \mathbb{X} \rangle$.
PDL	$\equiv$	$\langle \mathbb{X}' \rangle$.
First-order logic	$\equiv$	$\langle \mathbb{Y} \rangle$.
CTL*	$\equiv$	$\langle \mathbb{Y}' \rangle$.

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

Definition (Interesting classes of distributive tree algebras)

- $\mathbb{X}$ horizontal semigroup is commutative aperiodic;
- $\mathbb{X}'$ horizontal semigroup is commutative idempotent;
- $\mathbb{Y}$ horizontal semigroup is commutative aperiodic and the vertical semigroup is aperiodic;
- $\mathbb{Y}'$ horizontal semigroup is commutative idempotent and the vertical semigroup is aperiodic;

Conclusions

- We have given algebraic characterizations of FO, CL, PDL, CTL* using the notions known from words “lifted” to trees.
- This is in part possible thanks to a new interpretation of transformation semigroup.
- The characterizations use a kind of “wreath product principle”.

- The presented characterizations do not give decidability results.
- They point out though that the case of unranked trees may be easier.
- Algebra is probably not necessary but looks like a good way to understand what is happening.