An Algebraic Theory for Regular Languages of Unranked Trees

Szeged, Hungary; September, 2006

A generalization of the algebraic theory for regular word languages (syntactic monoid, Eilenberg variety theorems, decomposition theorems, etc.) to unranked trees (no bound on number of children a node may have).

In many ways simpler and more elegant than theory for binary trees or trees of bounded rank.

Fundamental idea due to M. Bojanczyk and I. Walukiewicz. (But don't blame them for the errors and half-truths in this talk!)

Related work

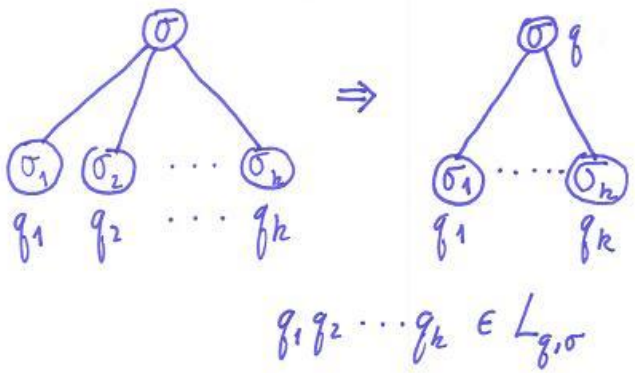
Much related work on binary trees and languages of trees of bounded rank—by almost everyone in the room except the speaker!

Also recent work of Benedikt and Ségoufin on tree languages definable in first-order logic with successor–their results on unranked trees *might* fit into this framework.

This theory is very much in its infancy; the **right** formulation may still be waiting to be discovered.

Motivation

Deterministic bottom-up automaton on Σ-labeled **trees** labels the tree nodes by states.



Root node labeled q iff $q_1 q_2 \cdots q_k \in L_{q,\sigma}$. The languages $L_{q,\sigma}$ are regular. Determinism means $L_{q,\sigma} \cap L_{q',\sigma} = \emptyset$ when $q \neq q'$.

There are two semigroups!

The languages $L_{q,\sigma}$ are all recognized by a single finite monoid (direct product of the syntactic monoids of these languages). $H =$ *horizontal semigroup*.

The maps $\sigma : q_1 \cdots q_k \mapsto q$ generate a semigroup of transformations on H. $V =$ *vertical semigroup.*

Tree Pre-algebra

A *tree pre-algebra* is a pair (H, V) where H is a monoid, V is a semigroup, and V acts on the right of H.

Always write the operation in H additively (and in our examples H is always commutative), write identity of H as 0. $h \cdot v$ or hv denotes the action of an element of V on an element of H.

We generally require the action of V on H to be *faithful*: If $hv = hv'$ for all $h \in H$ then $v = v'$. Any action can be collapsed to a faithful action.

A homomorphism of tree pre-algebras is a pair of homomorphisms $(\alpha, \beta) : (H_1, V_1) \to (H_2, V_2)$ satisfying

$$\alpha(hv) = \alpha(h)\beta(v)$$

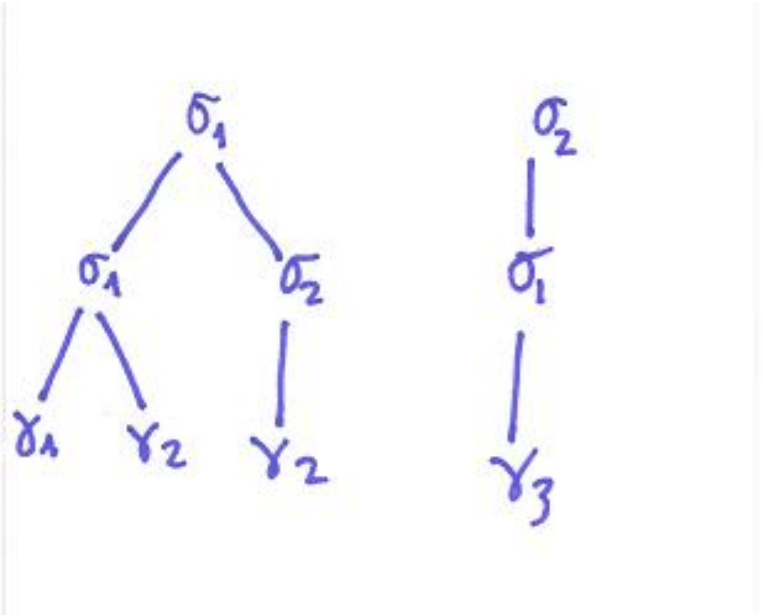
for all $h \in H_1$, $v \in V_1$.

Free Tree Pre-algebra $(H(\Sigma, \Gamma), V(\Sigma))$

Two sets of generators $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$, $\Sigma = \{\sigma_1, \ldots, \sigma_r\}$. $V(\Sigma) = \Sigma^+$. Elements of $H(\Sigma, \Gamma)$ are formal expressions like

$$((\gamma_1 + \gamma_2)\sigma_1 + \gamma_2\sigma_2)\sigma_1 + \gamma_3\sigma_1\sigma_2,$$

forests with leaves labeled by Γ and interior nodes labeled by Σ.



Usual universal property of free objects: Every pair of maps $\Gamma \to H$, $\Sigma \to V$ extends to homomorphism into (H, V).

Foundational Issues

Is H a semigroup of a monoid? Is V a semigroup or a monoid?

Is faithfulness necessary?

Forests or Trees?

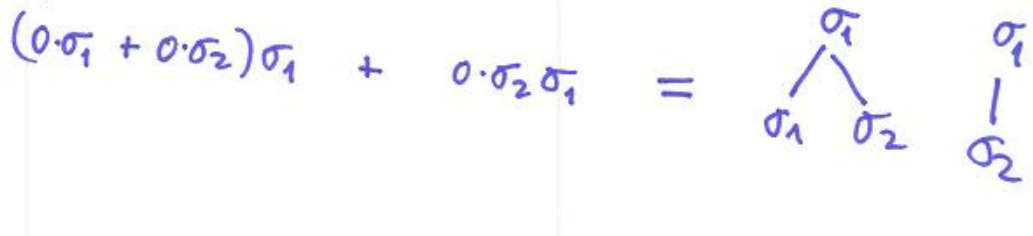
Different labels for leaves and interior nodes?

Tree prealgebras or Tree algebras? (We'll see these later.)

How do we handle a single alphabet of labels?

If you make H a monoid, then its zero is the empty forest. But then what is $0 \cdot \sigma$?

H should probably be a semigroup, but if you allow H to be a monoid, then the forests in which *every* leaf is 0 provide a good model for Σ-labeled trees and forests. We denote this smaller prealgebra $(H(\Sigma), V(\Sigma))$.



Monogenicity

A tree pre-algebra (H, V) is *monogenic* if it is generated by $0 \in H$ and V.

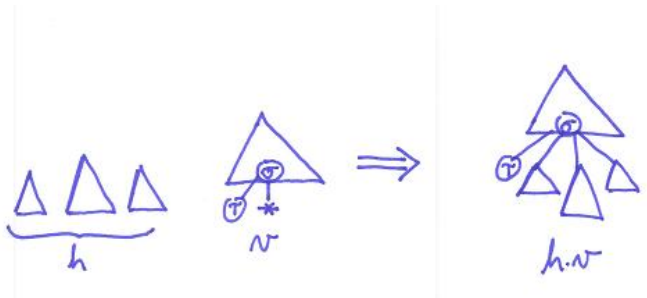
A homomorphism from a monogenic tree prealgebra is completely determined by its value on the vertical semigroup.

If (H, V) is monogenic then every map from Σ to V extends to a homomorphism from $(H(\Sigma), V(\Sigma))$ into (H, V).

Contexts

A *context* is a tree with a "hole" where one of its leaves should be.

In $(H(\Sigma), V(\Sigma))$ one can extend the action of letters on forests to the action of contexts on forests.



No "empty" contexts–every context v has at least one vertex with a label in σ, and so hv is always a tree.

This leads to an extension $(H(\Sigma), \hat{V}(\Sigma))$ of the original prealgebra.

Tree Algebras

In general, we can extend any tree prealgebra (H, V) to a larger one $(H, \hat{V})$ by closing under the operations

$$v \mapsto g * v, v \mapsto v * g, g \in H,$$

where

$$h(g * v) = (h + g)v, h(v * g) = (g + h)v.$$

A tree prealgebra closed under these operations is called a *tree algebra*.

Every homomorphism $(H_1, V_1) \to (H_2, V_2)$ has a unique extension to a homomorphism $(H_1, \hat{V}_1) \to (H_2, \hat{V}_2)$.

From now on we usually understand (H, V) to denote a tree algebra, so that $V = \hat{V}$.

Language Recognition

A set of forests $L \subseteq H(\Sigma)$ is recognized by a tree algebra (H, V) if and only if there is a homomorphism $(\alpha, \beta) : (H(\Sigma), V(\Sigma)) \to (H, V)$ and $X \subseteq H$ such that $L = \alpha^{-1}(X)$.

A set L of trees in $H(\Sigma)$ is recognized by (H, V) if and only if

$$L = \bigcup_{\sigma \in \Sigma} L_\sigma \sigma,$$

where each L_σ is recognized by (H, V). We also say (α, β) recognizes L.

Fact: L is a regular tree language iff it is recognized by a finite tree algebra (H, V).

Syntactic Tree Algebra

L a set of trees in $H(\Sigma)$. Define, for $h, h' \in H(\Sigma)$, $h \sim_L h'$ if and only if for all contexts v, $hv \in L \Leftrightarrow h'v \in L$.

It follows easily that $h_i \sim_L h'_i$ for $i = 1, 2$ implies $h_1 + h_2 \sim_L h'_1 + h'_2$, so that $\sim_L$ is a congruence on $H(\Sigma)$. Quotient denoted H_L.

We also have $h \sim_L h'$ implies $hv \sim_L h'v$, so that $\hat{V}(\Sigma)$ acts on H_L. Collapse to make this faithful, giving *syntactic tree algebra* (H_L, V_L) of L, and *syntactic morphism*

$$(\alpha_L, \beta_L) : (H(\Sigma), \hat{V}(\Sigma)) \to (H_L, V_L).$$

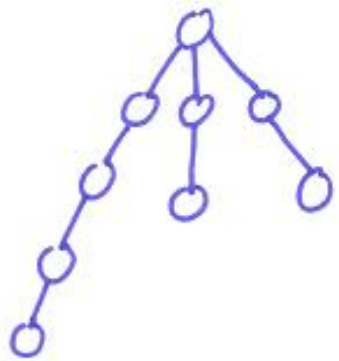
The Fundamental Theorem

Theorem

$(\alpha, \beta) : (H(\Sigma), \hat{V}(\Sigma)) \to (H, V)$ *recognizes* L *if and only if* (α_L, β_L) *factors through* (α, β).

An Example

$\Sigma = \{\sigma\}$. L is the set of trees in which every node has at most one child, with the possible exception of the root.



Classes of $\sim_L$: empty forest (0), forests consisting of a single path (1), sums of two or more such paths (2), everything else (∞.)

Observe $x + \infty = \infty$, $1 + 1 = 1 + 2 = 2$. $0\sigma = 1\sigma = 1$, $2\sigma = \infty\sigma = \sigma$.

A tree $h\sigma$ is in L if and only if $\alpha_L(h) \neq \infty$.

Why did we not use a simpler definition of recognition for tree languages?

Why not just say a set of L of trees is recognized by (H, V) if $L = \alpha^{-1}(X)$ for some $X \subseteq H$?

This fails to give canonical minimal algebra recognizing L: For the above example, $\{0, 1, 2, \delta, \infty\}$ with $\delta + \delta = \infty$ and $\{0, 1, 2, \epsilon, \infty\}$ with $\epsilon + \epsilon = \epsilon$ both recognize L in this stronger sense, but do not have a common quotient that recognizes L.

Varieties and the Eilenberg Correspondence

A variety of finite tree algebras is a family $\mathbf{V}$ of finite tree algebras closed under direct products, quotients, and subalgebras.

Note: For monogenic tree algebras, "subalgebra" and "direct product" must be qualified.

Let $\mathcal{T}_V(\Sigma)$ be the family of tree languages over Σ whose syntactic tree algebras belong to $\mathbf{V}$.

Theorem

$\mathbf{V} \mapsto \mathcal{T}_{\mathbf{V}}$ *is one-to-one.*

Eilenberg Correspondence, continued

$\mathcal{F}$ assigns a family of Σ-labeled forest languages to each alphabet Σ.

$\mathcal{T}_{\mathcal{F}}(\Sigma)$ consists of the languages $\bigcup_{\sigma\in\Sigma} U_\sigma$, where the U_σ are in $\mathcal{F}(\Sigma)$.

Theorem

$\mathcal{T}_{\mathcal{F}} = \mathcal{T}_{\mathbf{V}}$ for some variety $\mathbf{V}$ if and only if (i) $\mathcal{F}(\Sigma)$ is closed under boolean operations; (ii) If $L \in \mathcal{T}_{\mathcal{F}}(\Sigma)$ and $v \in \hat{V}(\Sigma)$, then $Lv^{-1} = \{h \in H(\Sigma) : hv \in L\} \in \mathcal{F}(\Sigma)$, (iii) if $(\alpha, \beta) : (H(\Gamma), \hat{V}(\Gamma)) \to (H(\Sigma), \hat{V}(\Sigma))$ is a homomorphism, and $L \in \mathcal{F}(\Sigma)$, then $\alpha^{-1}(L) \in \mathcal{F}(\Gamma)$.

Why do we consider algebras rather than prealgebras?

(Example due to M. Bojanczyk) Suppose membership of a tree in a regular language is determined by the set of labels of its leaves.

This property can be characterized by the identites

$$0\sigma v = 0\sigma + 0v, h_1 + h_2 = h_2 + h_1, h + h = h,$$

where any *context* can be substituted for v, but only a letter can be substituted for σ.

It is *not* preserved under inverse images of morphisms between free tree algebras, although it is preserved under inverse images of morphisms between free tree prealgebras.

Different kinds of varieties—analogous to $\mathcal{C}$-varieties for words.

Logically-defined classes of languages form varieties!

In practice, it is usually easy to show that the class of tree languages defined by some fragment of first-order or temporal logic satisfies the closure properties given in the Eilenberg theorem.

Consider, for example, $FO[<]$, where $x < y$ means node x is an ancestor of node y.

Easy to see that if the forest languages U_σ are all first-order definable, then so is the tree language $\bigcup_{\sigma\in\Sigma} U_\sigma\sigma$.

Closure properties follow from a simple application of Ehrenfeucht games: For example, one shows that if $(\alpha, \beta) : (H(\Gamma), V(\Gamma)) \to (H(\Sigma), V(\Sigma))$ is a homomorphism, and two forests $h_1, h_2 \in H(\Gamma)$ are indistinguishable in the r-round game, then the same is true for $\alpha(h_1), \alpha(h_2)$. This implies closure under inverse images for homomorphisms.

Logically-defined classes form varieties!

So, logically-defined classes of regular tree languages admit characterizations in terms of tree algebras.

Moreover, every variety is defined by a sequence of identities, or a set of pseudoidentities. and such identities, if found, can lead to effective characterizations.

Wreath Product

The wreath product, which is an operation on transformation semigroups, extends naturally to this setting:

$$(H_2, V_2) \circ (H_1, V_1) = (H_2 \times H_1, V_2^{H_1} \times V_1),$$

where

$$(h_2, h_1)(f, v_1) = (h_2 f(h_1), h_1 v_1).$$

One shows in the usual manner that the composition of two elements of the form (f, v) has the same form. Thus the wreath product is a tree prealgebra.

Easy to show that the wreath product of two tree algebras is a tree algebra.

A Sampling of Results-EX

Example: Consider the tree algebras that satisfy the identities:

$$g + h = h + g, g + g = g, gv_1 \cdots v_k = hv_1 \cdots v_k,$$

for some $k > 0$.

One can prove that a tree algebra satisfies these identities if and only if it embeds divides an iterated wreath product of copies of

$$(\{0, \infty\}, \{c_0, c_\infty\}),$$

where c_a denotes the constant map to a.

This is analogous to an old result of Stiffler on wreath products of transformation semigroups: A semigroup is definite if and only if it divides a wreath product of copies of $(\{a, b\}, \{c_a, c_b\})$.

This readily implies that these are exactly the syntactic algebras of tree languages definable in the temporal logic $TL[EX]$. ($EX\phi$ means there is a child satisfying ϕ.) (Compare results of Bojanczyk and Walukiewicz on EX for binary trees.)

A Sampling of Results-EF

Example: Consider the tree algebras that satisfy the identities:

$$g + h = h + g, g + g = g, (g + h)v = gv + (g + h)v.$$

One can prove that a tree algebra satisfies these identities if and only if it embeds in an iterated wreath product of copies of

$$(\{0, \infty\}, \{1, 0\}),$$

where 1 is the identity transformation, and 0 the constant map to ∞.

This is analogous to an old result of Stiffler on wreath products of transformation semigroups: A monoid is **R**-trivial if and only if it divides a wreath product of copies of $(\{0, 1\}, \{0, 1\})$.

This readily implies that these are exactly the syntactic algebras of tree languages definable in the temporal logic $TL[EF]$. ($EF\phi$ means there is a descendant satisfying ϕ.) (Due to M. Bojanczyk).

A Sampling of Results–Partially-ordered tree algebras and Simon's Theorem

L is defined by a boolean combination of Σ_1-sentences if and only if it is recognized by a tree algebra (H, V) satisfying the following conditions:

(i) H is commutative;

(ii) H admits a partial order $\leq$ that is compatible with addition ($h_i \leq h_i'$ for $i = 1, 2$, implies $h_1 + h_2 \leq h_1' + h_2'$), and with the action ($h \leq h'$ implies $hv \leq h'v$).

(iii) $hv \leq h$ for all $h \in H$, $v \in V$.

This does not immediately yield an effective criterion. A necessary condition is that the syntactic algebra is horizontally aperiodic and commutative and vertically $\mathcal{J}$-trivial. Is this condition sufficient?

A Sampling of Results-First-order logic

(Bojanczyk and Walukiewicz) L is in $FO[<]$ if and only if L is recognized by an iterated wreath product of tree algebras (H, V), each having one of the following two types:

1. H aperiodic and commutative, $hv = h'v$ for all $h, h \in H$, $v \in V$.

2. H is idempotent and commutative, V is aperiodic, and $(h + h')v = hv + h'v$ for all $h, h' \in H$, $v \in V$.

This does not directly yield an effective criterion–an important open problem.

Compare Benedikt and Ségoufin on first-order logic with successor. Here we have a decidable criterion and we know the class forms a variety of tree languages—can we derive/express an effective criterion in more algebraic language?