Tree Algebras

Saeed Salehi

&

Magnus Steinby

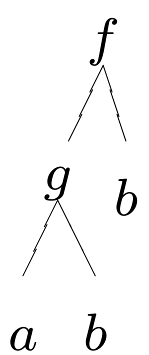
Turku Centre for Computer Science

saeed@cs.utu.fi

steinby@utu.fi

Trees: terms over a ranked alphabet Σ, $\mathtt{T}_\Sigma$.

Example $\Sigma_0 = \{a, b\}$, $\Sigma_2 = \{f, g\}$:



$$= f\Big(g(a,b), b\Big).$$

Contexts: terms over $\Sigma \cup \{\xi\}$ in which exactly one leaf is ξ, $\mathtt{C}_\Sigma$.

Congruences of a tree language $T \subseteq \mathtt{T}_\Sigma$:

(1) for trees, $t \sim^T t'$ iff

$$p[t] \in T \leftrightarrow p[t'] \in T$$

for every context p

Syntactic algebra of $T = \mathtt{T}_\Sigma / \sim^T$

▶ String case: Nerode Congruence, Minimal Automata.

(2) for contexts, $p \approx^T p'$ iff

$$q[p[t]] \in T \leftrightarrow q[p'[t]] \in T$$

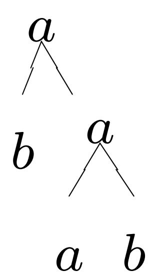
for all trees t, contexts q

Syntactic monoid of $T = \mathtt{C}_\Sigma / \approx^T$

▶ String case: Myhill/Syntactic Congruence, Syntactic monoid/semigroup.

Binary A-trees and A-contexts

$\Sigma_0^A = \{c_a \mid a \in A\}$ $\Sigma_2^A = \{f_a \mid a \in A\}$

Example 

$= f_a\Big(c_b, f_a(c_a, c_b)\Big)$

$T_A =$ set of A-trees

- $c_a \in T_A$ for every $a \in A$
- $f_a(t_1, t_2) \in T_A$ if $a \in A$ and $t_1, t_2 \in T_A$
- THAT IS IT!

C_A = set of A-contexts:

[• $\xi \in C_A$]

• $f_a(\xi, t), f_a(t, \xi) \in C_A$ if $a \in A$ and $t \in T_A$

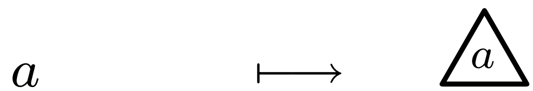
• $f_a(p, t), f_a(t, p) \in C_A$ if $a \in A$, $t \in T_A$ and $p \in C_A$

• THAT IS IT!

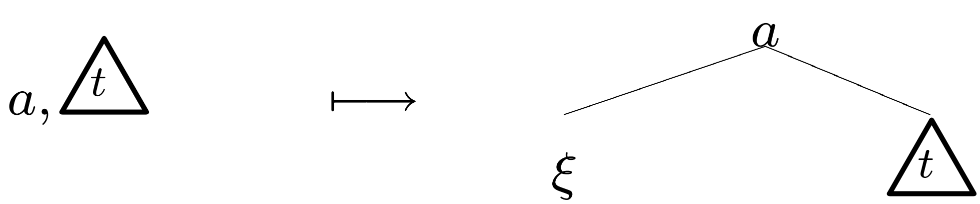
Signature of tree algebras $\Gamma = \{\iota, \kappa, \lambda, \rho, \eta, \sigma\}$

3 sorts: LABEL, TREE, CONTEXT

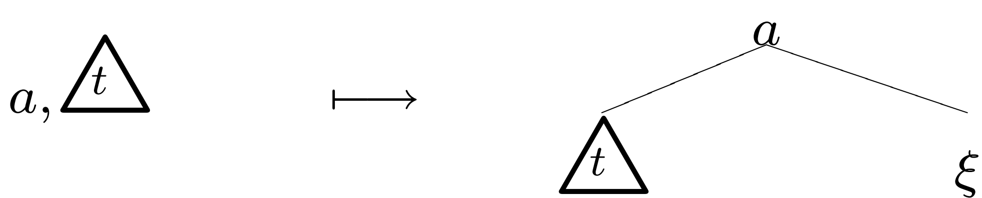
ι :



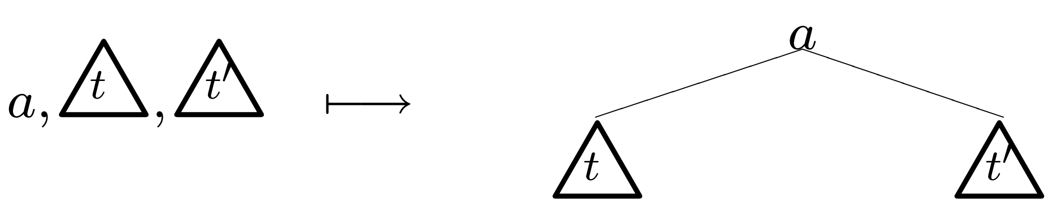
λ :



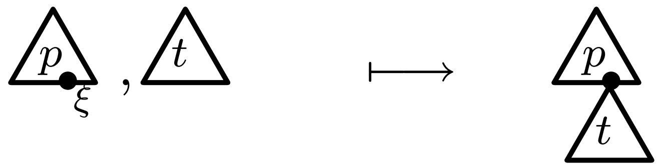
ρ :



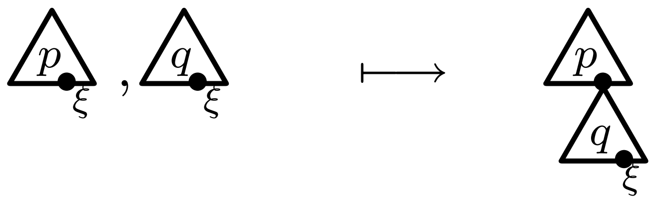
κ :



η :



σ :



$$\iota : LABEL \to TREE$$

$$\lambda : LABEL \times TREE \to CONTEXT$$

$$\rho : LABEL \times TREE \to CONTEXT$$

$$\kappa : LABEL \times TREE \times TREE \to TREE$$

$$\eta : CONTEXT \times TREE \to TREE$$

$$\sigma : CONTEXT \times CONTEXT \to CONTEXT$$

Γ-Algebra of A-trees and A-contexts

$\mathcal{T}_A = \langle A, T_A, C_A, \Gamma \rangle$

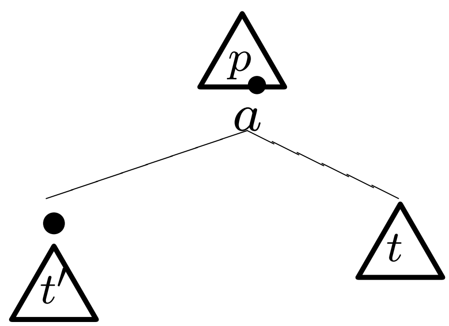
- $\iota^{\mathcal{T}_A}(a) = c_a$, • $\kappa^{\mathcal{T}_A}(a, t, t') = f_a(t, t')$,
- $\lambda^{\mathcal{T}_A}(a, t) = f_a(\xi, t)$, etc

Tree Algebra= a Γ-algebra satisfying Wilke's axioms:

- $\sigma(\sigma(p, q), r) = \sigma(p, \sigma(q, r))$ $p \circ (q \circ r) = (p \circ q) \circ r$
- $\eta(\sigma(p, q), t) = \eta(p, \eta(q, t))$ $(p \circ q)[t] = p[q[t]]$
- $\eta(\lambda(a, t), t')) = \kappa(a, t', t)$
- $\eta(\rho(a, t), t')) = \kappa(a, t, t')$

$\mathcal{T}_A$ is a tree algebra: all Wilkes' identities hold in A-trees and A-contexts.

Example: $\eta(\sigma(p, \lambda(a, t)), t') = \eta(p, \kappa(a, t', t))$



is derivable from the second and the third axioms by $q = \lambda(a, t)$.

Theorem This axiom system is sound and complete: every identity true in $\mathcal{T}_A$ is provable in the system, and vice versa.

Yield Function:

$$\mathbf{Y}(\iota(a)) = c_a$$

$$\mathbf{Y}(\kappa(a,t,t')) = f_a(\mathbf{Y}(t),\mathbf{Y}(t'))$$

$$\mathbf{Y}(\lambda(a,t)) = f_a(\xi,\mathbf{Y}(t))$$

$$\mathbf{Y}(\rho(a,t)) = f_a(\mathbf{Y}(t),\xi)$$

$$\mathbf{Y}(\eta(p,t)) = \mathbf{Y}(p)[\xi/\mathbf{Y}(t)]$$

$$\mathbf{Y}(\sigma(p,q)) = \mathbf{Y}(p)[\xi/\mathbf{Y}(q)]$$

Theorem Ker($\mathbf{Y}$) $=$ identities provable in Wilke's system.

Any axiom system satisfying the above theorem can be used for axiomatizing tree algebras.

Theorem Term-rewriting system

- $\sigma(\sigma(p,q),r) \longrightarrow \sigma(p,\sigma(q,r))$

- $\eta(\sigma(p,q),t) \longrightarrow \eta(p,\eta(q,t))$

- $\eta(\lambda(a,t),t')) \longrightarrow \kappa(a,t',t)$

- $\eta(\rho(a,t),t')) \longrightarrow \kappa(a,t,t')$

is convergent.

And the congruence it generates consists of the identities derivable from Wilke's axioms.

Corollary The Word Problem for tree algebras is solvable.

Characterizing families of tree languages

Congruence relations of $V \subseteq T_A$:

- $t \sim_T^V t' \equiv \forall p \in C_A \Big(\eta(p,t) \in V \leftrightarrow \eta(p,t') \in V \Big)$

- $p \sim_C^V p' \equiv \forall q \in C_A \forall t \in T_A$
  $$\Big(\eta(\sigma(q,p),t) \in V \leftrightarrow \eta(\sigma(q,p'),t) \in V \Big)$$

- $a \sim_L^V a' \equiv \forall p \in C_A \forall t,t' \in T_A$
  $$\Big(\eta(p,\iota(a)) \in V \leftrightarrow \eta(p,\iota(a')) \in V \Big) \ \&$$
  $$\Big(\eta(p,\kappa(a,t,t')) \in V \leftrightarrow \eta(p,\kappa(a',t,t')) \in V \Big)$$

Syntactic Tree Algebra =

Syntactic Algebra + Syntactic monoid

Steinby Thomas

Syntactic Tree Algebra of V =

$\langle A/\sim_A^V, \underbrace{\text{Syn Alg of } V}_{TREE}, \underbrace{\text{Syn monoid of } V}_{CONTEXT} \rangle$

not A

CounterExample $U = a \cdot (b \cup c)^+$: $b \sim_A^U c$, $b \neq c$.

Example L is regular iff its Syntactic Tree Algebra is finite.

Example Wilke's axiomatization of k-frontier testable tree languages. This can be simplified by taking out the CONTEXT sort.

...

Also The Syntactic Tree Algebras of tree languages can be characterized Γ-algebraically.