Compositions of Extended Top-down Tree Transducers

Andreas Maletti

March 30, 2007

Short Introduction

Motivation

- Extended tree transducers are used in machine translation [Knight & Graehl 05, Shieber 04]
- Compositions occur naturally
  1. transducers for specific (small) tasks are easier to train
  2. small transducers are simpler to understand
  3. "component" tree transducers can be reused

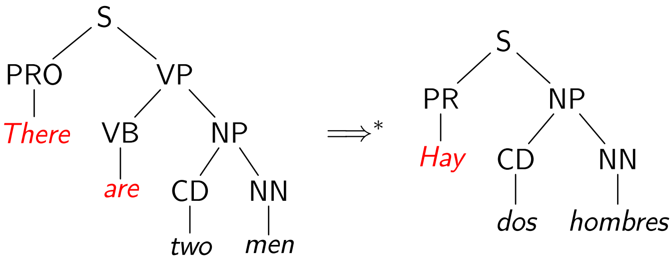
Short Introduction

Motivation

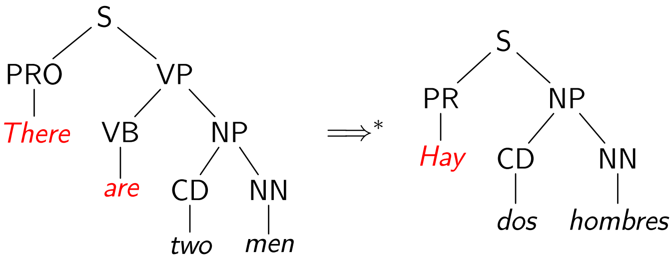
- Extended tree transducers are used in machine translation [Knight & Graehl 05, Shieber 04]
- Compositions occur naturally
  1. transducers for specific (small) tasks are easier to train
  2. small transducers are simpler to understand
  3. "component" tree transducers can be reused
- Extended tree transducers are (essentially) as powerful as tree substitution grammars [Knight & Graehl & Hopkins 07]
- Closure under composition of synchronous tree substitution grammar transformations open (since introduction in 80's)

Outline

Principal Problem of Top-down Tree Transducers



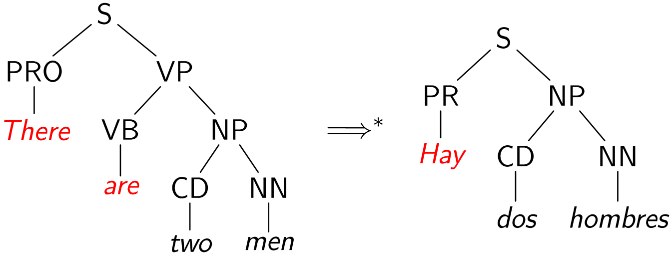
Principal Problem of Top-down Tree Transducers



Notes:

- difficult to implement without regular look-ahead
- solution: use copying

Principal Problem of Top-down Tree Transducers



Notes:

- difficult to implement without regular look-ahead
- solution: use copying — No! — closure under composition

The new device

Why do we not have multi-level rules?

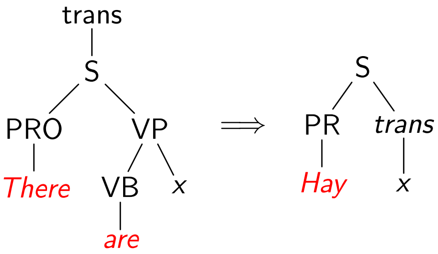
[Knight, Graehl: *Training Tree Transducers.* HLT-NAACL 2004]

The new device

Why do we not have multi-level rules?

[Knight, Graehl: *Training Tree Transducers.* HLT-NAACL 2004]

Then we could have rules like



Formal Syntax

Definition (cf. Knight & Graehl 04)

An extended top-down tree transducer is a tuple

$$M = (Q, \Sigma, \Delta, S, R)$$

- Q a finite set of states
- Σ and Δ input and output ranked alphabet, respectively;
- $S \subseteq Q$ a set of initial states

Formal Syntax

Definition (cf. Knight & Graehl 04)

An extended top-down tree transducer is a tuple

$$M = (Q, \Sigma, \Delta, S, R)$$

- Q a finite set of states
- Σ and Δ input and output ranked alphabet, respectively;
- $S \subseteq Q$ a set of initial states
- $R \subseteq Q(T_\Sigma(X)) \times T_\Delta(Q(X))$ a finite set of rules such that

  $$\mathrm{var}(r) \subseteq \mathrm{var}(l)$$

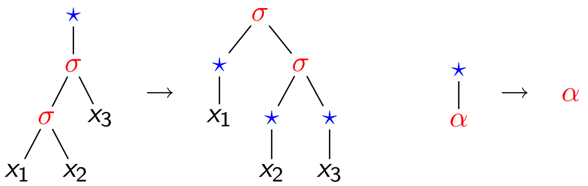
  and l is linear for every rule $(l, r) \in R$.

An extended top-down tree transducer

Example

- $Q = S = \{\star\}$;
- $\Sigma = \Delta = \{\sigma^{(2)}, \alpha^{(0)}\}$;
- R contains the rules

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$



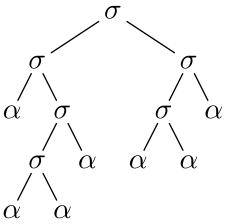
… in action

Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

Derivation:



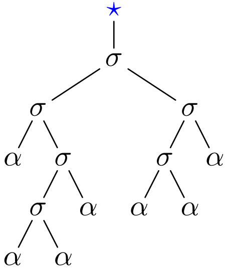
… in action

Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

Derivation:



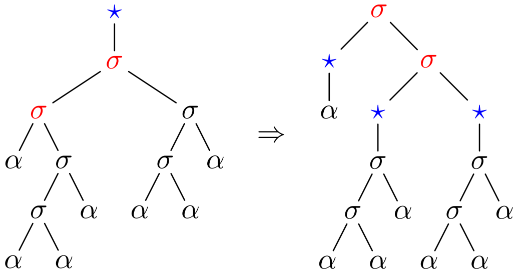
… in action

Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

Derivation:



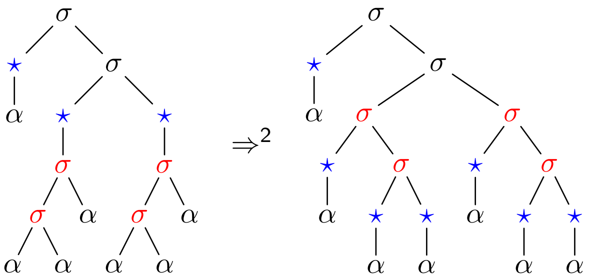
… in action

Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

Derivation:



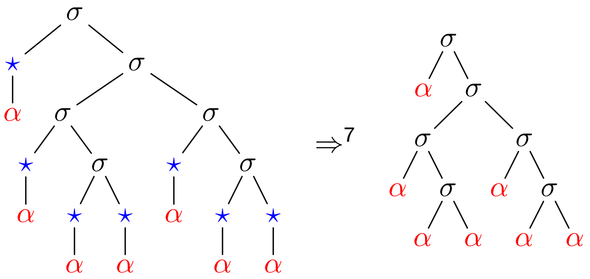
… in action

Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

Derivation:



… in action

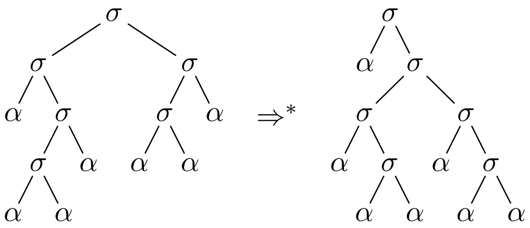
Example

Rules:

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$

$$\star(\alpha) \to \alpha$$

Derivation:



Semantics

Definition

The tree transformation computed by M is $\tau_M \subseteq T_\Sigma \times T_\Delta$

$$\tau_M = \{(t, u) \mid q(t) \Rightarrow^* u \text{ for some initial state } q\}$$

Notation

XTOP = class of transf. computed by extended tree transducers

Syntactic Restrictions

Let $M = (Q, \Sigma, \Delta, S, R)$ be an extended tree transducer.

Definition

M is called linear and nondeleting if for every rule $l \to r$

$$\text{var}(l) = \text{var}(r)$$

and no variable appears more than once in r.

Example

Our example transducer with rules

$$\star(\sigma(\sigma(x_1, x_2), x_3)) \to \sigma(\star(x_1), \sigma(\star(x_2), \star(x_3)))$$
$$\star(\alpha) \to \alpha$$

is linear and nondeleting.

Quest Log

Question

Is the class of transformations computed by linear and nondeleting extended tree transducers closed under composition?

Answer [Knight & Graehl & Hopkins 07]

Quest Log

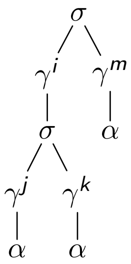
Question

Is the class of transformations computed by linear and nondeleting extended tree transducers closed under composition?

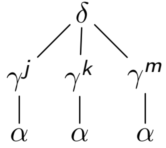
Answer [Knight & Graehl & Hopkins 07]

No!

Transform



into



Two linear and nondeleting extended tree transducers can do that; but a single one cannot.

Quest Log

Open Problems

- Understand linear and nondeleting extended tree transducers better!
- Find subclasses that are closed under composition!
- Identify a suitable superclass that is closed under composition!

Quest Log

Open Problems

- Understand linear and nondeleting extended tree transducers better! (bimorphism)
- Find subclasses that are closed under composition! (unsolved)
- Identify a suitable superclass that is closed under composition! (transformations induced by certain bottom-up devices)

Extended Top-down Tree Transducer

Bimorphism

Multi Bottom-up Tree Transducer

Composition

Bimorphism

Let Σ, Δ, Γ be ranked alphabets.

Definition

A bimorphism is a triple (φ, L, ψ) with

- $\varphi\colon T_\Gamma \to T_\Sigma$ the input homomorphism;
- $L \subseteq T_\Gamma$ the recognizable center;
- $\psi\colon T_\Gamma \to T_\Delta$ the output homomorphism.

Definition

Let $B = (\varphi, L, \psi)$ be a bimorphism. The tree transformation computed by B is

$$\tau_B \subseteq T_\Sigma \times T_\Delta$$
$$\tau_B = \{(\varphi(s), \psi(s)) \mid s \in L\}$$

Equivalently: $\tau_B = \varphi^{-1} \circ \mathrm{id}_L \circ \psi$ (composition of relations)

Illustration

Example

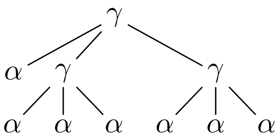
(φ, L, ψ) bimorphism with

- $\Sigma = \Delta = \{\sigma^{(2)}, \alpha^{(0)}\}$ and $\Gamma = \{\gamma^{(3)}, \alpha^{(0)}\}$;
- $L = T_\Gamma$;
- φ and ψ be the homomorphisms such that

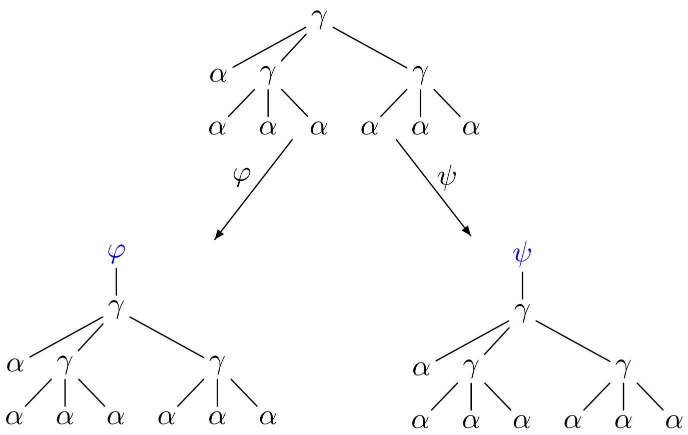
$$\varphi(\gamma) = \sigma(\sigma(x_1, x_2), x_3)$$
$$\psi(\gamma) = \sigma(x_1, \sigma(x_2, x_3))$$

$$\varphi(\alpha) = \alpha$$
$$\psi(\alpha) = \alpha$$

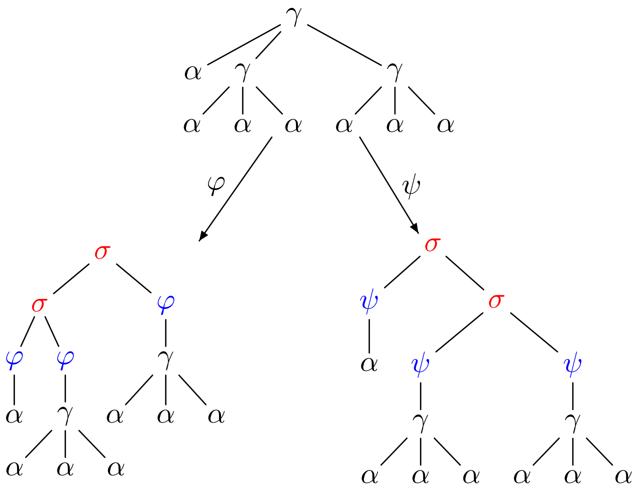
Semantics



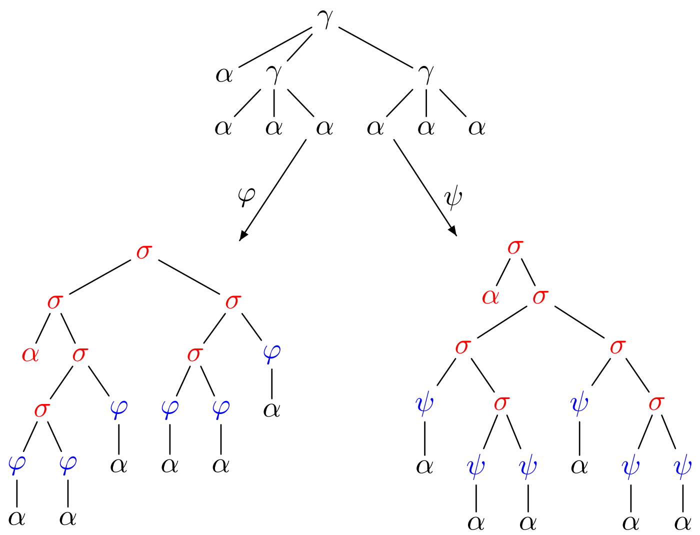
Semantics



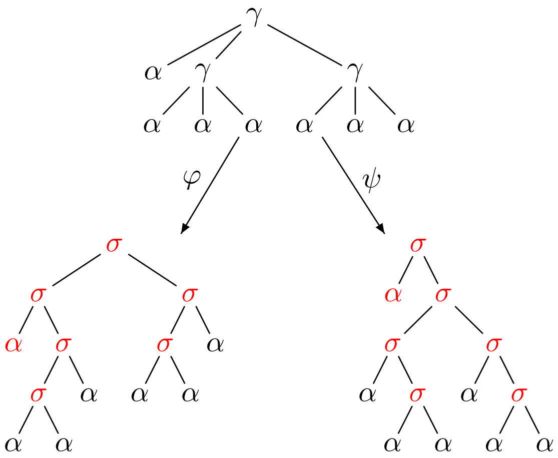
Semantics



Semantics



Semantics



A Relation

Definition

Homomorphism $h\colon T_\Gamma \to T_\Sigma$ is linear and complete if $h(\gamma)$ is linear and nondeleting in X_k for every $k \geq 0$ and $\gamma \in \Gamma^{(k)}$.

Theorem (Knight & Graehl & Hopkins 07, M. 07)

Bimorphisms with linear and complete homomorphisms are as powerful as linear and nondeleting extended tree transducers.

$$\mathrm{BM}(\mathrm{LC}, \mathrm{LC}) = \mathsf{ln}\text{-}\mathrm{XTOP}$$

A Relation

Definition

Homomorphism $h\colon T_\Gamma \to T_\Sigma$ is linear and complete if $h(\gamma)$ is linear and nondeleting in X_k for every $k \geq 0$ and $\gamma \in \Gamma^{(k)}$.

Theorem (Knight & Graehl & Hopkins 07, M. 07)

Bimorphisms with linear and complete homomorphisms are as powerful as linear and nondeleting extended tree transducers.

$$\mathrm{BM}(\mathrm{LC}, \mathrm{LC}) = \text{ln-}\mathrm{XTOP}$$

Theorem (Arnold & Dauchet 82)

Bimorphisms with linear and complete ε-free homomorphisms are not closed under composition.

$$\mathrm{BM}(\mathrm{LCE}, \mathrm{LCE}) \subset \mathrm{BM}(\mathrm{LCE}, \mathrm{LCE})^2 = \mathrm{BM}(\mathrm{LCE}, \mathrm{LCE})^3$$

Quest Log

Achievement

We showed that extended tree transducers consist of three (simple) phases:

- an inverse homomorphism (pattern matcher)
- a recognizable restriction (finite control)
- an output homomorphism (interpretation)

Question

- Which device can implement all phases?
- Is the class of transformations computed by the device closed under composition?

Extended Top-down Tree Transducer

Bimorphism

Multi Bottom-up Tree Transducer

Composition

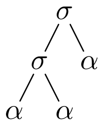
Example Multi Bottom-Up Rules

Rules

Binary state q_σ and unary final state q_α

$$\begin{aligned} \alpha &\to q_\alpha(\alpha) \\ \sigma(q_\alpha(x_1), q_\alpha(x_2)) &\to q_\sigma(x_1, x_2) \\ \sigma(q_\sigma(x_1, x_2), q_\alpha(x_3)) &\to q_\alpha(\sigma(x_1, \sigma(x_2, x_3))) \end{aligned}$$

Illustration



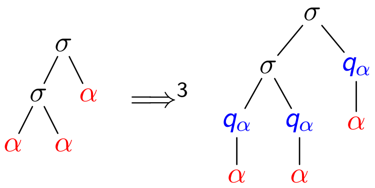
Example Multi Bottom-Up Rules

Rules

Binary state q_σ and unary final state q_α

$$\begin{aligned} \alpha &\to q_\alpha(\alpha) \\ \sigma(q_\alpha(x_1), q_\alpha(x_2)) &\to q_\sigma(x_1, x_2) \\ \sigma(q_\sigma(x_1, x_2), q_\alpha(x_3)) &\to q_\alpha(\sigma(x_1, \sigma(x_2, x_3))) \end{aligned}$$

Illustration



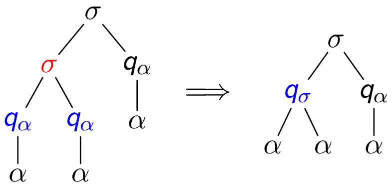
Example Multi Bottom-Up Rules

Rules

Binary state q_σ and unary final state q_α

$$\begin{aligned} \alpha &\to q_\alpha(\alpha) \\ \sigma(q_\alpha(x_1), q_\alpha(x_2)) &\to q_\sigma(x_1, x_2) \\ \sigma(q_\sigma(x_1, x_2), q_\alpha(x_3)) &\to q_\alpha(\sigma(x_1, \sigma(x_2, x_3))) \end{aligned}$$

Illustration



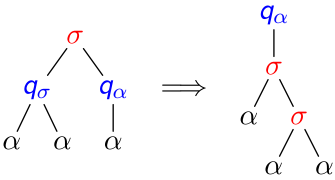
Example Multi Bottom-Up Rules

Rules

Binary state q_σ and unary final state q_α

$$\begin{aligned} \alpha &\to q_\alpha(\alpha) \\ \sigma(q_\alpha(x_1), q_\alpha(x_2)) &\to q_\sigma(x_1, x_2) \\ \sigma(q_\sigma(x_1, x_2), q_\alpha(x_3)) &\to q_\alpha(\sigma(x_1, \sigma(x_2, x_3))) \end{aligned}$$

Illustration



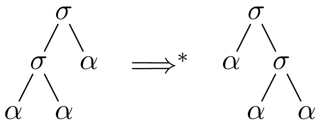
Example Multi Bottom-Up Rules

Rules

Binary state q_σ and unary final state q_α

$$\begin{aligned} \alpha &\to q_\alpha(\alpha) \\ \sigma(q_\alpha(x_1), q_\alpha(x_2)) &\to q_\sigma(x_1, x_2) \\ \sigma(q_\sigma(x_1, x_2), q_\alpha(x_3)) &\to q_\alpha(\sigma(x_1, \sigma(x_2, x_3))) \end{aligned}$$

Illustration



Syntax

Definition (Fülöp & Kühnemann & Vogler 04)

A multi bottom-up tree transducer (mbutt) is a tuple

$$M = (Q, \Sigma, \Delta, F, R)$$

- Q is a ranked alphabet of states
- Σ and Δ are input and output ranked alphabet, respectively
- $F \subseteq Q^{(1)}$ is a set of final states
- R is a finite set of rules of the form

$$\sigma(q_1(x_{1,1}, \ldots, x_{1,n_1}), \ldots, q_k(x_{k,1}, \ldots, x_{k,n_k})) \to q(t_1, \ldots, t_n)$$

with $\sigma \in \Sigma^{(k)}$, $q_1, \ldots, q_k \in Q$, and $t_1, \ldots, t_n \in T_\Delta(X)$.

Semantics

Definition

The tree transformation computed by M is

$$\tau_M \subseteq T_\Sigma \times T_\Delta$$
$$\tau_M = \{(t, u) \mid t \Rightarrow^* q(u) \text{ for some } q \in F\}$$

Definition

MBOT = class of transformations computed by mbutt

Pattern Matching (Phase 1 of 3)

Definition

Let $h\colon T_\Gamma \to T_\Sigma$ be a homomorphism.
h is called ε-free, if $h(\gamma) \notin X$ for every $\gamma \in \Gamma^{(k)}$.

Theorem (M. 07)

The inverse of every ε-free linear and complete homomorphism can be implemented by a linear and nondeleting mbutt

$$\text{lce-}\mathrm{HOM}^{-1} \subseteq \text{ln-}\mathrm{MBOT}$$

Proof sketch.

- recognize pattern occurrences by states
- save processed subtrees in parameters

□

Finite Control (Phase 2 of 3)

Short Recall

The class of recognizable tree languages is the class of languages that are recognized by top-down tree automata (FTA).

Theorem

Every recognizable partial identity can be implemented by a linear and nondeleting mbutt

$$\mathrm{FTA} \subseteq \mathsf{ln}\text{-}\mathrm{BOT} \subseteq \mathsf{ln}\text{-}\mathrm{MBOT}$$

Interpretation (Phase 3 of 3)

Theorem

Every linear and complete homomorphism can be implemented by a linear and nondeleting mbutt

$$\mathsf{lc}\text{-}\mathrm{HOM} \subseteq \mathsf{ln}\text{-}\mathrm{BOT} \subseteq \mathsf{ln}\text{-}\mathrm{MBOT}$$

Quest Log

Corollary

All phases (with one small restriction) can be implemented by linear and nondeleting mbutt

$$\mathsf{lce}\text{-}\mathrm{HOM}^{-1} \cup \mathrm{FTA} \cup \mathsf{lc}\text{-}\mathrm{HOM} \subseteq \mathsf{ln}\text{-}\mathrm{MBOT}$$

Question

Is

$$\mathsf{lce}\text{-}\mathrm{HOM}^{-1} \circ \mathrm{FTA} \circ \mathsf{lc}\text{-}\mathrm{HOM} \subseteq \mathsf{ln}\text{-}\mathrm{MBOT} \ ?$$

Extended Top-down Tree Transducer

Bimorphism

Multi Bottom-up Tree Transducer

Composition

Compositions

Theorem (cf. Kühnemann 06 for deterministic mbutt)

The class of transformations computed by linear and nondeleting mbutt is closed under composition

$$\mathsf{ln}\text{-}\mathrm{MBOT}^2 = \mathsf{ln}\text{-}\mathrm{MBOT}$$

Corollary

Linear and nondeleting mbutt are at least as powerful as bimorphisms with linear and complete homomorphisms and an ε-free input homomorphism.

$$\mathrm{BM}(\mathrm{LCE}, \mathrm{LC}) \subseteq \mathsf{ln}\text{-}\mathrm{MBOT}$$

Are We Too Powerful?

Question

Are linear and nondeleting mbutt too powerful?

Answer

No! (see Theorem)

Theorem

Every linear and nondeleting mbutt can be simulated by a composition of a stateful relabeling and a deterministic top-down tree transducer

$$\text{ln-}\mathrm{MBOT} \subseteq \mathrm{QREL} \circ \text{d-}\mathrm{TOP}$$

References

André Arnold and Max Dauchet.
Morphismes et bimorphismes d'arbres.
Theor. Comput. Sci., 20:33–93, 1982.

Z. Fülöp, A. Kühnemann, and H. Vogler.
A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead.
Inform. Proc. Letters, 91:57–67, 2004.

Jonathan Graehl and Kevin Knight.
Training tree transducers.
In *Proc. HLT/NAACL*, pages 105–112. Association for Computational Linguists, 2004.

Kevin Knight and Jonathan Graehl.
An overview of probabilistic tree transducers for natural language processing.
In *Proc. 6th Int. Conf. Comput. Linguistics and Intel. Text Proc.*, volume 3406 of *LNCS*, pages 1–24. Springer, 2005.

Kevin Knight, Jonathan Graehl, and Mark Hopkins.
Extended top-down tree transducers.
Manuscript, 2007.

Armin Kühnemann.
Composition of deterministic multi bottom-up tree transducers.
Manuscript, 2006.

Stuart M. Shieber.
Synchronous grammars as tree transducers.
In *Proc. 7th Int. Workshop Tree Adjoining Grammars and Related Formalisms*, pages 88–95, 2004.