Compositions of Top-down Tree Transducers with ε-Rules

Andreas Maletti[1] and Heiko Vogler[2]

[1] URV, Tarragona, Spain
[2] Technische Universität Dresden, Germany

andreas.maletti@urv.cat

Pretoria — July 24, 2009

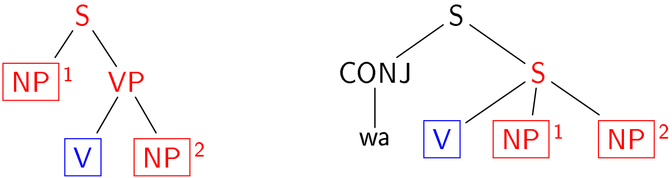
First author acknowledges the support of MEC grants JDCI-2007-760 and MTM-2007-63422.

Synchronous Tree Substitution Grammars

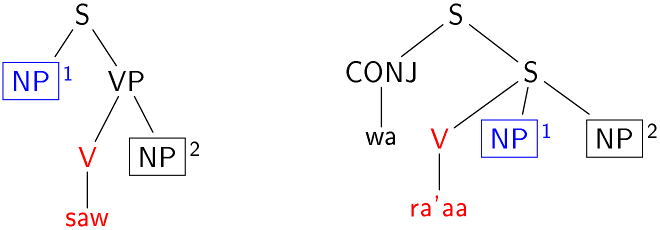
$\boxed{S}$ $\boxed{S}$

Synchronous Tree Substitution Grammars

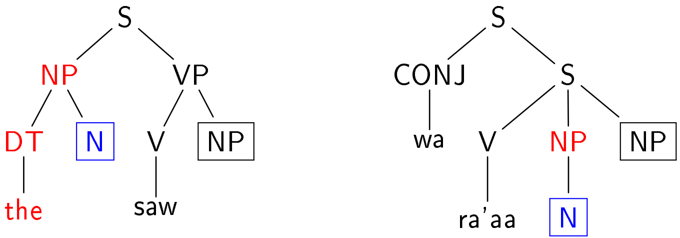
Synchronous Tree Substitution Grammars



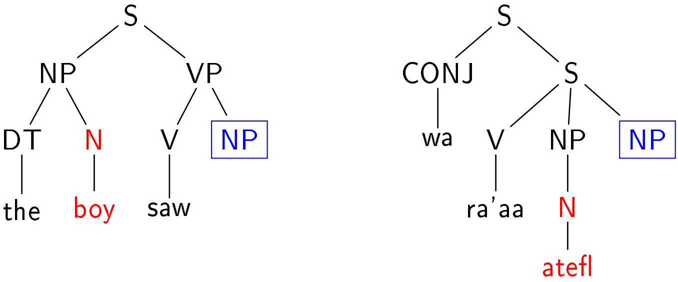
Synchronous Tree Substitution Grammars



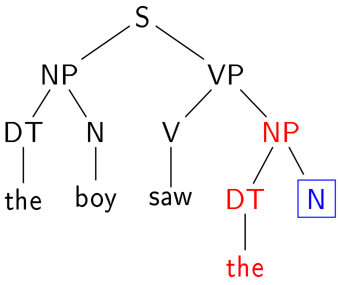
Synchronous Tree Substitution Grammars



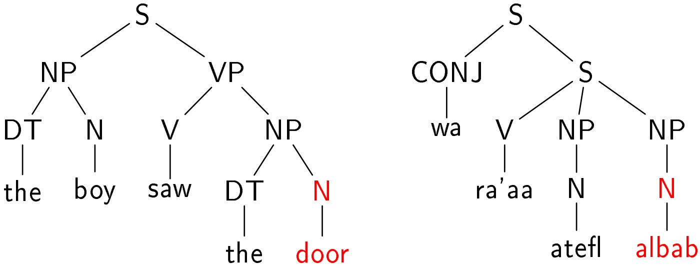
Synchronous Tree Substitution Grammars



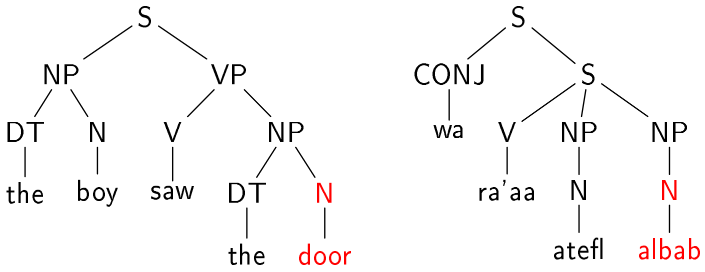
Synchronous Tree Substitution Grammars



Synchronous Tree Substitution Grammars



Synchronous Tree Substitution Grammars

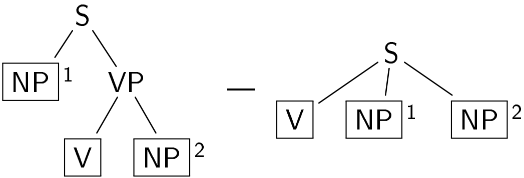


Implementation

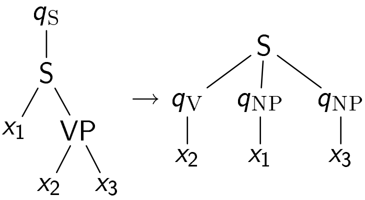
- input-oriented with hidden states ($\longrightarrow$ tree transducer)
- linear nondeleting extended top-down tree transducer in Tiburon [May, Knight '06]

Synchronous Tree Substitution Grammars (cont'd)

Synchronous tree substitution grammar rule:

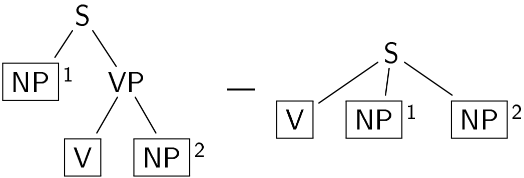


Tree transducer rule:

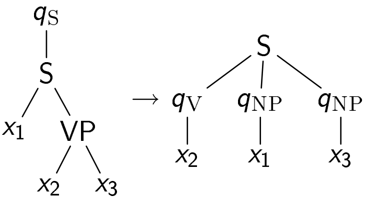


Synchronous Tree Substitution Grammars (cont'd)

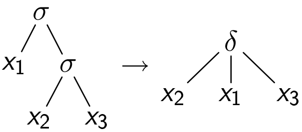
Synchronous tree substitution grammar rule:



Tree transducer rule:

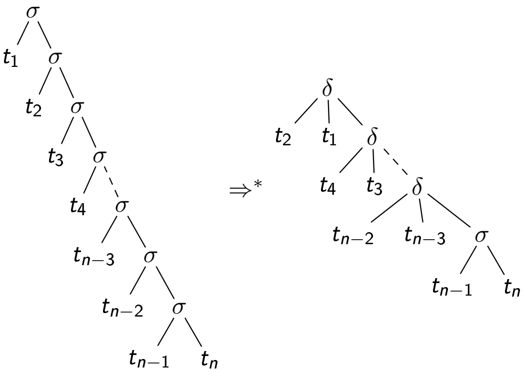


Simplified transducer rule:



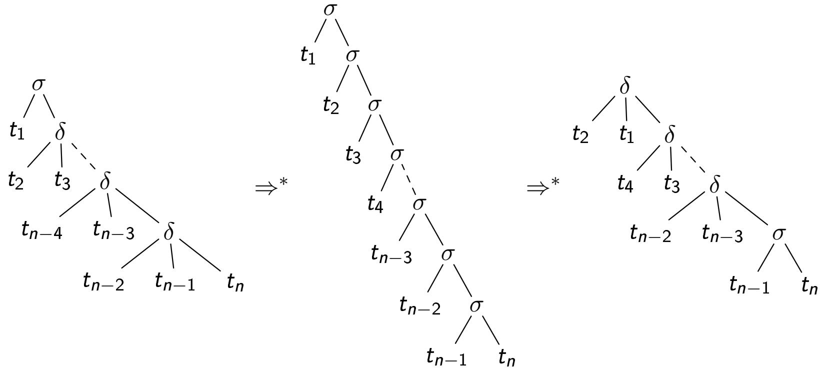
Synchronous Tree Substitution Grammars (cont'd)

Derivation:



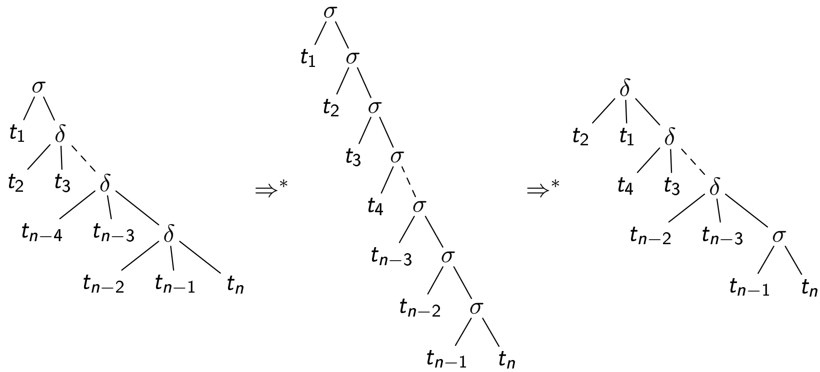
Synchronous Tree Substitution Grammars (cont'd)

Derivation:



Synchronous Tree Substitution Grammars (cont'd)

Derivation:



Conclusion

We cannot compose them!

Synchronous Tree Substitution Grammars (cont'd)

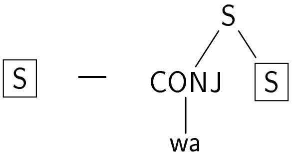
Why?

- linear nondeleting top-down tree transducers can be composed [Engelfriet '75, Baker '79]

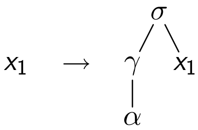
$\longrightarrow$ large left-hand sides cause problems

$\longrightarrow$ What about ε-rules?

STSG rule:



Simplified tree transducer rule:



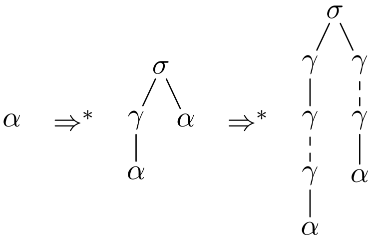
Synchronous Tree Substitution Grammars (cont'd)

Derivation:

$$\alpha \Rightarrow^* \sigma(\gamma(\alpha), \alpha)$$

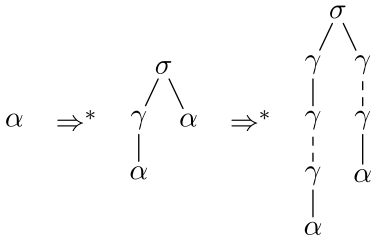
Synchronous Tree Substitution Grammars (cont'd)

Derivation:



Synchronous Tree Substitution Grammars (cont'd)

Derivation:



Conclusion

We cannot even compose linear nondeleting top-down tree transducers with ε-rules!

Table of Contents

Syntax

Definition (cf. [Rounds '70] & [Thatcher '70])

Top-down tree transducer (εtdtt) with ε-rules $(Q, \Sigma, \Delta, I, R)$

- Q alphabet of *states*
- Σ and Δ ranked alphabets of input and output symbols
- $I \subseteq Q$ *initial states*
- R finite set of *(rewrite) rules* $q(l) \to r$ with
  - (i) $q \in Q$,
  - (ii) $l = x_1$ or $l = \sigma(x_1, \ldots, x_k)$ for some $\sigma \in \Sigma_k$, and
  - (iii) $r \in T_\Delta(Q(\text{var}(l)))$

A full example

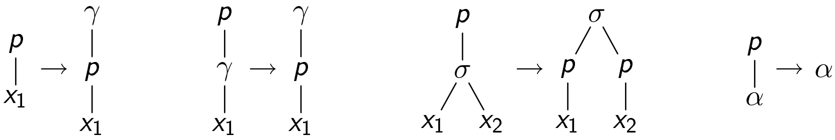
Example

εtdtt $M = (P, \Sigma, \Sigma, P, R)$ with

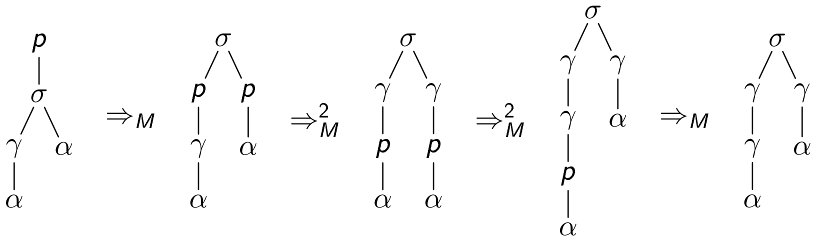
- $P = \{p\}$
- $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$
- the following rules in R

$$p(x_1) \to \gamma(p(x_1)) \qquad p(\gamma(x_1)) \to \gamma(p(x_1)) \qquad p(\sigma(x_1, x_2)) \to \sigma(p(x_1), p(x_2)) \qquad p(\alpha) \to \alpha$$

A full example (cont'd)



Example



Semantics

Definition

$M = (Q, \Sigma, \Delta, I, R)$ εtdtt

$$\tau_M = \{(t, u) \mid \exists q \in I \colon q(t) \Rightarrow_M^* u\}$$

Semantics

Definition

$M = (Q, \Sigma, \Delta, I, R)$ εtdtt

$$\tau_M = \{(t, u) \mid \exists q \in I \colon q(t) \Rightarrow^*_M u\}$$

Example

Our example εtdtt can include γ-symbols anywhere in the input tree.

Syntactical restrictions

Definition

εtdtt $M = (Q, \Sigma, \Delta, I, R)$ is

- linear if every right-hand side (of a rule) does not contain duplicate variables
- nondeleting, if every right-hand side contains all variables of its left-hand side
- total if for every $q \in Q$ and $t \in T_\Sigma$ there exists $u \in T_\Delta$ such that $q(t) \Rightarrow^*_M u$

Syntactical restrictions

Definition

εtdtt $M = (Q, \Sigma, \Delta, I, R)$ is

- linear if every right-hand side (of a rule) does not contain duplicate variables
- nondeleting, if every right-hand side contains all variables of its left-hand side
- total if for every $q \in Q$ and $t \in T_\Sigma$ there exists $u \in T_\Delta$ such that $q(t) \Rightarrow^*_M u$

Example

Our example εtdtt is linear, nondeleting, and total.

Table of Contents

One-symbol form

Definition (cf. [Berstel '79] & [Engelfriet et al '08])

An εtdtt is in one-symbol form if at most one output symbol occurs in each rule.

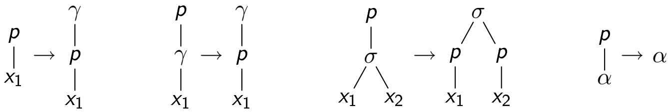
One-symbol form

Definition (cf. [Berstel '79] & [Engelfriet et al '08])

An εtdtt is in one-symbol form if at most one output symbol occurs in each rule.

Example

Our example εtdtt is in one-symbol form.



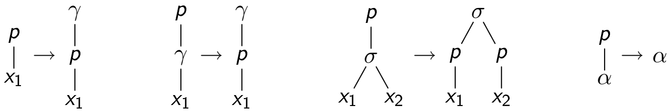
One-symbol form

Definition (cf. [Berstel '79] & [Engelfriet et al '08])

An εtdtt is in one-symbol form if at most one output symbol occurs in each rule.

Example

Our example εtdtt is in one-symbol form.



Theorem

For every εtdtt we can construct an equivalent one in one-symbol form.

Maximally output-separated

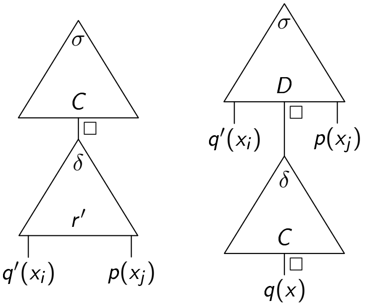
Definition

An εtdtt is maximally output-separated if for every rule $q(l) \to r \in R$

(i) $r \neq C[r']$ for every nontrivial context $C \in C_\Delta$ and $r' \in \Delta(T_\Delta(Q(X)))$,

(ii) $r \neq D[C[q(x)]]$ for every nontrivial context $D \in C_\Delta(Q(X))$ and nontrivial context $C \in C_\Delta$.

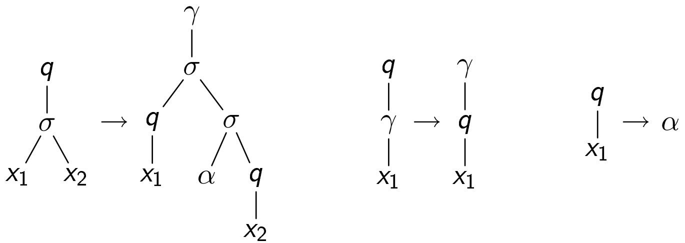
Illustration



(i) (ii)

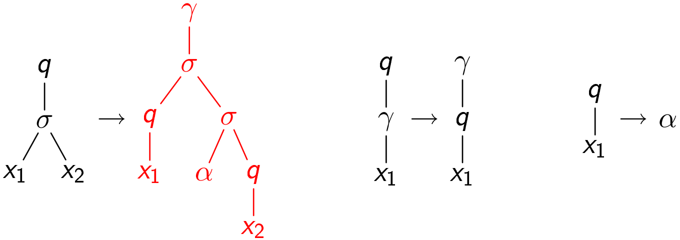
Maximally output-separated (cont'd)

Example



Maximally output-separated (cont'd)

Example



is not maximally output-separated because

(i) $r = C[r']$ with $C = \gamma(\Box)$ and $r' = \sigma(q(x_1), \sigma(\alpha, q(x_2)))$

(ii) $r = D[C[q(x_2)]]$ with $D = \gamma(\sigma(q(x_1), \Box))$ and $C = \sigma(\alpha, \Box)$

Maximally output-separated (cont'd)

Theorem

For every (linear, nondeleting) εtdtt there exists an equivalent maximally output-separated εtdtt with the same properties.

Proof.

Separate the context C from a rule creating two rules, which need to be applied one after the other by introducing a new state. $\square$

Maximally output-separated (cont'd)

Theorem

For every (linear, nondeleting) εtdtt there exists an equivalent maximally output-separated εtdtt with the same properties.

Proof.

Separate the context C from a rule creating two rules, which need to be applied one after the other by introducing a new state. $\square$

Example

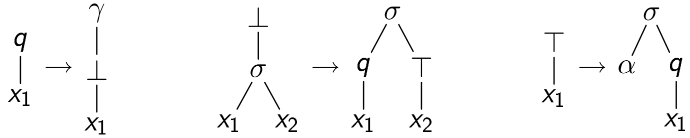


Table of Contents

Composition construction

Definition

εtdtt $M = (P, \Sigma, \Gamma, I_1, R_1)$ and $N = (Q, \Gamma, \Delta, I_2, R_2)$ construct

$$M ; N = (Q \times P, \Sigma, \Delta, I_2 \times I_1, R_1' \cup R_2' \cup R')$$

with 3 types of rules:

1. rules R_1' constructed from rules of R_1 that do not produce output

$$R_1' = \{q(l) \to q(r) \mid q \in Q, l \to r \in R_1, r \in P(X)\}$$

Composition construction

Definition

εtdtt $M = (P, \Sigma, \Gamma, I_1, R_1)$ and $N = (Q, \Gamma, \Delta, I_2, R_2)$ construct

$$M \mathbin{;} N = (Q \times P, \Sigma, \Delta, I_2 \times I_1, R_1' \cup R_2' \cup R')$$

with 3 types of rules:

1. rules R_1' constructed from rules of R_1 that do not produce output
2. epsilon rules R_2' constructed from epsilon-rules of R_2

$$R_2' = \{l[p(x_1)] \to r[p(x_1)] \mid p \in P, l \to r \in R_2, l \in Q(X)\}$$

Composition construction

Definition

εtdtt $M = (P, \Sigma, \Gamma, I_1, R_1)$ and $N = (Q, \Gamma, \Delta, I_2, R_2)$ construct

$$M \,;\, N = (Q \times P, \Sigma, \Delta, I_2 \times I_1, R_1' \cup R_2' \cup R')$$

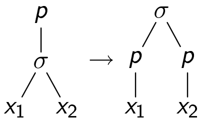
with 3 types of rules:

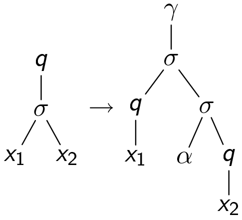
1. rules R_1' constructed from rules of R_1 that do not produce output
2. epsilon rules R_2' constructed from epsilon-rules of R_2
3. rules R' constructed from rules of R_1 that contain an output symbol that is immediately consumed by a rule of R_2

$$R' = \{q(l) \to r' \mid q \in Q, l \to r \in R_1, r \in \Gamma(P(X)), q(r) \Rightarrow_N r'\}$$

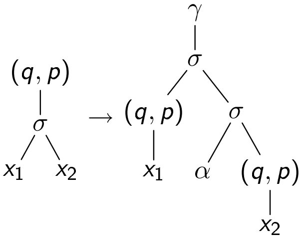
Composition construction (cont'd)

Example





Resulting rule of R':



Composition

Theorem (cf. [Engelfriet '75] & [Baker '79])

Let M and N be εtdtts. Then $M\,;N$ computes $\tau_M\,;\tau_N$ if

- *N is linear,*
- *M is total or N is nondeleting, and*
- *M is in one-symbol form.*

Composition

Theorem (cf. [Engelfriet '75] & [Baker '79])

Let M and N be εtdtts. Then $M \mathbin{;} N$ computes $\tau_M \mathbin{;} \tau_N$ if

- *N is linear,*
- *M is total or N is nondeleting, and*
- *M is in one-symbol form.*

Consequences

- The class of transformations computed by linear nondeleting εtdtt in one-symbol form is closed under composition.

Composition

Theorem (cf. [Engelfriet '75] & [Baker '79])

Let M and N be εtdtts. Then $M \,;\, N$ computes $\tau_M \,;\, \tau_N$ if

- *N is linear,*
- *M is total or N is nondeleting, and*
- *M is in one-symbol form.*

Consequences

- The class of transformations computed by linear nondeleting εtdtt in one-symbol form is closed under composition.
- The class of transformations computed by synchronous context-free grammars (incl. chain rules) is closed under composition.

Composition (cont'd)

Theorem

Any composition of linear nondeleting εtdtts can be simulated by an εtdtt.

$$\text{ln-eTOP} ; \cdots ; \text{ln-eTOP} \subseteq \text{eTOP}$$

Proof.

1. Take first εtdtt into one-symbol form (losing linearity and nondeletion)
2. Compose with next linear and nondeleting εtdtt
3. Result computes composition of first 2 εtdtts
4. Repeat □

References

- Baker: Composition of top-down and bottom-up tree transformations. *Inform. Control* 41(2), 1979
- Berstel: Transductions and context-free languages. Teubner 1979
- Engelfriet: Bottom-up and top-down tree transformations — a comparison. *Math. Systems Theory* 9(3), 1975
- Engelfriet, Lilin, Maletti: Extended multi bottom-up tree transducers. Proc. DLT. LNCS 5257, 2008
- Rounds: Mappings and grammars on trees. *Math. Systems Theory* 4(3), 1970
- Thatcher: Generalized^2 sequential machine maps. *J. Comput. System Sci.* 4(4), 1970

Thank you for your attention!