Unidirectional Derivation Semantics for Synchronous Tree-Adjoining Grammars

Matthias Büchse[1] and Andreas Maletti[2] and Heiko Vogler[1]

[1] Faculty of Computer Science
Technische Universität Dresden
01062 Dresden, Germany

[2] Institute for Natural Language Processing
Universität Stuttgart
70569 Stuttgart, Germany

maletti@ims.uni-stuttgart.de

Taipei — August 17, 2012

Tree-Adjoining Grammars

Motivation [JOSHI]

- mildly context-sensitive formalism
- **local** dependencies in rules

Tree-Adjoining Grammars

Motivation [JOSHI]

- mildly context-sensitive formalism
- **local** dependencies in rules
- but **global** dependencies in derivation

Tree-Adjoining Grammars

Motivation [Joshi]

- mildly context-sensitive formalism
- **local** dependencies in rules
- but **global** dependencies in derivation

Applications

- TAG for English [XTAG Group 2001]
- TAG for German [Kallmeyer et al. 2010]

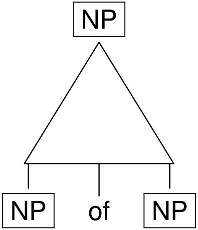
Tree-Adjoining Grammars

Definition (Joshi et al. 1969)

Tree-adjoining grammar (TAG) has a finite set of

- **substitution rules**
- adjunction rules

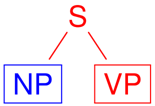
Substitution rule (rules of a regular tree grammar):



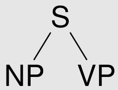
Tree-Adjoining Grammars

S

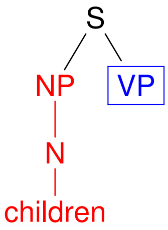
Tree-Adjoining Grammars



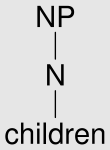
Used substitution rule



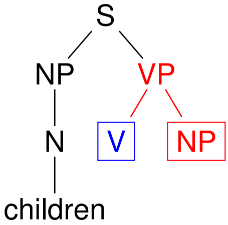
Tree-Adjoining Grammars



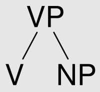
Used substitution rule



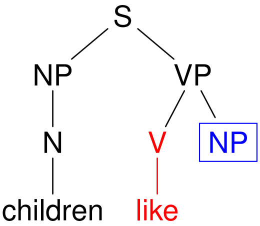
Tree-Adjoining Grammars



Used substitution rule

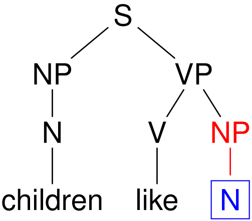


Tree-Adjoining Grammars



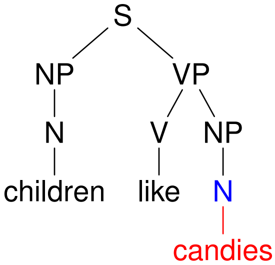
Used substitution rule

Tree-Adjoining Grammars

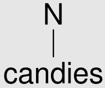


Used substitution rule

Tree-Adjoining Grammars



Used substitution rule



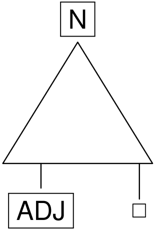
Tree-Adjoining Grammars

Definition (JOSHI et al. 1969)

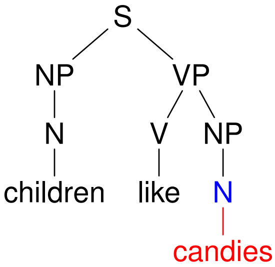
Tree-adjoining grammar (TAG) has a finite set of

- substitution rules
- **adjunction rules**

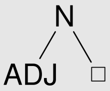
Adjunction rule:



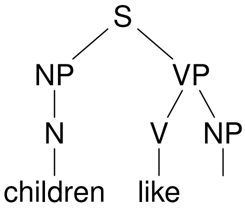
Tree-Adjoining Grammars



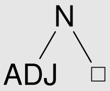
Used adjunction rule

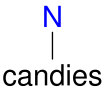


Tree-Adjoining Grammars

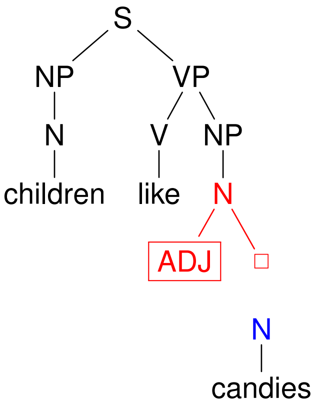


Used adjunction rule

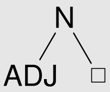




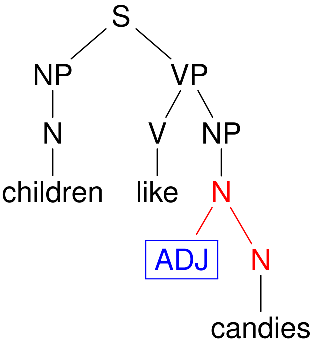
Tree-Adjoining Grammars



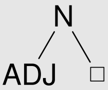
Used adjunction rule



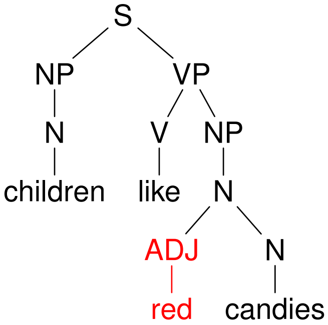
Tree-Adjoining Grammars



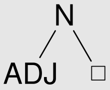
Used adjunction rule



Tree-Adjoining Grammars



Used adjunction rule



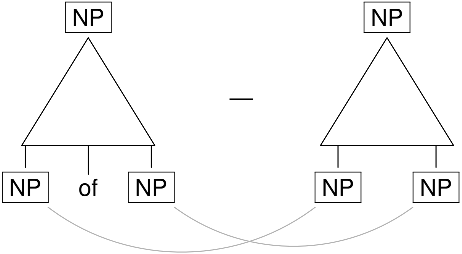
Used substitution rule

Synchronous Tree-Adjoining Grammars

Definition (SHIEBER and SCHABES 1990)

Synchronous tree-adjoining grammar (STAG) consists of two synchronized TAG

Substitution rule:

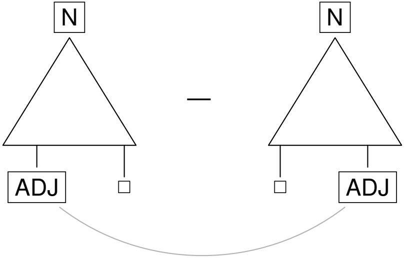


Synchronous Tree-Adjoining Grammars

Definition (SHIEBER and SCHABES 1990)

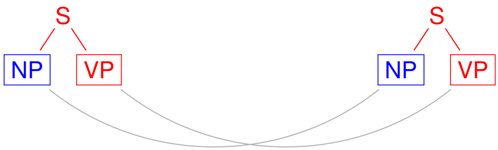
Synchronous tree-adjoining grammar (STAG) consists of two synchronized TAG

Adjunction rule:



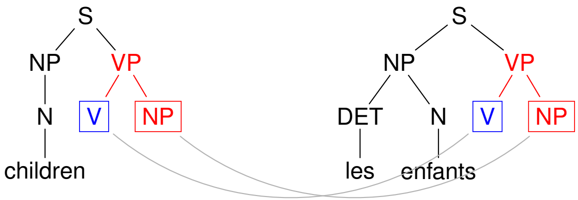
Synchronous Tree-Adjoining Grammars

Synchronous Tree-Adjoining Grammars

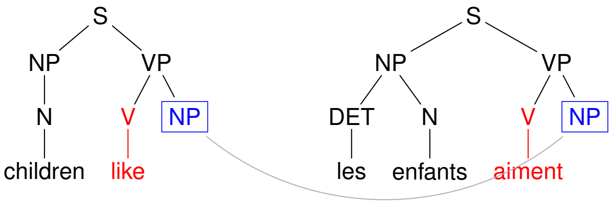


Synchronous Tree-Adjoining Grammars

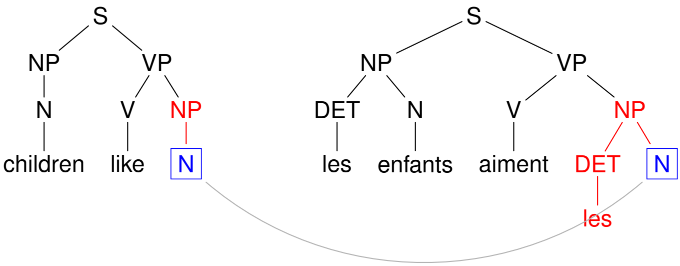
Synchronous Tree-Adjoining Grammars



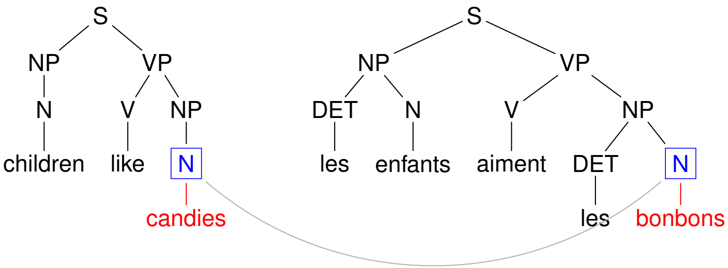
Synchronous Tree-Adjoining Grammars



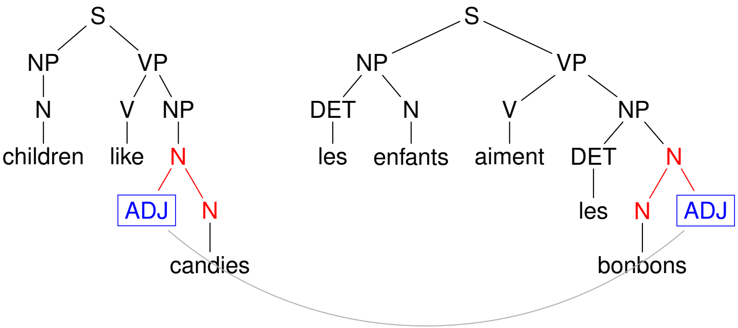
Synchronous Tree-Adjoining Grammars



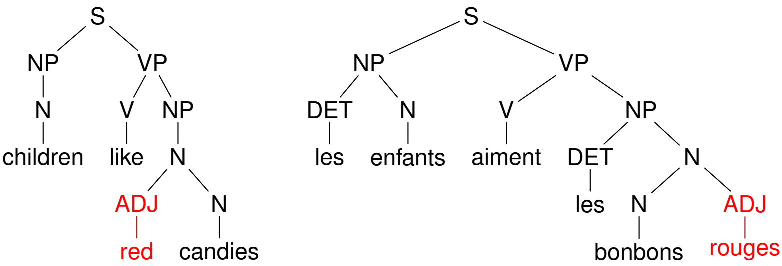
Synchronous Tree-Adjoining Grammars



Synchronous Tree-Adjoining Grammars



Synchronous Tree-Adjoining Grammars

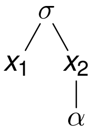


Contents

Tree Substitution

Types

- **first-order**: $t(\!|u|\!)^0_v$ replaces leaf at v in t by u
- **second-order**: $t(\!|u|\!)^1_v$ replaces unary node at v in t by u
  (with the subtree at $v1$ substituted into u)

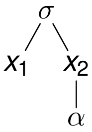


t

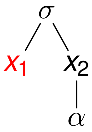
Tree Substitution

Types

- **first-order**: $t⦇u⦈^0_v$ replaces leaf at v in t by u
- **second-order**: $t⦇u⦈^1_v$ replaces unary node at v in t by u (with the subtree at $v1$ substituted into u)



t



$t⦇\alpha⦈^0_1$

Tree Substitution

Types

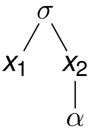
- **first-order**: $t\llparenthesis u \rrparenthesis_v^0$ replaces leaf at v in t by u
- **second-order**: $t\llparenthesis u \rrparenthesis_v^1$ replaces unary node at v in t by u (with the subtree at $v1$ substituted into u)

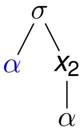
t $\qquad$ $t\llparenthesis \alpha \rrparenthesis_1^0$

Tree Substitution

Types

- **first-order**: $t⦇u⦈_v^0$ replaces leaf at v in t by u
- **second-order**: $t⦇u⦈_v^1$ replaces unary node at v in t by u
  (with the subtree at $v1$ substituted into u)





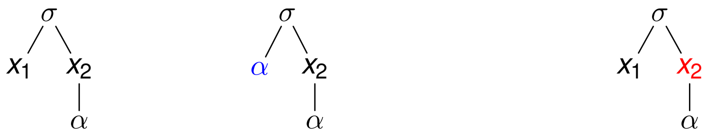
t

$t⦇\alpha⦈_1^0 = t⦇x_1/\alpha⦈^0$

Tree Substitution

Types

- **first-order**: $t⦇u⦈^0_v$ replaces leaf at v in t by u
- **second-order**: $t⦇u⦈^1_v$ replaces unary node at v in t by u
  (with the subtree at $v1$ substituted into u)

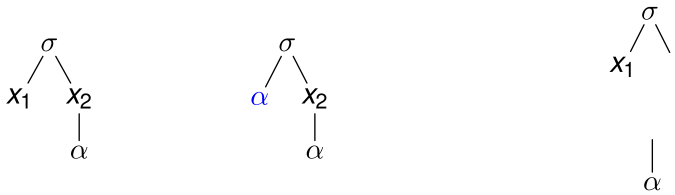


t $\qquad$ $t⦇\alpha⦈^0_1 = t⦇x_1/\alpha⦈^0$ $\qquad$ $t⦇\gamma(\square, \beta)⦈^1_2$

Tree Substitution

Types

- **first-order**: $t⦇u⦈^0_v$ replaces leaf at v in t by u
- **second-order**: $t⦇u⦈^1_v$ replaces unary node at v in t by u
  (with the subtree at $v1$ substituted into u)

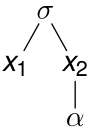


t $\qquad$ $t⦇\alpha⦈^0_1 = t⦇x_1/\alpha⦈^0$ $\qquad$ $t⦇\gamma(\square, \beta)⦈^1_2$

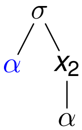
Tree Substitution

Types

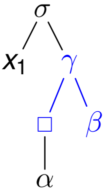
- **first-order**: $t(\!|u|\!)^0_v$ replaces leaf at v in t by u
- **second-order**: $t(\!|u|\!)^1_v$ replaces unary node at v in t by u
  (with the subtree at $v1$ substituted into u)



t



$t(\!|\alpha|\!)^0_1 = t(\!|x_1/\alpha|\!)^0$

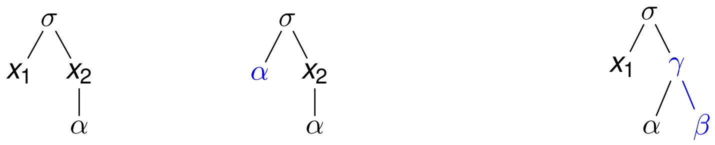


$t(\!|\gamma(\square, \beta)|\!)^1_2$

Tree Substitution

Types

- **first-order**: $t(\!|u|\!)_v^0$ replaces leaf at v in t by u
- **second-order**: $t(\!|u|\!)_v^1$ replaces unary node at v in t by u (with the subtree at $v1$ substituted into u)

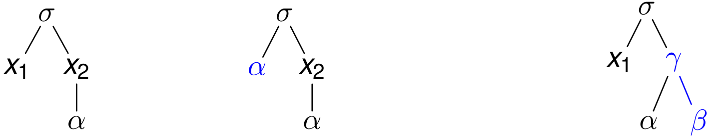


t $\qquad$ $t(\!|\alpha|\!)_1^0 = t(\!|x_1/\alpha|\!)^0$ $\qquad$ $t(\!|\gamma(\Box, \beta)|\!)_2^1$

Tree Substitution

Types

- **first-order**: $t⦇u⦈^0_v$ replaces leaf at v in t by u
- **second-order**: $t⦇u⦈^1_v$ replaces unary node at v in t by u (with the subtree at $v1$ substituted into u)



t

$t⦇\alpha⦈^0_1 = t⦇x_1/\alpha⦈^0$

$t⦇\gamma(\square, \beta)⦈^1_2 = t⦇x_2/\gamma(\square, \beta)⦈^1$

Monadic Doubly Ranked Alphabet

Definition

Alphabet Q with a mapping $\text{rk}\colon Q \to \{0, 1\}^2$

Monadic Doubly Ranked Alphabet

Definition

Alphabet Q with a mapping $\mathrm{rk}\colon Q \to \{0, 1\}^2$

Notes

- input and output rank rk_1 and rk_2
- rank 0 $\to$ first-order substitution $t\llparenthesis \cdots \rrparenthesis^0$
- rank 1 $\to$ second-order substitution $t\llparenthesis \cdots \rrparenthesis^1$

Monadic Doubly Ranked Alphabet

Definition

Alphabet Q with a mapping $\mathrm{rk} \colon Q \to \{0, 1\}^2$

Notes

- input and output rank rk_1 and rk_2
- rank 0 $\rightarrow$ first-order substitution $t(\!|\cdots|\!)^0$
- rank 1 $\rightarrow$ second-order substitution $t(\!|\cdots|\!)^1$
- $Q^{(i,j)} = \{q \in Q \mid \mathrm{rk}(q) = (i, j)\}$

Synchronous Tree-Adjoining Grammar

Definition (Büchse, Nederhof, Vogler 2011)

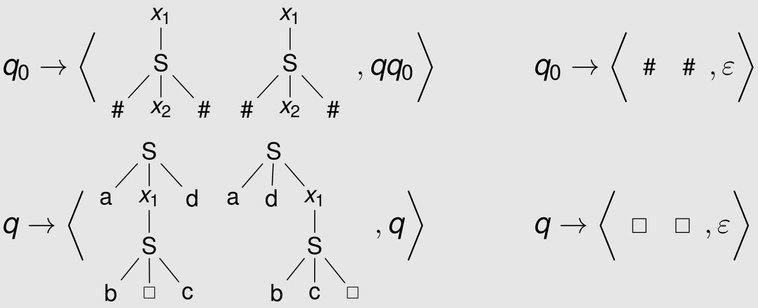
(Q, Σ, q_0, R) **synchronous tree-adjoining grammar** (STAG) if

- Q monadic doubly-ranked alphabet — *states*
- Σ alphabet — *terminals*
- $q_0 \in Q^{(0,0)}$ — *initial state*
- R finite set of elements of the form $q \to \langle \zeta\zeta', q_1 \cdots q_m \rangle$ — *rules*
  - ζ, ζ' trees over $\Sigma \cup \{x_1, \ldots, x_m\} \cup \{\square\}$
  - $\square$ occurs according to rank of q in (ζ, ζ')
  - x_j occurs exactly once in ζ and ζ'
  - rank of x_j in (ζ, ζ') equals rank of q_j

Synchronous Tree-Adjoining Grammar

Example

(Q, Σ, q_0, R) with $q_0 \in Q^{(0,0)}$ and $q \in Q^{(1,1)}$



Unidirectional Derivation Semantics

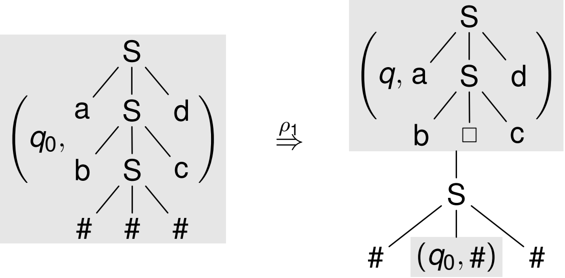
Definition

$\xi_1 \overset{\rho}{\Rightarrow} \xi_2$ with $\rho = q \to \langle \zeta\zeta', q_1 \cdots q_m \rangle$ if there are
(i) minimal redex position v in ξ_1 and (ii) trees $t_1, \ldots, t_m$

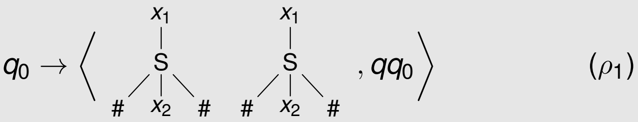
1. $\square$ occurs in t_j according to $\mathrm{rk}_1(q_j)$
2. $\xi_1(v) = (q, \zeta\theta_1 \cdots \theta_m)$ with $\theta_j = (\!| x_j / t_j |\!)^{\mathrm{rk}_1(q_j)}$
3. $\xi_2 = \xi_1 (\!| \zeta'\theta'_1 \cdots \theta'_m |\!)_v^{\mathrm{rk}_2(q)}$ with

$$\theta'_j = \begin{cases} (\!| x_j / (q_j, t_j) |\!)^0 & \text{if } \mathrm{rk}_2(q_j) = 0 \\ (\!| x_j / (q_j, t_j)(\square) |\!)^1 & \text{otherwise} \end{cases}$$

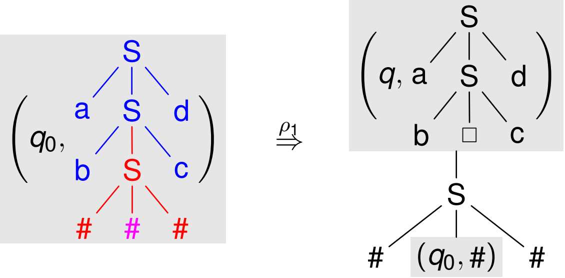
Unidirectional Derivation Semantics



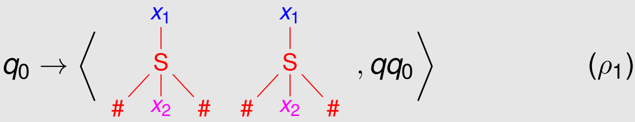
Example (Used rule)



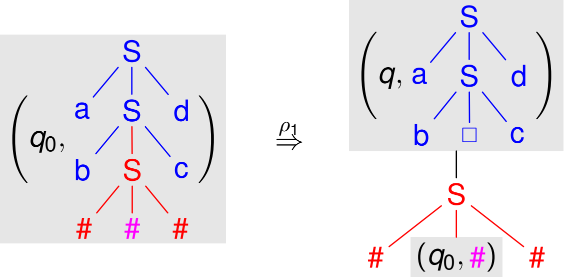
Unidirectional Derivation Semantics



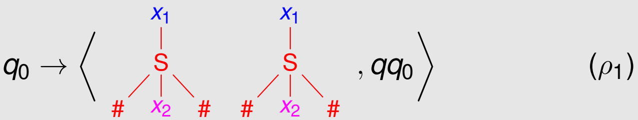
Example (Used rule)



Unidirectional Derivation Semantics



Example (Used rule)



Unidirectional Derivation Semantics

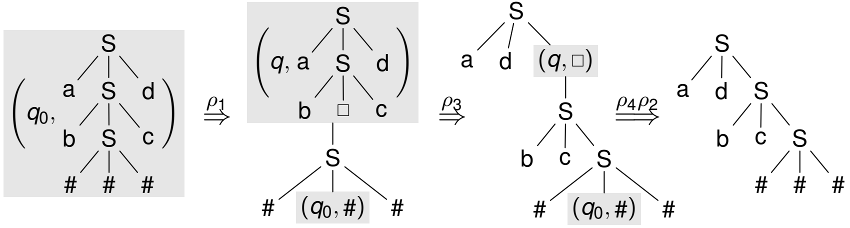
$\overset{d}{\Rightarrow} = \overset{\rho_1}{\Rightarrow} ; \cdots ; \overset{\rho_n}{\Rightarrow}$ with $d = \rho_1 \cdots \rho_n$

Definition

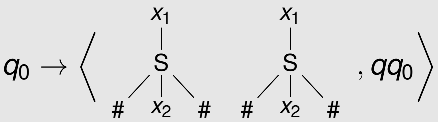
STAG G *derivation-induces*

$$\kappa_G = \{(s, t) \mid \exists d \in R^* : (q_0, s) \overset{d}{\Rightarrow} t\}$$

Unidirectional Derivation Semantics

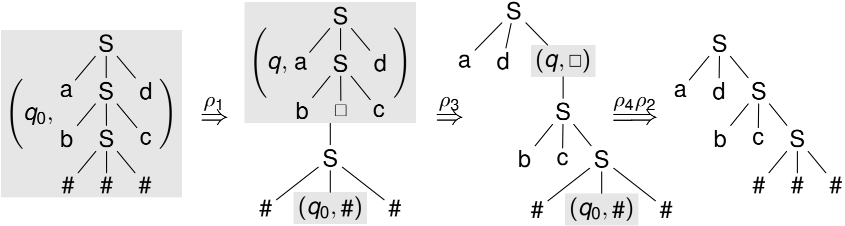


Example (Rules)

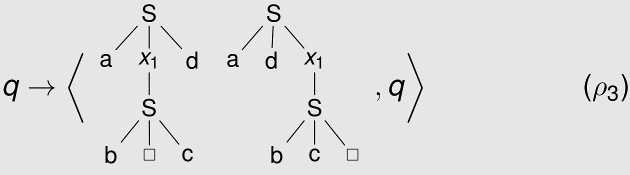


(ρ_1)

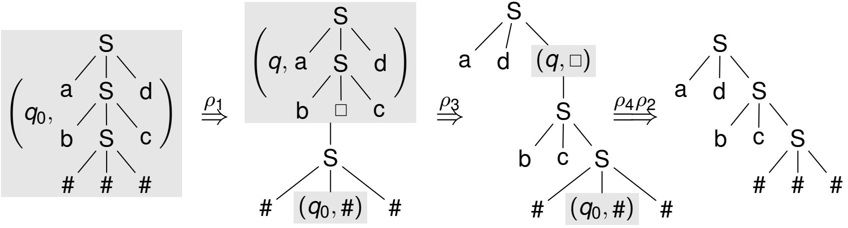
Unidirectional Derivation Semantics



Example (Rules)



Unidirectional Derivation Semantics



Example (Rules)

$$q_0 \to \left\langle \# \quad \# \;, \varepsilon \right\rangle \qquad (\rho_4)$$

$$q \to \left\langle \square \quad \square \;, \varepsilon \right\rangle \qquad (\rho_2)$$

Contents

Extended Top-down Tree Transducer

Definition

STAG (Q, Σ, q_0, R) is a (linear and nondeleting)
extended top-down tree transducer (XTOP) if $Q = Q^{(0,0)}$

Explicit Substitution

$\underline{\Sigma} = \Sigma \cup \{\cdot[\cdot], \bigcirc\}$ where

- $\cdot[\cdot]$ binary substitution symbol
- $\bigcirc$ nullary substitution site symbol

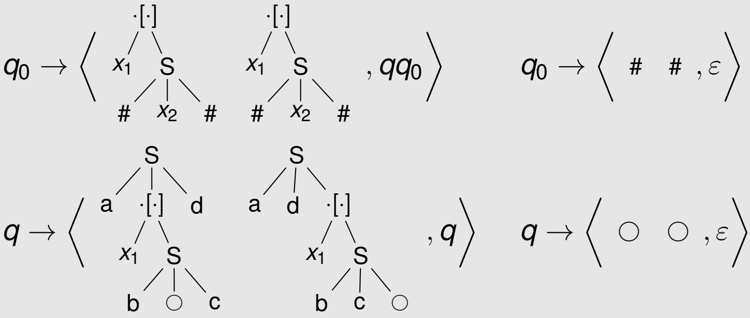
Definition

Evaluation $\cdot^{\mathrm{E}} \colon T_{\underline{\Sigma}} \to T_{\Sigma \cup \{\Box\}}$

- $\bigcirc^{\mathrm{E}} = \Box$
- $\sigma(t_1, \dots, t_k)^{\mathrm{E}} = \sigma(t_1^{\mathrm{E}}, \dots, t_k^{\mathrm{E}})$
- $\cdot[\cdot](t, u)^{\mathrm{E}} = t^{\mathrm{E}}[\Box \leftarrow u^{\mathrm{E}}]$

XTOP using Explicit Substitution

Example



Definition

Tree $t \in T_\Sigma$ is **well-behaved** (under $\cdot^{\mathrm{E}}$) if

- $t^{\mathrm{E}} \in T_\Sigma$
- $t_1^{\mathrm{E}} \in C_\Sigma$ for every subtree of the form $\cdot[\cdot](t_1, t_2)$ in t

Definition

Tree $t \in T_\Sigma$ is **well-behaved** (under $\cdot^{\mathrm{E}}$) if

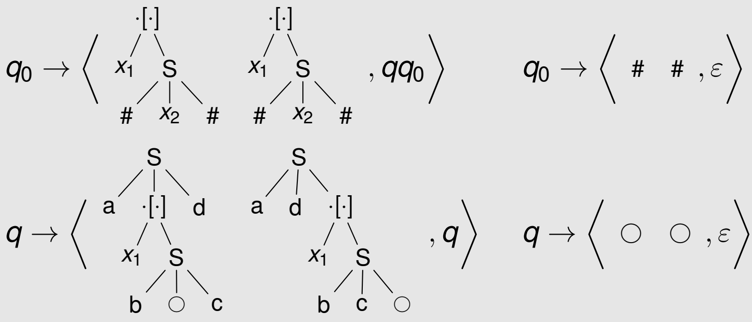
- $t^{\mathrm{E}} \in T_\Sigma$
- $t_1^{\mathrm{E}} \in C_\Sigma$ for every subtree of the form $\cdot[\cdot](t_1, t_2)$ in t

Lemma

Well-behaved trees form a regular tree language

Well-behaved XTOP

Example



Main Theorem

κ_G and κ_M: unidirectional derivation semantics

Theorem

For every STAG G there is a well-behaved XTOP M such that

$$\kappa_G = (\kappa_M)^{\mathrm{E}}$$

and vice versa

Contents

Bimorphism Semantics

Note

The bimorphism semantics is

- taken from [BÜCHSE, NEDERHOF, VOGLER 2011]
- similar (and equivalent) to the synchronous derivation semantics
- written as τ_G

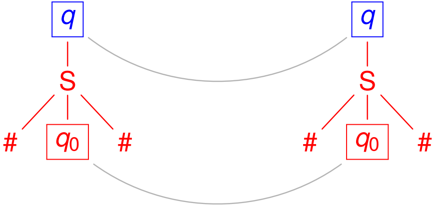
Theorem (Theorem 4 of [MALETTI 2007])

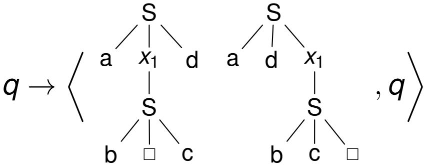
$\tau_M = \kappa_M$ *for every XTOP M*

Bimorphism Semantics

$$q_0 \to \left\langle \; x_1\big(S(\#, x_2, \#)\big) \;\; x_1\big(S(\#, x_2, \#)\big) \;, qq_0 \right\rangle \qquad (\rho_1)$$

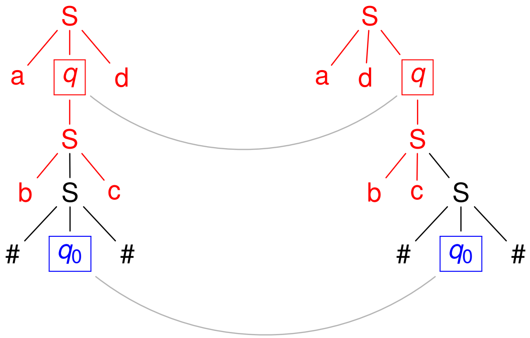
Bimorphism Semantics





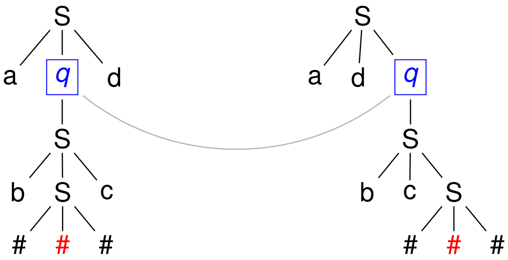
(ρ_3)

Bimorphism Semantics



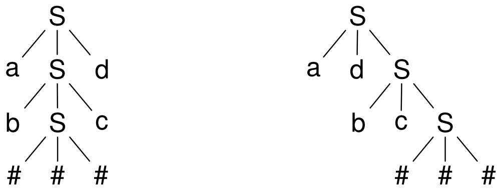
$$q_0 \to \left\langle \# \quad \# \, , \varepsilon \right\rangle \qquad (\rho_4)$$

Bimorphism Semantics



$$q \rightarrow \left\langle \square \quad \square \; , \varepsilon \right\rangle \qquad (\rho_2)$$

Bimorphism Semantics



$$q \to \left\langle \square \quad \square \;, \varepsilon \right\rangle \qquad (\rho_2)$$

Main Theorem

τ_G and τ_M: bimorphism semantics

Theorem

For every STAG G there is a well-behaved XTOP M such that

$$\tau_G = (\tau_M)^{\mathrm{E}}$$

and vice versa

Contents

Uniform STAG

Definition

STAG (Q, Σ, q_0, R) **uniform** if

- $Q = Q^{(1,1)} \cup \{q_0\}$
- q_0 does not occur in the right-hand sides

Theorem

For every STAG G there is a uniform STAG G' with $\tau_G = \tau_{G'}$

Summary

κ_G and κ_M: unidirectional derivation semantics
τ_G and τ_M: bimorphism semantics

Corollary

For a tree transformation τ, the following are equivalent:

1. $\exists$ *STAG G with* $\tau = \kappa_G$
2. $\exists$ *well-behaved XTOP M with* $\tau = (\kappa_M)^{\mathrm{E}}$
3. $\exists$ *well-behaved XTOP M with* $\tau = (\tau_M)^{\mathrm{E}}$
4. $\exists$ *STAG G with* $\tau = \tau_G$
5. $\exists$ *uniform STAG G with* $\tau = \tau_G$

References

- BÜCHSE, NEDERHOF, VOGLER: Tree parsing with synchronous tree-adjoining grammars. In *Parsing Technologies* (IWPT 2011)
- JOSHI, KOSARAJU, YAMADA: String adjunct grammars. In *Switching and Automata Theory* (SWAT 1969)
- KALLMEYER: A lexicalized tree-adjoining grammar for a fragment of German focussing on syntax and semantics. *Emmy-Noether group 2010*
- MALETTI: Compositions of extended top-down tree transducers. *Inform. Comput.* 206(9–10), 2008
- MALETTI: A tree transducer model for synchronous tree-adjoining grammars. In *Association for Computational Linguistics* (ACL 2010)
- SHIEBER, SCHABES: Synchronous tree-adjoining grammars. In *Computational Linguistics* (CoLing 1990)
- XTAG GROUP: A lexicalized tree adjoining grammar for English. *Techn. Report IRCS-01-03*. University of Pennsylvania, 2001