Tree Transformations and Dependencies

Andreas Maletti

Institute for Natural Language Processing
University of Stuttgart

andreas.maletti@ims.uni-stuttgart.de

Nara — September 6, 2011

Contents

1 Motivation

2 Extended top-down tree transducer

3 Extended multi bottom-up tree transducer

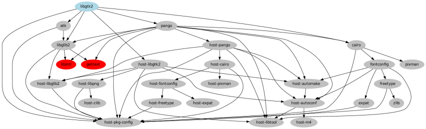
4 Synchronous tree-sequence substitution grammar

Warning!

This won’t be a regular survey talk.

Warning!

This won't be a regular survey talk.



Rather I want to show a useful technique.

Context-free Language

Question

How do you show that a language is not context-free?

Context-free Language

Question

How do you show that a language is not context-free?

Answers

- pumping lemma (BAR-HILLEL lemma)
- semi-linearity (PARIKH's theorem)
- ...

Context-free Language

Hardly ever ...

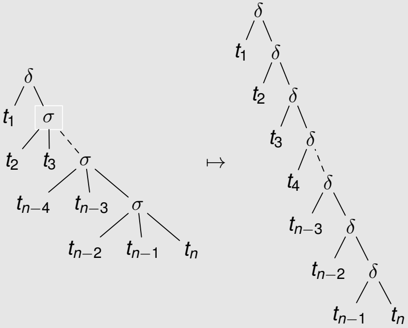
Let us assume that there is a CFG ...
(long case distinction on the shape of its rules)
Contradiction!

Tree Transformation

Application areas

- NLP
- XML processing
- syntax-directed semantics

Example



Tree Transducer

Definition

A formal model computing a tree transformation

Tree Transducer

Definition

A formal model computing a tree transformation

Example

- top-down or bottom-up tree transducer
- synchronous grammar (SCFG, STSG, STAG, etc.)
- multi bottom-up tree transducer
- bimorphism

Tree Transducer

Question

How do you show that a tree transformation cannot be computed by a class of tree transducers?

Tree Transducer

Question

How do you show that a tree transformation cannot be computed by a class of tree transducers?

Answers

- pumping lemma (????)
- size and height dependencies
- domain and image properties
- properties under composition (??)

Tree Transducer

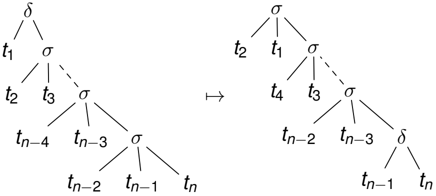
Question

How do you show that a tree transformation cannot be computed by a class of tree transducers?

Typical approach

- Assume that it can be computed .... Contradiction!

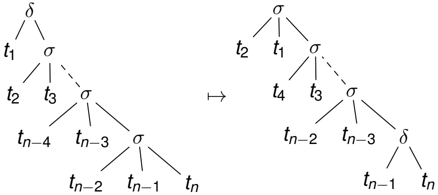
Example



Question

Can this tree transformation be computed by a linear, nondeleting extended top-down tree transducer (XTOP)?

Example



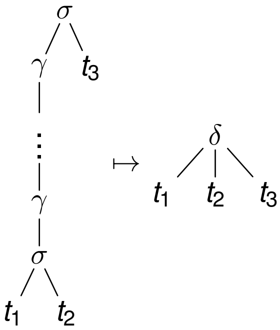
Question

Can this tree transformation be computed by a linear, nondeleting extended top-down tree transducer (XTOP)?

Answer [Arnold, Dauchet 1982]

No! (via indirect proof)

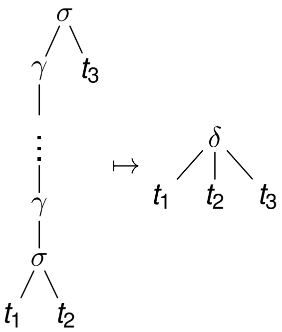
Another Example



Question

Can this tree transformation be computed by an XTOP?

Another Example



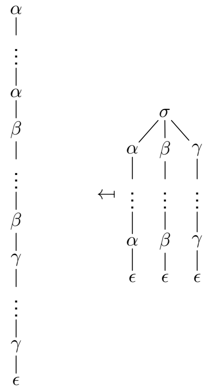
Question

Can this tree transformation be computed by an XTOP?

Answer [∼, Graehl, Hopkins, Knight 2009]

No! (via indirect proof)

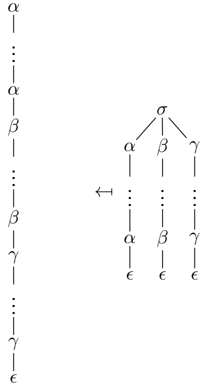
Yet Another Example



Question

Can this tree transformation be computed by an MBOT?

Yet Another Example



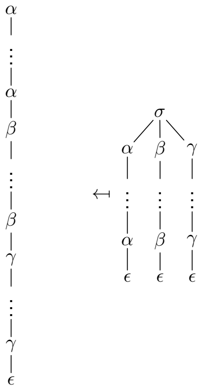
Question

Can this tree transformation be computed by an MBOT?

MBOT

linear, nondeleting multi bottom-up tree transducer

Yet Another Example



Question

Can this tree transformation be computed by an MBOT?

MBOT

linear, nondeleting multi bottom-up tree transducer

Answer

We'll know at the end

Contents

1 Motivation

2 Extended top-down tree transducer

3 Extended multi bottom-up tree transducer

4 Synchronous tree-sequence substitution grammar

Extended Top-down Tree Transducer

Definition (XTOP)

tuple $(Q, \Sigma, \Delta, I, R)$

- Q: *states*
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q$: *initial states*

[ARNOLD, DAUCHET: *Morphismes et bimorphismes d'arbres.* Theor. Comput. Sci. 1982]

Extended Top-down Tree Transducer

Definition (XTOP)

tuple $(Q, \Sigma, \Delta, I, R)$

- Q: *states*
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q$: *initial states*
- $R \subseteq T_\Sigma(Q) \times Q \times T_\Delta(Q)$: finite set of *rules*
  - l and r are linear in Q
  - $\text{var}(l) = \text{var}(r)$

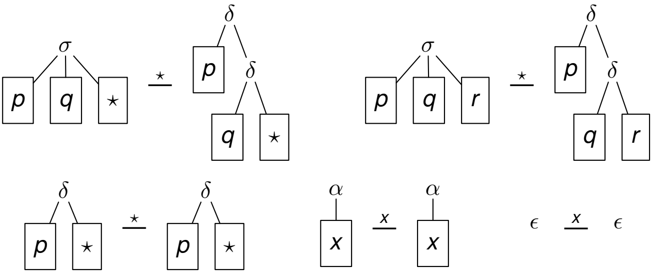
  for every $l \xrightarrow{q} r \in R$

[ARNOLD, DAUCHET: *Morphismes et bimorphismes d'arbres.* Theor. Comput. Sci. 1982]

Example XTOP

Example

XTOP $(Q, \Sigma, \Sigma, \{\star\}, R)$ with $Q = \{\star, p, q, r\}$ and $\Sigma = \{\sigma, \delta, \alpha, \epsilon\}$



Sentential Form

$$\mathcal{L} = \{S \mid S \subseteq \mathbb{N}^* \times \mathbb{N}^*\}$$

Sentential Form

$$\mathcal{L} = \{S \mid S \subseteq \mathbb{N}^* \times \mathbb{N}^*\}$$

Definition (Sentential form)

$\langle \xi, D, \zeta \rangle \in T_\Sigma(Q) \times \mathcal{L} \times T_\Delta(Q)$ such that

- $v \in \text{pos}(\xi)$ and $w \in \text{pos}(\zeta)$ for every $(v, w) \in D$.

Link Structure

Definition

rule $l \xrightarrow{q} r \in R$, positions $v, w \in \mathbb{N}^*$

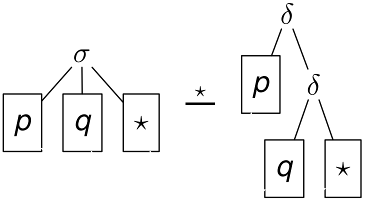
$$\text{links}_{v,w}(l \xrightarrow{q} r) = \bigcup_{p \in Q} \{(vv', ww') \mid v' \in \text{pos}_p(l), w' \in \text{pos}_p(r)\}$$

Link Structure

Definition

rule $l \xrightarrow{q} r \in R$, positions $v, w \in \mathbb{N}^*$

$$\text{links}_{v,w}(l \xrightarrow{q} r) = \bigcup_{p \in Q} \{(vv', ww') \mid v' \in \text{pos}_p(l), w' \in \text{pos}_p(r)\}$$

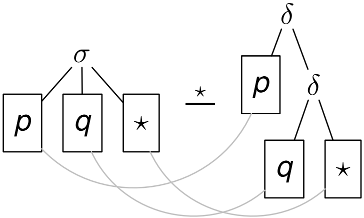


Link Structure

Definition

rule $l \xrightarrow{q} r \in R$, positions $v, w \in \mathbb{N}^*$

$$\text{links}_{v,w}(l \xrightarrow{q} r) = \bigcup_{p \in Q} \{(vv', ww') \mid v' \in \text{pos}_p(l), w' \in \text{pos}_p(r)\}$$



Derivation

Definition

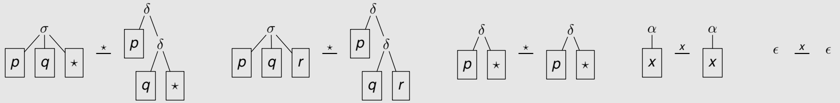
$$\langle \xi, D, \zeta \rangle \Rightarrow \langle \xi[l]_v,\ D \cup \text{links}_{v,w}(l \xrightarrow{q} r),\ \zeta[r]_w \rangle$$

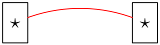
if

- rule $l \xrightarrow{q} r \in R$
- position $v \in \text{pos}_q(\xi)$
- position $w \in \text{pos}_q(\zeta)$ such that $(v, w) \in D$

Derivation

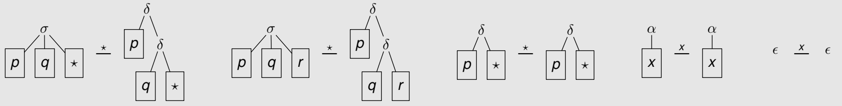
Rules

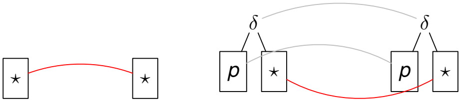




Derivation

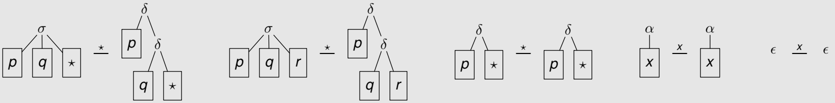
Rules

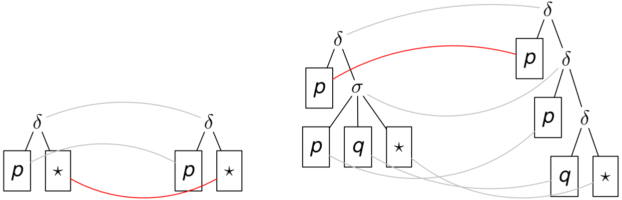




Derivation

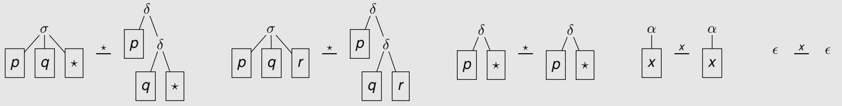
Rules

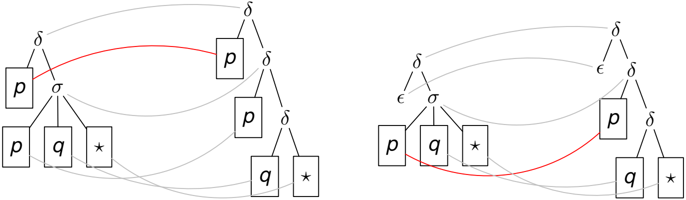




Derivation

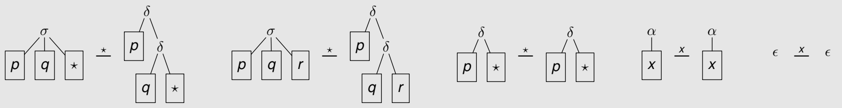
Rules

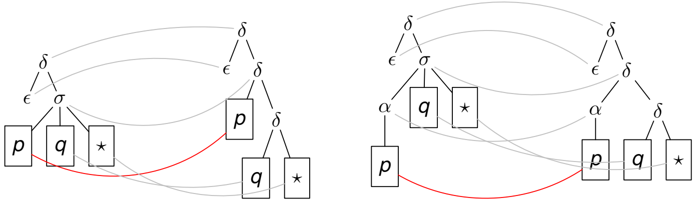




Derivation

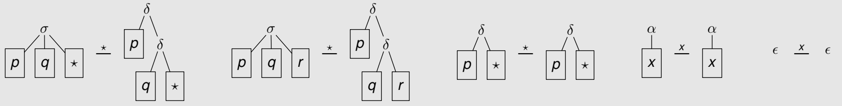
Rules

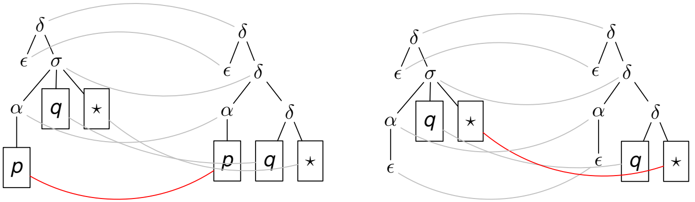




Derivation

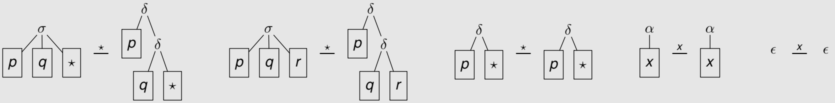
Rules

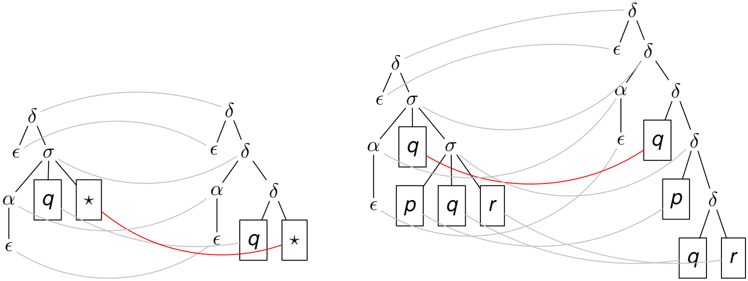




Derivation

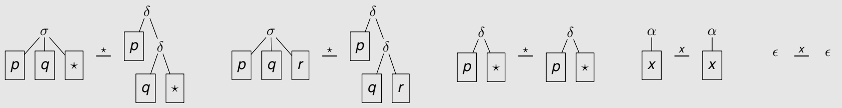
Rules

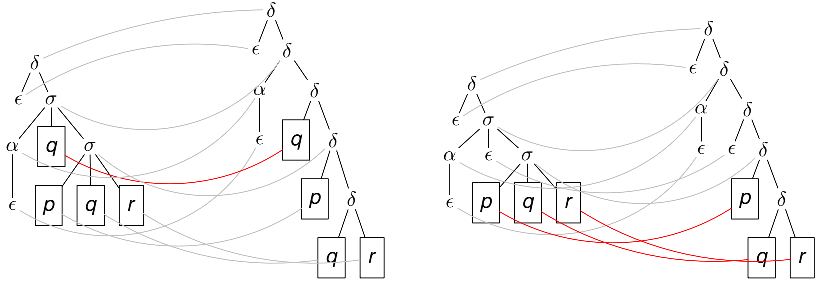




Derivation

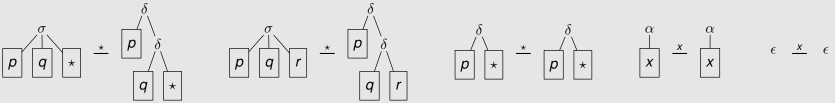
Rules

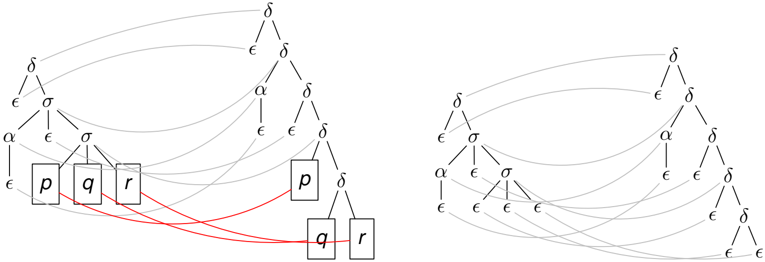




Derivation

Rules





Semantics of XTOP

Definition

- the dependencies dep(M)

$$\{\langle t, D, u\rangle \in T_\Sigma \times \mathcal{L} \times T_\Delta \mid \exists q \in I \colon \langle q, \{(\varepsilon, \varepsilon)\}, q\rangle \Rightarrow^*_M \langle t, D, u\rangle\}$$

- the tree transformation $M = \{(t, u) \mid \langle t, D, u\rangle \in \text{dep}(M)\}$

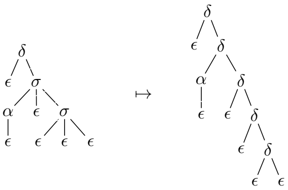
Semantics of XTOP

Definition

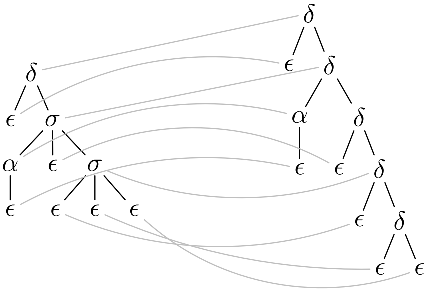
- the dependencies dep(M)

$$\{\langle t, D, u\rangle \in T_\Sigma \times \mathcal{L} \times T_\Delta \mid \exists q \in I \colon \langle q, \{(\varepsilon, \varepsilon)\}, q\rangle \Rightarrow^*_M \langle t, D, u\rangle\}$$

- the tree transformation $M = \{(t, u) \mid \langle t, D, u\rangle \in \text{dep}(M)\}$



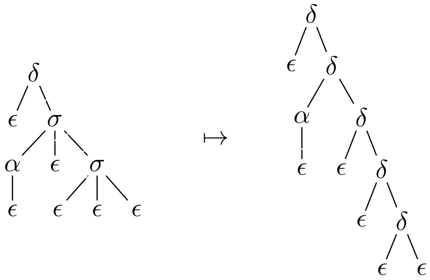
Semantics of XTOP



Semantics of XTOP

Example

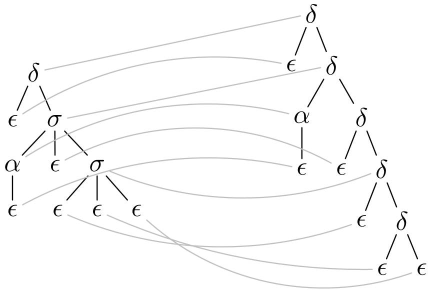
$$D = \{(\varepsilon, \varepsilon), (1,1), (2,2), (21,21), (211,211), (22,221), (23,222), (231,2221), (232,22221), (233,22222)\}$$



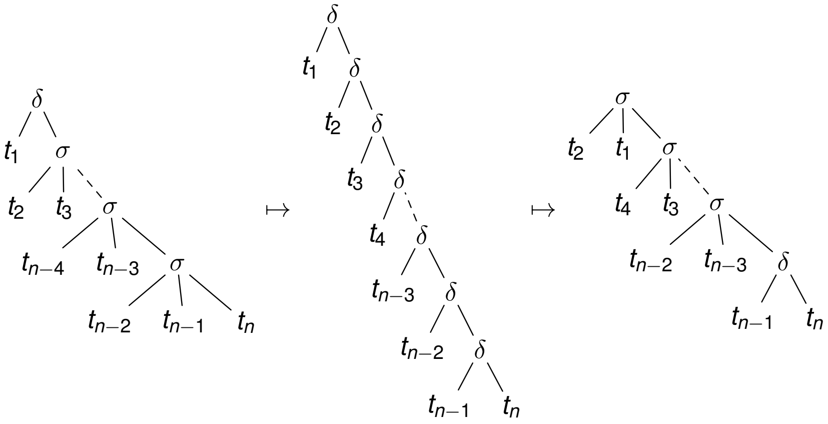
Semantics of XTOP

Example

$$D = \{(\varepsilon, \varepsilon), (1, 1), (2, 2), (21, 21), (211, 211), (22, 221), (23, 222), (231, 2221), (232, 22221), (233, 22222)\}$$



Non-closure under Composition



[ARNOLD, DAUCHET: *Morphismes et bimorphismes d'arbres.* Theor. Comput. Sci. 1982]

Special Linking Structure

Definition

$D \in \mathcal{L}$ is input hierarchical if for all $(v_1, w_1), (v_2, w_2) \in D$

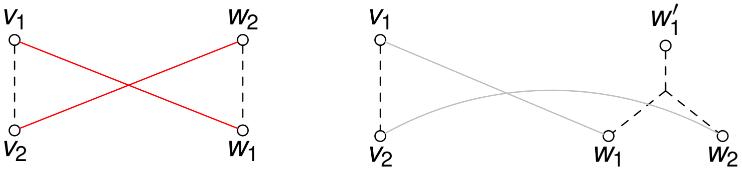
- $v_1 < v_2$ implies $w_2 \not< w_1$ and $\exists w_1' \leq w_2 \colon (v_1, w_1') \in D$

Special Linking Structure

Definition

$D \in \mathcal{L}$ is input hierarchical if for all $(v_1, w_1), (v_2, w_2) \in D$

- $v_1 < v_2$ implies $w_2 \not< w_1$ and $\exists w_1' \leq w_2 \colon (v_1, w_1') \in D$

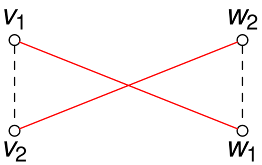


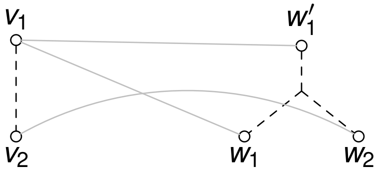
Special Linking Structure

Definition

$D \in \mathcal{L}$ is input hierarchical if for all $(v_1, w_1), (v_2, w_2) \in D$

- $v_1 < v_2$ implies $w_2 \not< w_1$ and $\exists w_1' \leq w_2 : (v_1, w_1') \in D$



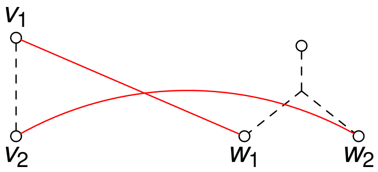


Special Linking Structure

Definition

$D \in \mathcal{L}$ is strictly input hierarchical if for all $(v_1, w_1), (v_2, w_2) \in D$

- $v_1 < v_2$ implies $w_2 \not< w_1$ and $\exists w_1' \leq w_2 \colon (v_1, w_1') \in D$
- $w_1 \not\leq w_2$ implies $v_1 \not\leq v_2$

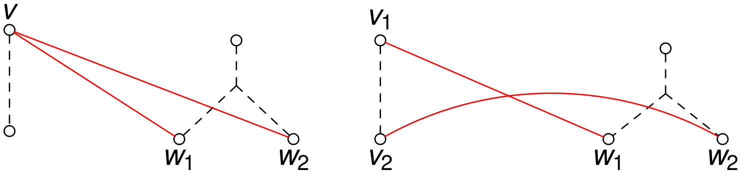


Special Linking Structure

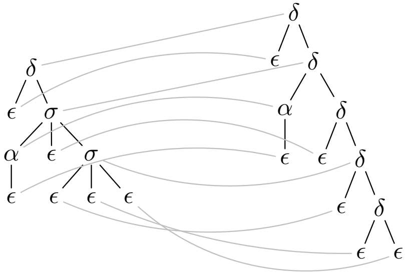
Definition

$D \in \mathcal{L}$ is strictly input hierarchical if for all $(v_1, w_1), (v_2, w_2) \in D$

- $v_1 < v_2$ implies $w_2 \nless w_1$ and $\exists w_1' \leq w_2 \colon (v_1, w_1') \in D$
- $w_1 \nleq w_2$ implies $v_1 \nleq v_2$



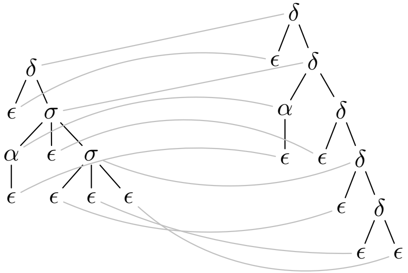
Dependencies

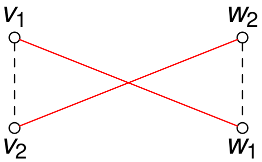


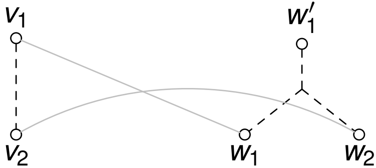
Example

This linking structure is strictly (input and output) hierarchical

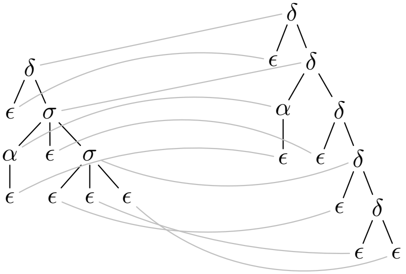
Dependencies

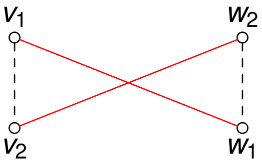


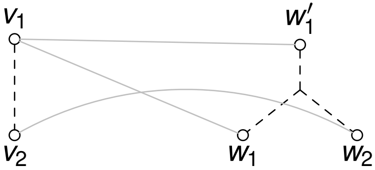




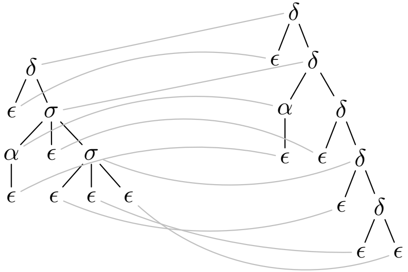
Dependencies

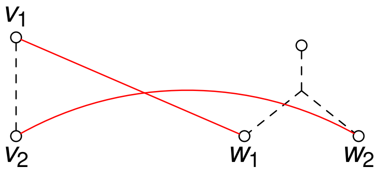




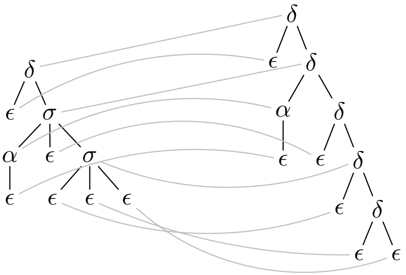


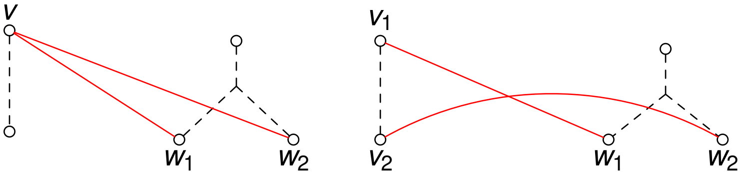
Dependencies





Dependencies





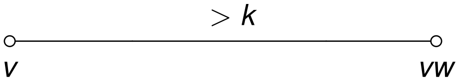
Another Property

Definition

$\mathcal{D} \subseteq \mathcal{L}$ has bounded distance if $\exists k \in \mathbb{N}$ such that $\forall D \in \mathcal{D}$

- if $v, vw \in \mathrm{dom}(D)$ with $|w| > k$, then $\exists w' \leq w$ with
  - $vw' \in \mathrm{dom}(D)$
  - $|w'| \leq k$

Illustration:



Another Property

Definition

$\mathcal{D} \subseteq \mathcal{L}$ has bounded distance if $\exists k \in \mathbb{N}$ such that $\forall D \in \mathcal{D}$

- if $v, vw \in \text{dom}(D)$ with $|w| > k$, then $\exists w' \leq w$ with
  - $vw' \in \text{dom}(D)$
  - $|w'| \leq k$

Illustration:

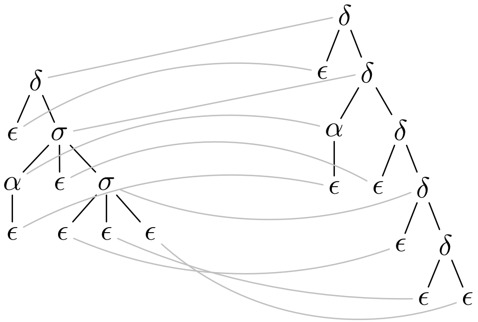
Another Property

Definition

$\mathcal{D} \subseteq \mathcal{L}$ has bounded distance if $\exists k \in \mathbb{N}$ such that $\forall D \in \mathcal{D}$

- if $v, vw \in \text{dom}(D)$ with $|w| > k$, then $\exists w' \leq w$ with
  - $vw' \in \text{dom}(D)$
  - $|w'| \leq k$
- (symmetric property for range)

Bounded Distance



Example

- input side: bounded by 1
- output side: bounded by 2

Main Result

Theorem

The dependencies computed by an XTOP are:

- *strictly (input and output) hierarchical*
- *bounded distance*

Compatibility

Definition

- $\langle \xi, D, \zeta \rangle$ is compatible with $\langle \xi, D', \zeta \rangle$ if $D \subseteq D'$

Compatibility

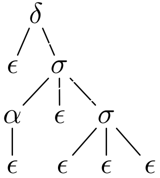
Definition

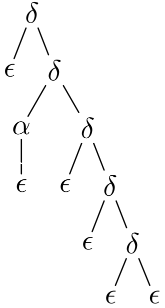
- $\langle \xi, D, \zeta \rangle$ is compatible with $\langle \xi, D', \zeta \rangle$ if $D \subseteq D'$
- set L is compatible with L' if
  for all $\langle \xi, \_, \zeta \rangle \in L$ (some computation)
  - exists $\langle \xi, D, \zeta \rangle \in L$ (implemented computation)
  - exists compatible $\langle \xi, D', \zeta \rangle \in L'$ (compatible computation)

Compatibility

Illustration

Some computation

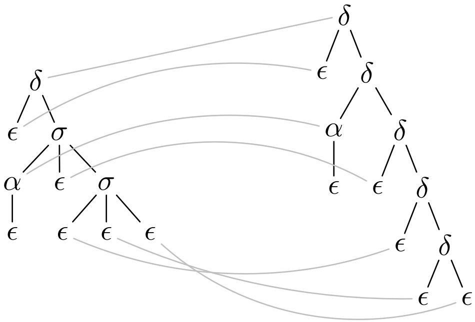




Compatibility

Illustration

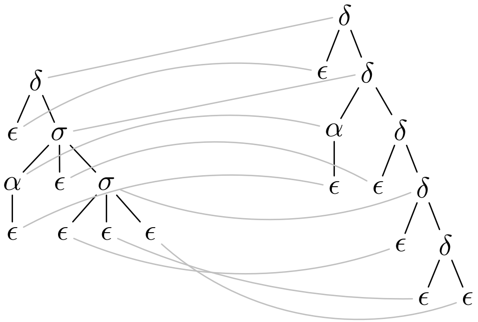
Implemented computation



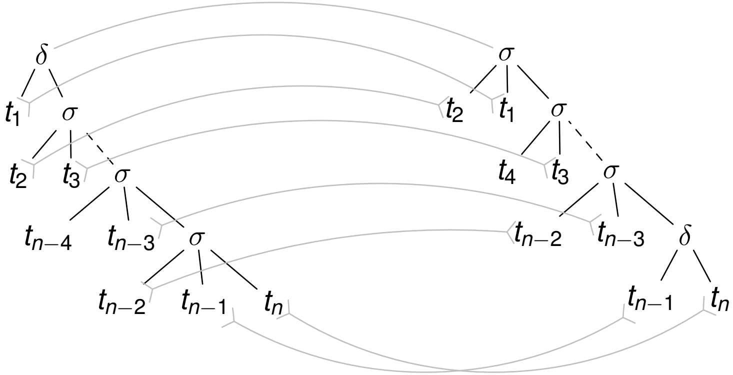
Compatibility

Illustration

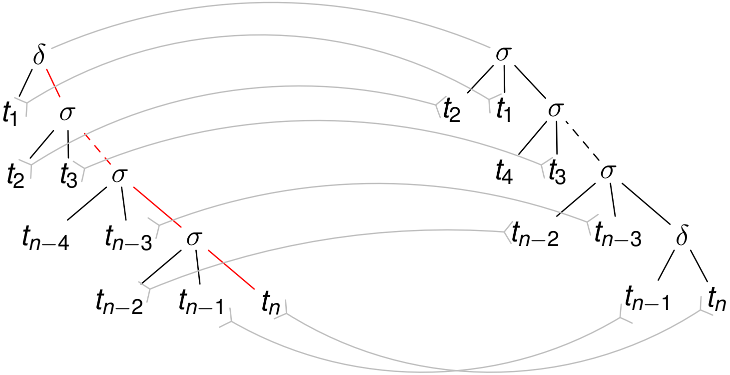
Compatible computation



Back to the Example

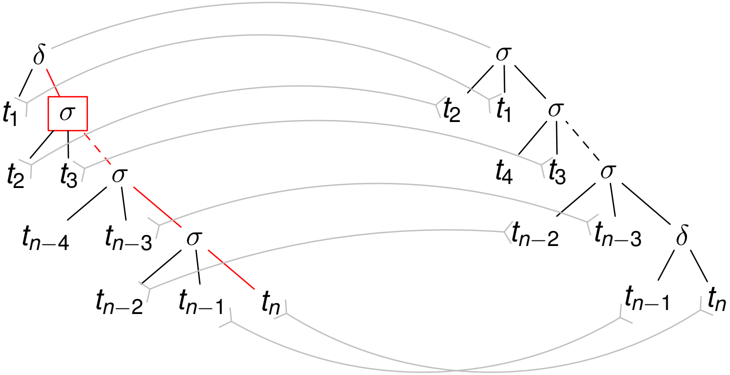


Back to the Example



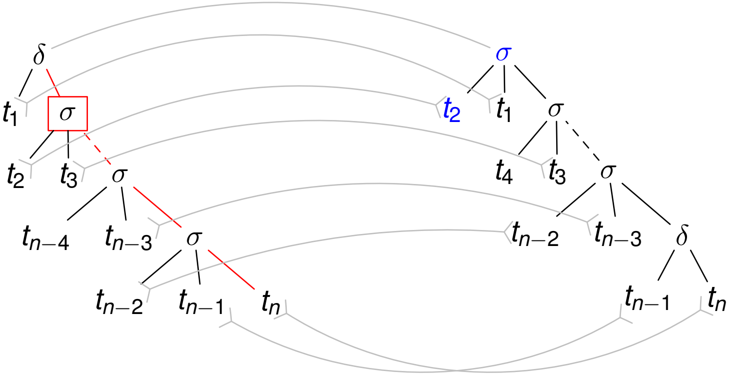
use boundedness

Back to the Example

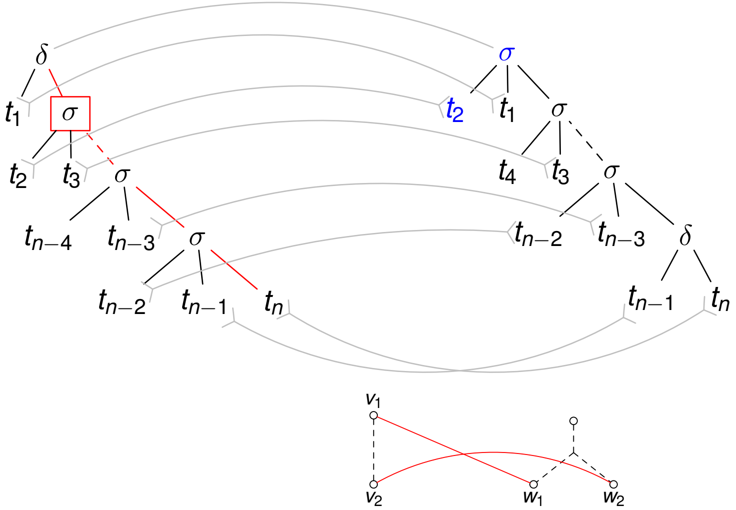


use boundedness

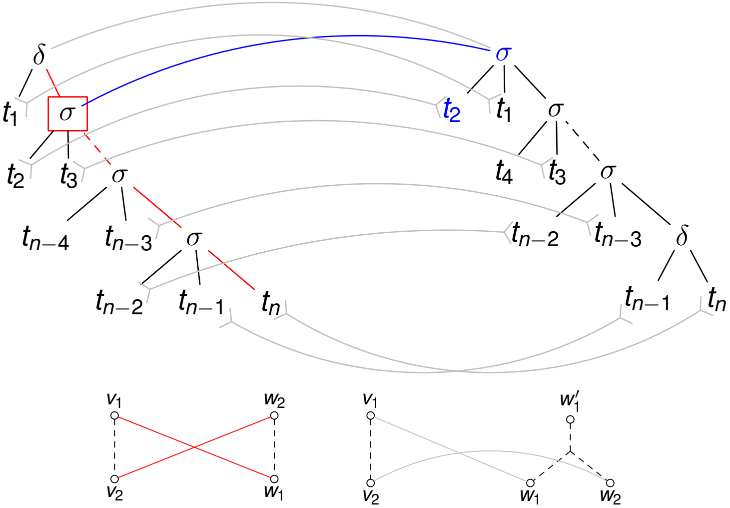
Back to the Example



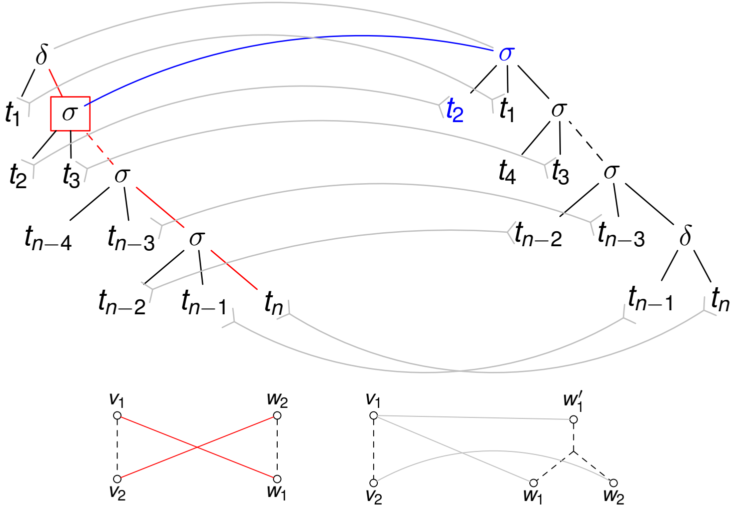
Back to the Example



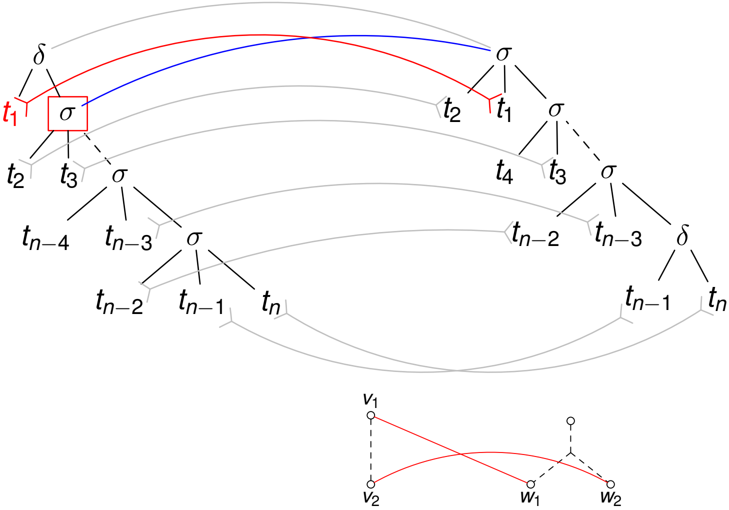
Back to the Example



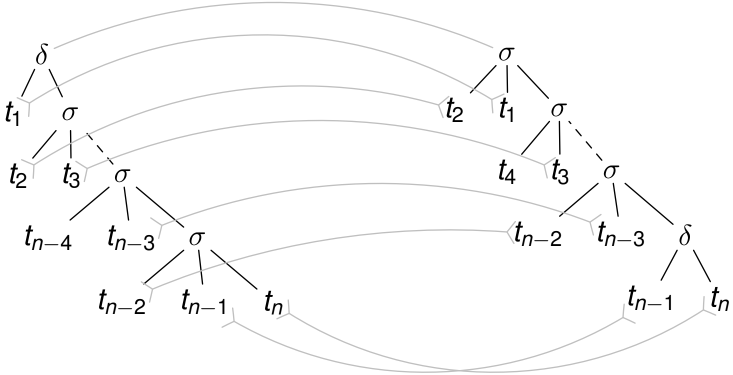
Back to the Example



Back to the Example



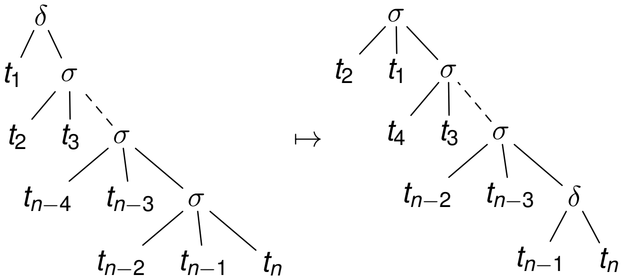
Back to the Example



Theorem

The dependencies above are incompatible with any XTOP

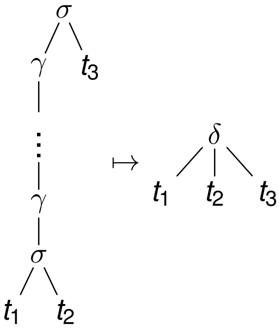
A First Instance



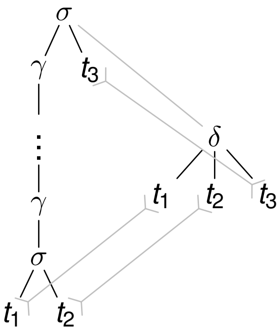
Theorem (ARNOLD, DAUCHET 1982)

The transformation above cannot be computed by any XTOP

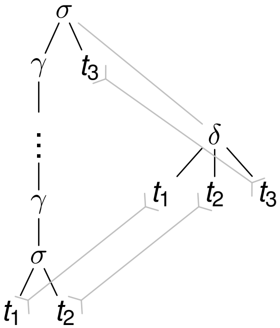
A Second Instance



A Second Instance

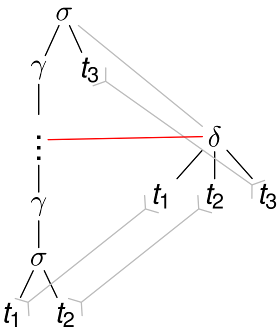


A Second Instance



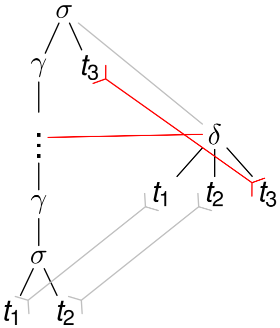
use boundedness

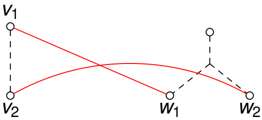
A Second Instance



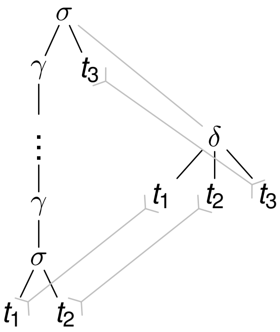
use boundedness

A Second Instance





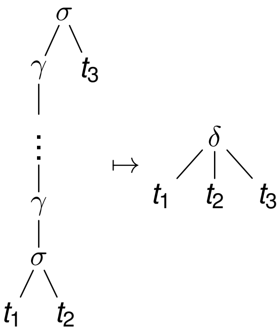
A Second Instance



Theorem

The dependencies above are incompatible with any XTOP

A Second Instance



Theorem (∼, GRAEHL, HOPKINS, KNIGHT 2009)

The transformation above cannot be computed by any XTOP

Contents

1 Motivation

2 Extended top-down tree transducer

3 Extended multi bottom-up tree transducer

4 Synchronous tree-sequence substitution grammar

Extended Multi Bottom-up Tree Transducer

Definition (MBOT)

tuple $(Q, \Sigma, \Delta, I, R)$

- Q: ranked alphabet of *states*
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q_1$: unary *initial states*

Extended Multi Bottom-up Tree Transducer

Definition (MBOT)

tuple $(Q, \Sigma, \Delta, I, R)$

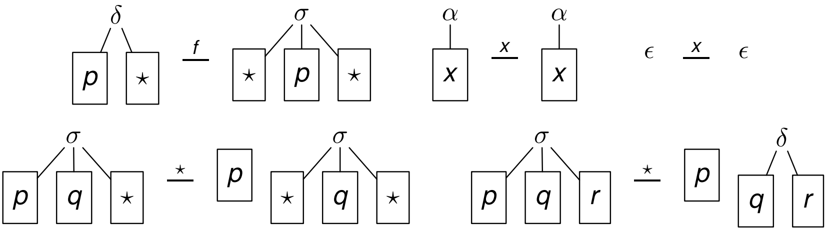
- Q: ranked alphabet of *states*
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q_1$: unary *initial states*
- $R \subseteq T_\Sigma(Q) \times Q \times T_\Delta(Q)^*$: finite set of rules
  - l is linear in Q
  - $\text{rk}(q) = |\vec{r}|$
  - $\text{var}(\vec{r}) \subseteq \text{var}(l)$

  for every $(l, q, \vec{r}) \in R$

Example MBOT

Example

$(Q, \Sigma, \Sigma, \{f\}, R)$ with $Q = \{\star^{(2)}, p^{(1)}, q^{(1)}, r^{(1)}, f^{(1)}\}$ and
$\Sigma = \{\sigma, \delta, \alpha, \epsilon\}$



New Linking Structure

Definition

rule $l \xrightarrow{q} \vec{r} \in R$, positions $v, w_1, \ldots, w_n \in \mathbb{N}^*$
$\vec{w} = w_1 \cdots w_n$ with $n = \mathrm{rk}(q)$

$$\mathrm{links}_{v,\vec{w}}(l \xrightarrow{q} \vec{r}) = \bigcup_{p \in Q} \bigcup_{i=1}^{n} \{(vv', w_i w_i') \mid v' \in \mathrm{pos}_p(l), w_i' \in \mathrm{pos}_p(r_i)\}$$

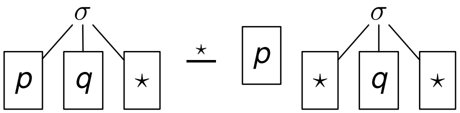
New Linking Structure

Definition

rule $l \xrightarrow{q} \vec{r} \in R$, positions $v, w_1, \ldots, w_n \in \mathbb{N}^*$
$\vec{w} = w_1 \cdots w_n$ with $n = \mathrm{rk}(q)$

$$\mathrm{links}_{v,\vec{w}}(l \xrightarrow{q} \vec{r}) = \bigcup_{p \in Q} \bigcup_{i=1}^{n} \{(vv', w_i w_i') \mid v' \in \mathrm{pos}_p(l), w_i' \in \mathrm{pos}_p(r_i)\}$$

Illustration:



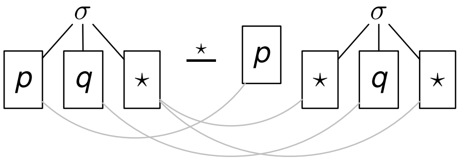
New Linking Structure

Definition

rule $l \xrightarrow{q} \vec{r} \in R$, positions $v, w_1, \dots, w_n \in \mathbb{N}^*$
$\vec{w} = w_1 \cdots w_n$ with $n = \mathrm{rk}(q)$

$$\mathrm{links}_{v,\vec{w}}(l \xrightarrow{q} \vec{r}) = \bigcup_{p \in Q} \bigcup_{i=1}^{n} \{(vv', w_i w_i') \mid v' \in \mathrm{pos}_p(l), w_i' \in \mathrm{pos}_p(r_i)\}$$

Illustration:



Derivation

Definition

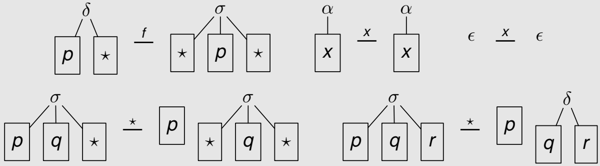
$$\langle \xi, D, \zeta \rangle \Rightarrow \langle \xi[l]_v,\; D \cup \text{links}_{v,\vec{w}}(l \xrightarrow{q} \vec{r}),\; \zeta[\vec{r}]_{\vec{w}} \rangle$$

if

- rule $l \xrightarrow{q} \vec{r} \in R$ with $\text{rk}(q) = n$
- position $v \in \text{pos}_q(\xi)$
- $\vec{w} = w_1 \cdots w_n \in \text{pos}_q(\zeta)^*$ with
  - $w_1 \sqsubset \cdots \sqsubset w_n$
  - $\{w_1, \dots, w_n\} = \{w \mid (v, w) \in D\}$

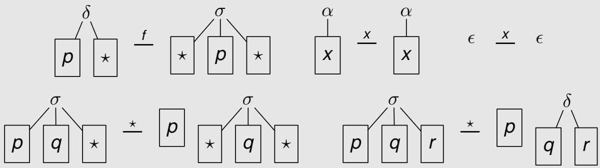
Derivation

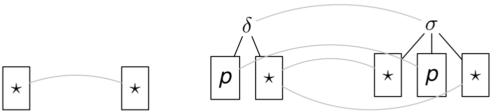
Rules



Derivation

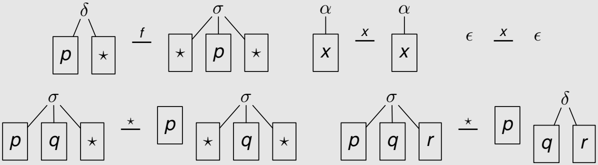
Rules

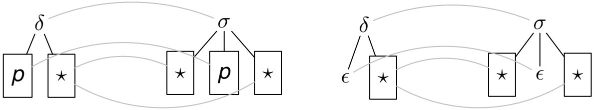




Derivation

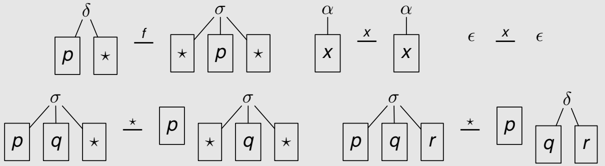
Rules

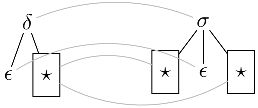


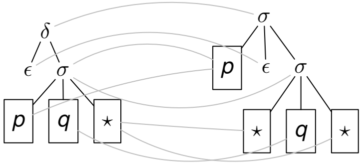


Derivation

Rules

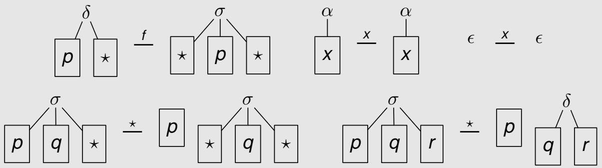


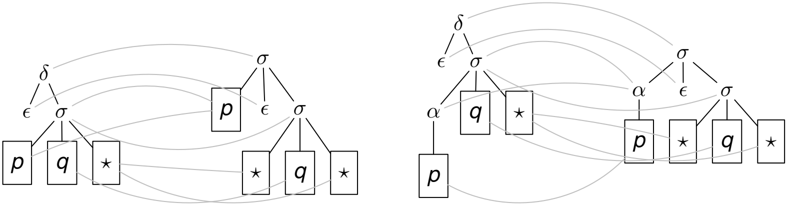




Derivation

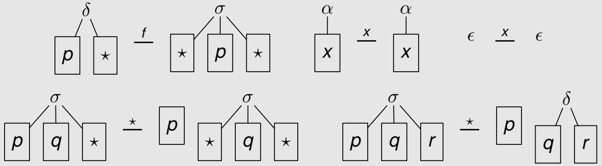
Rules

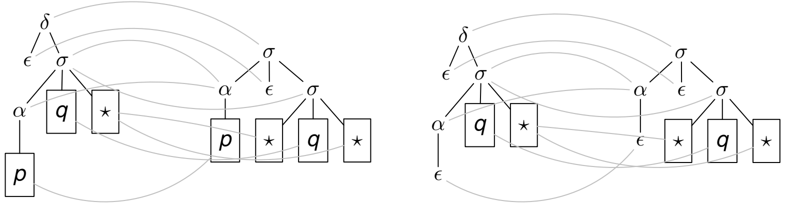




Derivation

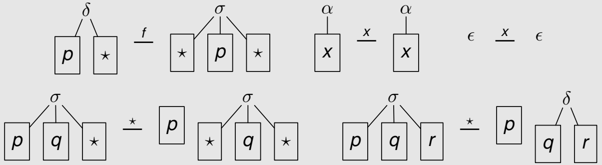
Rules

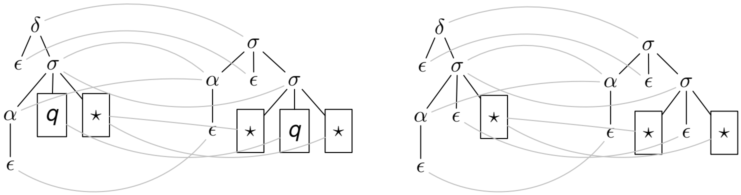




Derivation

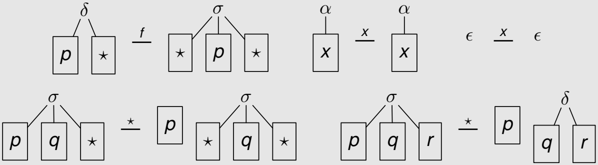
Rules

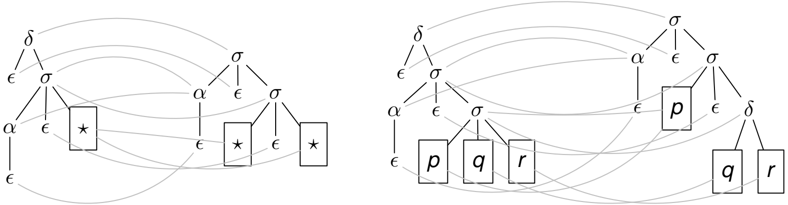




Derivation

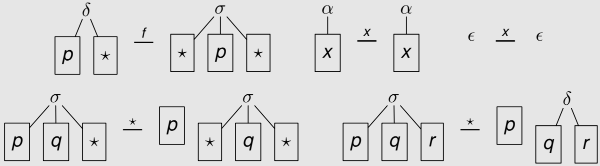
Rules

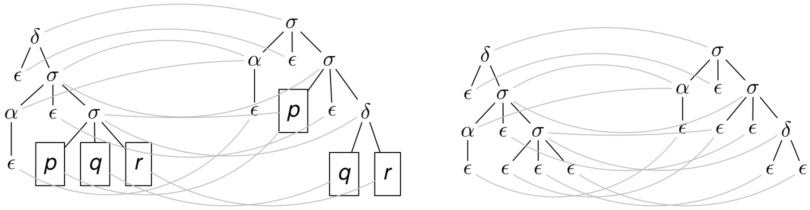




Derivation

Rules





Semantics of MBOT

Definition

MBOT M computes

- dependencies $\text{dep}(M)$

$$\{\langle t, D, u\rangle \in T_\Sigma \times \mathcal{L} \times T_\Delta \mid \exists q \in I \colon \langle q, \{(\varepsilon, \varepsilon)\}, q\rangle \Rightarrow^*_M \langle t, D, u\rangle\}$$

- the tree transformation $M = \{(t, u) \mid \langle t, D, u\rangle \in \text{dep}(M)\}$

Semantics of MBOT

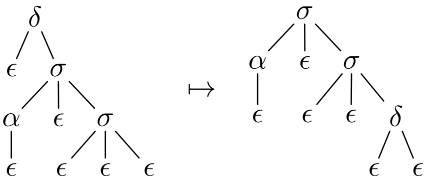
Definition

MBOT M computes

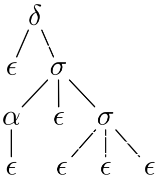
- dependencies $\mathrm{dep}(M)$

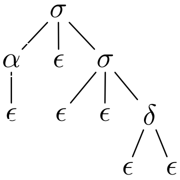
$$\{\langle t, D, u\rangle \in T_\Sigma \times \mathcal{L} \times T_\Delta \mid \exists q \in I \colon \langle q, \{(\varepsilon, \varepsilon)\}, q\rangle \Rightarrow_M^* \langle t, D, u\rangle\}$$

- the tree transformation $M = \{(t, u) \mid \langle t, D, u\rangle \in \mathrm{dep}(M)\}$



Semantics of MBOT

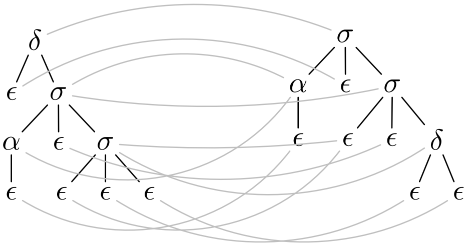




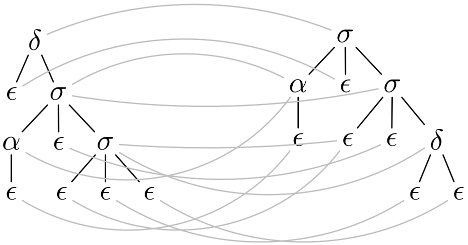
Example

$$D = \{(\varepsilon, \varepsilon), (1, 2), (2, 1), (2, 3), (21, 1), (211, 11), (22, 32), (23, 31), (23, 33), (231, 31), (232, 331), (233, 332)\}$$

Semantics of MBOT



Semantics of MBOT

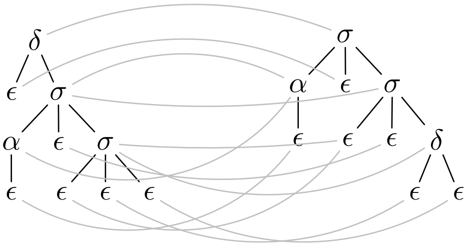


Example

Above dependencies are

- input hierarchical
- strictly output hierarchical

Semantics of MBOT

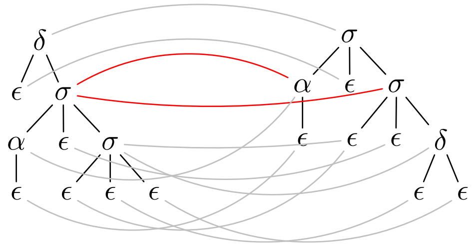


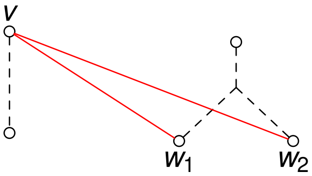
Example

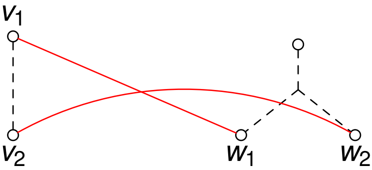
Above dependencies are

- input hierarchical but not strictly
- strictly output hierarchical

Semantics of MBOT







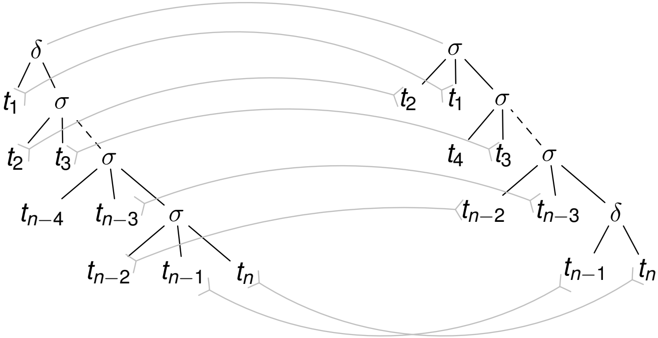
Main Result

Theorem

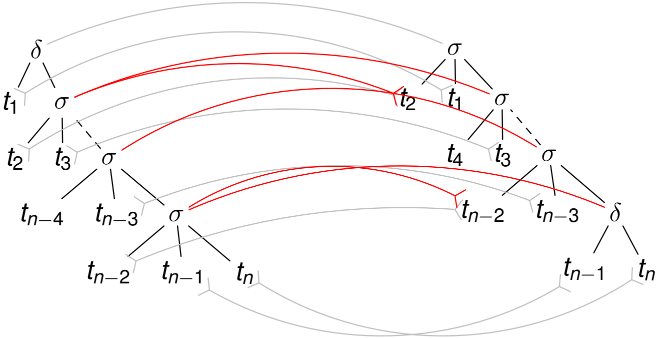
The dependencies computed by an MBOT are:

- *input hierarchical*
- *strictly output hierarchical*
- *bounded distance*

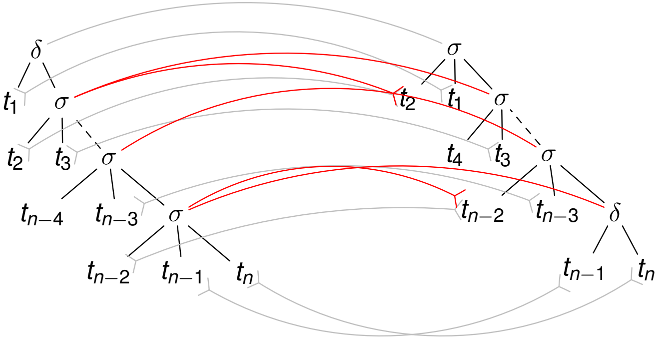
Link Structure for the Example



Link Structure for the Example



Link Structure for the Example



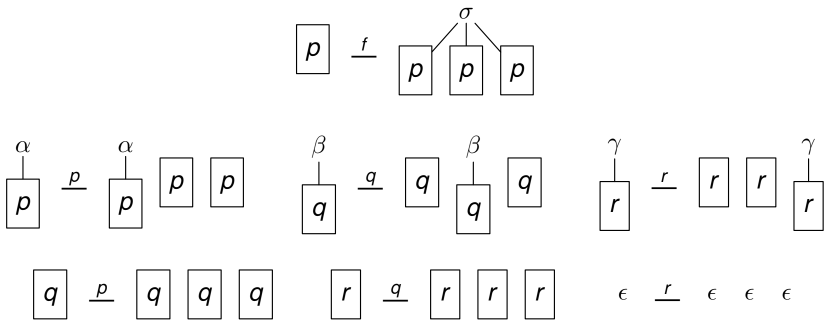
Theorem

Above dependencies are compatible with an MBOT

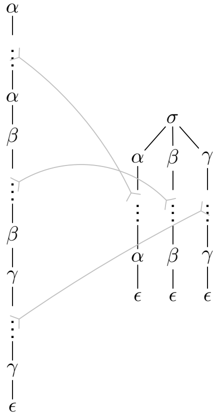
Another MBOT

Example

$(Q, \Sigma, \Delta, \{f\}, R)$ with $Q = \{p^{(3)}, q^{(3)}, r^{(3)}, f^{(1)}\}$
$\Sigma = \{\epsilon, \alpha, \beta, \gamma\}$ and $\Delta = \Sigma \cup \{\sigma\}$



Another MBOT

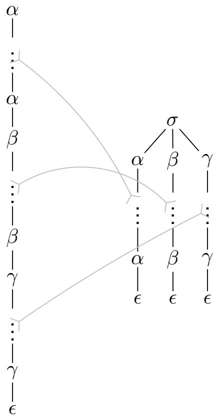


taken from

Example 4.5 of [RADMACHER: *An automata theoretic approach to the theory of rational tree relations.* TR AIB-2008-05, RWTH Aachen, 2008]

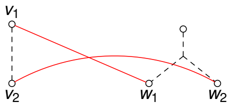
$$\{(\alpha^{\ell}(\beta^{m}(\gamma^{n}(\epsilon))),\ \sigma(\alpha^{\ell}(\epsilon), \beta^{m}(\epsilon), \gamma^{n}(\epsilon))) \mid \ell, m, n \in \mathbb{N}\}$$

Another MBOT

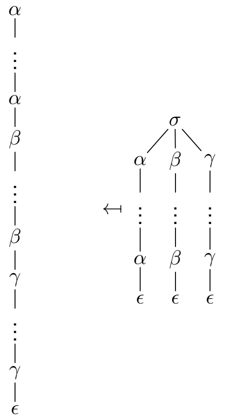


Theorem

These inverse dependencies are not compatible with any MBOT
(because the displayed dependencies are not strictly input hierarchical)

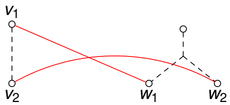


Another MBOT



Theorem

These inverse dependencies are not compatible with any MBOT
(because the displayed dependencies are not strictly input hierarchical)



Theorem

This tree transformation cannot be computed by any MBOT

Contents

1 Motivation

2 Extended top-down tree transducer

3 Extended multi bottom-up tree transducer

4 Synchronous tree-sequence substitution grammar

Synchronous Tree-Sequence Substitution Grammar

Definition (STSSG)

tuple $(Q, \Sigma, \Delta, I, R)$ with

- Q: doubly ranked alphabet
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q_{1,1}$: doubly unary *initial states*

[ZHANG et al.: *A tree sequence alignment-based tree-to-tree translation model.* Proc. ACL, 2008]

Synchronous Tree-Sequence Substitution Grammar

Definition (STSSG)

tuple $(Q, \Sigma, \Delta, I, R)$ with

- Q: doubly ranked alphabet
- Σ, Δ: *input* and *output symbols*
- $I \subseteq Q_{1,1}$: doubly unary *initial states*
- $R \subseteq T_\Sigma(Q)^* \times Q \times T_\Delta(Q)^*$: finite set of rules
  - $\text{rk}(q) = (|\vec{l}|, |\vec{r}|)$ for every $(\vec{l}, q, \vec{r}) \in R$

[ZHANG et al.: *A tree sequence alignment-based tree-to-tree translation model.* Proc. ACL, 2008]

Yet Another Linking Structure

Definition

rule $\vec{l} \xrightarrow{q} \vec{r} \in R$, positions $v_1, \dots, v_m, w_1, \dots, w_n \in \mathbb{N}^*$
with $\mathrm{rk}(q) = (m, n)$

$$\begin{aligned} &\mathrm{links}_{\vec{v},\vec{w}}(\vec{l} \xrightarrow{q} \vec{r}) \\ &= \bigcup_{p \in Q} \bigcup_{j=1}^{m} \bigcup_{i=1}^{n} \{(v_j v_j', w_i w_i') \mid v_j' \in \mathrm{pos}_p(l_j), w_i' \in \mathrm{pos}_p(r_i)\} \end{aligned}$$

where $\vec{v} = v_1 \cdots v_m$ and $\vec{w} = w_1 \cdots w_n$

Derivation

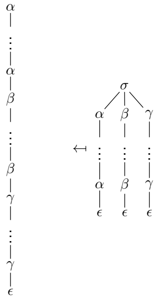
Definition

$$\langle \xi, D, \zeta \rangle \Rightarrow \langle \xi[\vec{l}]_{\vec{v}},\ D \cup \text{links}_{\vec{v},\vec{w}}(\vec{l} \stackrel{q}{\text{---}} \vec{r}),\ \zeta[\vec{r}]_{\vec{w}} \rangle$$

if

- rule $\vec{l} \stackrel{q}{\text{---}} \vec{r} \in R$
- $\vec{v} = v_1 \cdots v_m \in \text{pos}_q(\xi)^*$ and $\vec{w} = w_1 \cdots w_n \in \text{pos}_q(\zeta)^*$
- $v_1 \sqsubset \cdots \sqsubset v_m$, $w_1 \sqsubset \cdots \sqsubset w_n$, and $\text{rk}(q) = (m, n)$
- linked positions
  $\{w_1, \ldots, w_n\} = \bigcup_{j=1}^{m} \{w \mid (v_j, w) \in D\}$
  $\{v_1, \ldots, v_m\} = \bigcup_{i=1}^{n} \{v \mid (v, w_i) \in D\}$

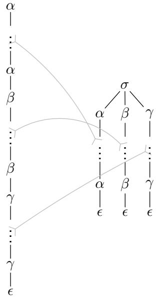
A Final Theorem



Example

This transformation can be computed by an STSSG
(because it is an inverse MBOT)

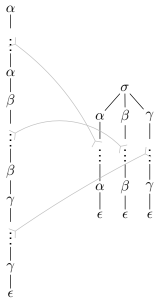
A Final Theorem



Example

This transformation can be computed by an STSSG
(because it is an inverse MBOT)

A Final Theorem



Example

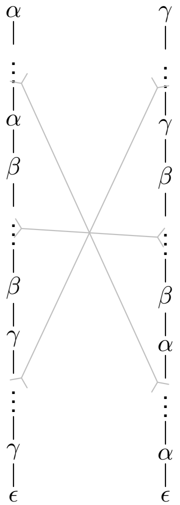
This transformation can be computed by an STSSG
(because it is an inverse MBOT)

Theorem

Dependencies computed by STSSG are

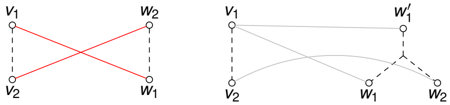
- *(input and output) hierarchical*
- *bounded distance*

A Final Instance

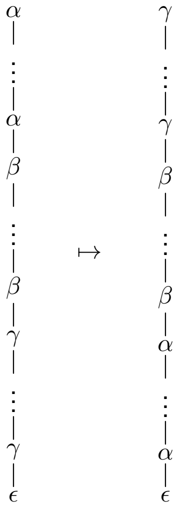


Theorem

These dependencies are not compatible with any STSSG

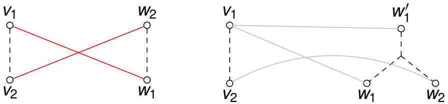


A Final Instance



Theorem

These dependencies are not compatible with any STSSG



Theorem

This transformation cannot be computed by any STSSG

Summary

Computed dependencies by device

	input	output
XTOP	strictly hierarchical	strictly hierarchical

Summary

Computed dependencies by device

	input	output
XTOP	strictly hierarchical	strictly hierarchical
MBOT	hierarchical	strictly hierarchical

Summary

Computed dependencies by device

	input	output
XTOP	strictly hierarchical	strictly hierarchical
MBOT	hierarchical	strictly hierarchical
STSSG	hierarchical	hierarchical

Summary

Computed dependencies by device

	input	output
XTOP	strictly hierarchical	strictly hierarchical
MBOT	hierarchical	strictly hierarchical
STSSG	hierarchical	hierarchical
STAG	—	—

That's all, folks!

Thank you for your attention!

References

- ARNOLD, DAUCHET: *Morphismes et bimorphismes d'arbres.* Theor. Comput. Sci. 20, 1982
- ENGELFRIET, LILIN, MALETTI: *Extended multi bottom-up tree transducers.* Acta Inf. 46, 2009
- MALETTI, GRAEHL, HOPKINS, KNIGHT: *The power of extended top-down tree transducers.* SIAM J. Comput. 39, 2009
- RADMACHER: *An automata theoretic approach to rational tree relations.* Proc. SOFSEM 2008
- RAOULT: *Rational tree relations.* Bull. Belg. Math. Soc. Simon Stevin 4, 1997
- ZHANG, JIANG, AW, LI, TAN, LI: *A tree sequence alignment-based tree-to-tree translation model.* Proc. ACL 2008