Preservation of Recognizability for Weighted Extended Top-down Tree Transducers

Andreas Maletti

Institute for Natural Language Processing
University of Stuttgart

andreas.maletti@ims.uni-stuttgart.de

Nara, Japan — September 7, 2011

Collaborators

reporting joint work with

- JOOST ENGELFRIET (LIACS, Leiden, The Netherlands)
- ZOLTÁN FÜLÖP (U Szeged, Hungary)
- ERIC LILIN (U Lille, France)
- GIORGIO SATTA (U Padua, Italy)
- HEIKO VOGLER (TU Dresden, Germany)

Machine Translation — Unweighted Setup

Schema

Input $\longrightarrow$ $\boxed{\text{Machine translation system}}$ $\longrightarrow$ Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable tree language L such that $L \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Unweighted Setup

Schema

Input $\longrightarrow$ $\boxed{\text{XTT}}$ $\longrightarrow$ Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable tree language L such that $L \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Unweighted Setup

Schema

$$\text{Input} \longrightarrow \boxed{\text{XTT}} \longrightarrow \text{Output}$$

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable tree language L such that $L \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Unweighted Setup

Schema

$$\text{Input} \longrightarrow \boxed{\text{XTT}} \longrightarrow \text{Output}$$

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable tree language L such that $L \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Unweighted Setup

Problem

$M(L)$ is not necessarily recognizable

Answer

- in many cases it fortunately is

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓	✓
XTOP	✗	✓
ln-MBOT	✗	✓
l-MBOT	✗	✓
MBOT	✗	✓
ln-STSSG	✗	✗

Machine Translation — Unweighted Setup

Problem

$M(L)$ is not necessarily recognizable

Answer

- in many cases it fortunately is

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓	✓
XTOP	✗	✓
ln-MBOT	✗	✓
l-MBOT	✗	✓
MBOT	✗	✓
ln-STSSG	✗	✗

Machine Translation — Weighted Setup

Schema

weighted Input $\longrightarrow$ $\boxed{\text{Machine translation system}}$ $\longrightarrow$ weighted Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable weighted tree language L such that $\text{supp}(L) \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Weighted Setup

Schema

weighted Input $\longrightarrow$ WXTT $\longrightarrow$ weighted Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable weighted tree language L such that $\text{supp}(L) \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Weighted Setup

Schema

weighted Input $\longrightarrow$ $\boxed{\text{WXTT}}$ $\longrightarrow$ weighted Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable weighted tree language L such that $\operatorname{supp}(L) \subseteq \{t \mid \operatorname{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Weighted Setup

Schema

weighted Input $\longrightarrow$ $\boxed{\text{WXTT}}$ $\longrightarrow$ weighted Output

Question

What are the translations of sentence f?

Answer

- take recognizable language $\{f\}$
- parse f giving a recognizable weighted tree language L such that $\text{supp}(L) \subseteq \{t \mid \text{yield}(t) = f\}$
- compute forward application $M(L)$

Machine Translation — Weighted Setup

Problem

Again, $M(L)$ is not necessarily recognizable

Answer

- in fewer cases it is

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗	✓
XTOP	✗	✗
ln-MBOT	✗	✓
l-MBOT	✗	✓
MBOT	✗	✗
ln-STSSG	✗	✗

Machine Translation — Weighted Setup

Problem

Again, $M(L)$ is not necessarily recognizable

Answer

- in fewer cases it is

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗	✓
XTOP	✗	✗
ln-MBOT	✗	✓
l-MBOT	✗	✓
MBOT	✗	✗
ln-STSSG	✗	✗

Machine Translation — Weighted Setup

Problem

Again, $M(L)$ is not necessarily recognizable

Answer

- in fewer cases it is

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗ (✓)	✓
XTOP	✗	✗ (✓)
ln-MBOT	✗	✓
l-MBOT	✗	✓(✓)
MBOT	✗	✗ (✓)
ln-STSSG	✗	✗

Contents

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Idempotent if $c + c = c$

Example

- BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$ (idempotent)
- Semiring $(\mathbb{N}, +, \cdot, 0, 1)$ of natural numbers
- Tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ (idempotent)
- Any field, ring, etc.

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Idempotent if $c + c = c$

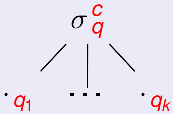
Example

- BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$ (idempotent)
- Semiring $(\mathbb{N}, +, \cdot, 0, 1)$ of natural numbers
- Tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ (idempotent)
- Any field, ring, etc.

Weighted tree automaton

Definition (Berstel, Reutenauer 1982)

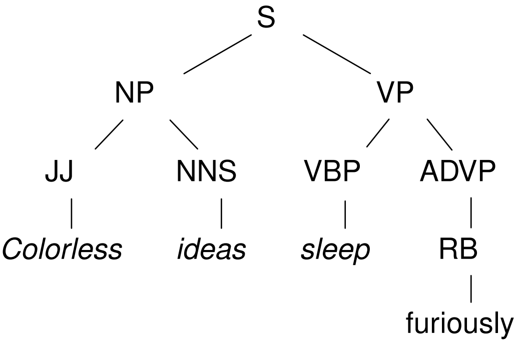
Weighted tree automaton (WTA) $A = (Q, \Sigma, F, \delta)$ with rules



- states $q, q_1, \dots, q_k \in Q$
- rule weight $c \in C$
- k-ary input symbol $\sigma \in \Sigma_k$

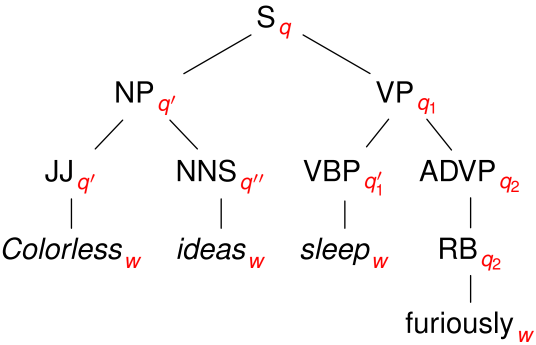
[Berstel, Reutenauer: Recognizable formal power series on trees. Theor. Comput. Sci. 1982]

Run



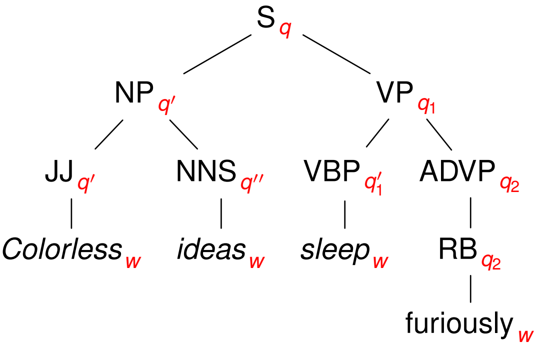
1. arbitrarily assign states
2. look-up rule weights

Run

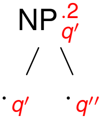


1. arbitrarily assign states
2. look-up rule weights

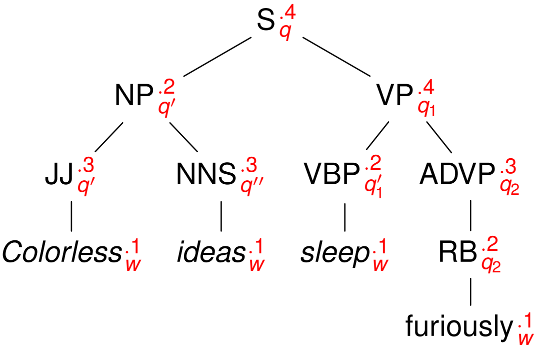
Run



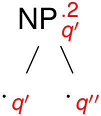
1. arbitrarily assign states
2. look-up rule weights



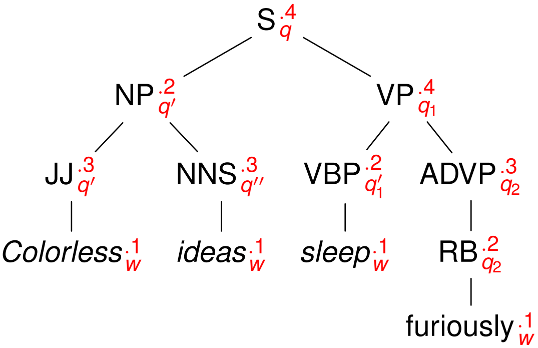
Run



1. arbitrarily assign states
2. look-up rule weights



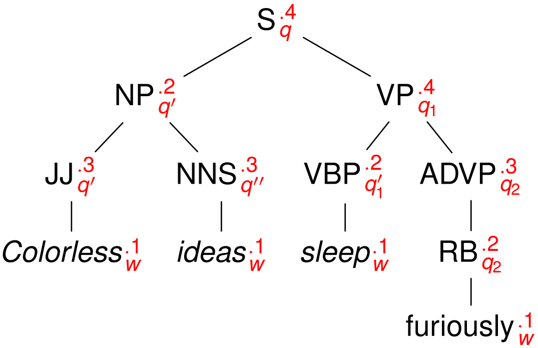
Run



Definition

Weight $\text{wt}(r)$ of run r = product of its weights

Run



Definition

Weight $\mathrm{wt}(r)$ of run r = product of its weights

Example (Weight of the run)

$$\mathrm{wt}(r) = 0.4 \cdot 0.2 \cdot 0.3 \cdot 0.1 \cdot 0.3 \cdot 0.1 \cdot 0.4 \cdot 0.2 \cdot 0.1 \cdot 0.3 \cdot 0.2 \cdot 0.1$$

Semantics

Definition

Weight $A(t)$ of tree t = sum of weights of runs scaled by final weight

$$A(t) = \sum_{r \text{ run on } t} \mathrm{wt}(r) \cdot F(\mathrm{root}(r))$$

Definition

Weighted tree language recognizable if computable by WTA

Semantics

Definition

Weight $A(t)$ of tree t = sum of weights of runs scaled by final weight

$$A(t) = \sum_{r \text{ run on } t} \mathrm{wt}(r) \cdot F(\mathrm{root}(r))$$

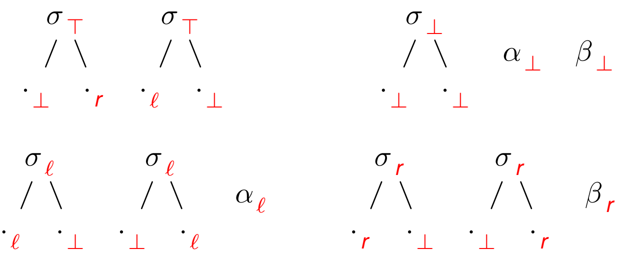
Definition

Weighted tree language recognizable if computable by WTA

Unweighted example

Example

$A = (\{l, r, \top, \bot\}, \Sigma, \{\top\}, \delta)$ with $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}, \beta^{(0)}\}$



- $\top$ reached on ℓ or r in left or right subtree
- $\bot$ can accept any tree
- ℓ and r accept α and β and propagate

Unweighted example

Example

Input tree

$$\sigma(\sigma(\alpha, \alpha), \beta)$$

Recognized language

$$A = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\}$$

Unweighted example

Example

Accepting run

$$
\begin{array}{ccccc}
 & & \sigma_{\top} & & \\
 & / & & \backslash & \\
 & \sigma_{\ell} & & & \beta_{\perp} \\
/ & & \backslash & & \\
\alpha_{\ell} & & \alpha_{\perp} & &
\end{array}
$$

Recognized language

$$A = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\}$$

Unweighted example

Example

Accepting run

$$\sigma_{\top}(\sigma_{\ell}(\alpha_{\ell}, \alpha_{\perp}), \beta_{\perp}) \qquad \sigma_{\top}(\sigma_{\ell}(\alpha_{\perp}, \alpha_{\ell}), \beta_{\perp})$$

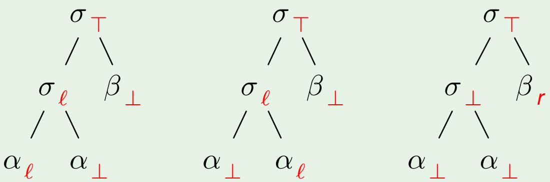
Recognized language

$$A = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\}$$

Unweighted example

Example

Accepting run



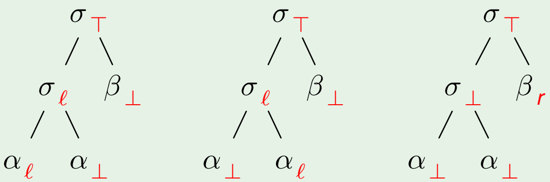
Recognized language

$$A = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\}$$

Unweighted example

Example

Accepting run



Recognized language

$$A = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\}$$

Unweighted example

Example

Accepting run

$$\sigma_{\top}(\sigma_{\ell}(\alpha_{\ell}, \alpha_{\perp}), \beta_{\perp}) \qquad \sigma_{\top}(\sigma_{\ell}(\alpha_{\perp}, \alpha_{\ell}), \beta_{\perp}) \qquad \sigma_{\top}(\sigma_{\perp}(\alpha_{\perp}, \alpha_{\perp}), \beta_{r})$$

Recognized language

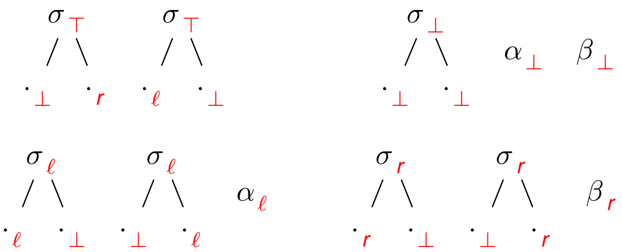
$$\begin{aligned} A &= \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq 0 \text{ or } |t_2|_\beta \neq 0\} \\ &= \{\sigma(t_1, t_2) \mid |t_1|_\alpha + |t_2|_\beta \neq 0\} \end{aligned}$$

Weighted example

Example

$A = (\{l, r, \top, \bot\}, \Sigma, F, \delta)$ over the field $(\mathbb{R}, +, \cdot, 0, 1)$ of reals

- $F(\top) = 1$ and $F(q) = 0$ otherwise
- $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}, \beta^{(0)}\}$



Weighted example

Example

$A = (\{l, r, \top, \bot\}, \Sigma, F, \delta)$ over the field $(\mathbb{R}, +, \cdot, 0, 1)$ of reals

- $F(\top) = 1$ and $F(q) = 0$ otherwise
- $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}, \beta^{(0)}\}$

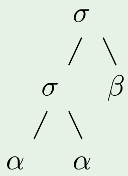
$$\sigma^{1}_{\top}(\cdot_{\bot}, \cdot_{r}) \qquad \sigma^{1}_{\top}(\cdot_{\ell}, \cdot_{\bot}) \qquad \sigma^{1}_{\bot}(\cdot_{\bot}, \cdot_{\bot}) \qquad \alpha^{1}_{\bot} \qquad \beta^{1}_{\bot}$$

$$\sigma^{1}_{\ell}(\cdot_{\ell}, \cdot_{\bot}) \qquad \sigma^{1}_{\ell}(\cdot_{\bot}, \cdot_{\ell}) \qquad \alpha^{1}_{\ell} \qquad \sigma^{1}_{r}(\cdot_{r}, \cdot_{\bot}) \qquad \sigma^{1}_{r}(\cdot_{\bot}, \cdot_{r}) \qquad \beta^{-1}_{r}$$

Weighted example

Example

Input tree



Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

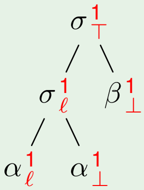
Note

Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Weighted example

Example

Non-zero weighted run



Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

Note

Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Weighted example

Example

Non-zero weighted run

$$\sigma^{1}_{\top}\big(\sigma^{1}_{\ell}(\alpha^{1}_{\ell}, \alpha^{1}_{\bot}), \beta^{1}_{\bot}\big) \qquad \sigma^{1}_{\top}\big(\sigma^{1}_{\ell}(\alpha^{1}_{\bot}, \alpha^{1}_{\ell}), \beta^{1}_{\bot}\big)$$

Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

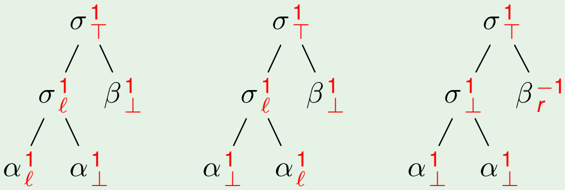
Note

Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Weighted example

Example

Non-zero weighted run



Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

Note

Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Weighted example

Example

Non-zero weighted run

$$\sigma^{1}_{\top}\big(\sigma^{1}_{\ell}(\alpha^{1}_{\ell},\ \alpha^{1}_{\perp}),\ \beta^{1}_{\perp}\big) \qquad \sigma^{1}_{\top}\big(\sigma^{1}_{\ell}(\alpha^{1}_{\perp},\ \alpha^{1}_{\ell}),\ \beta^{1}_{\perp}\big) \qquad \sigma^{1}_{\top}\big(\sigma^{1}_{\perp}(\alpha^{1}_{\perp},\ \alpha^{1}_{\perp}),\ \beta^{-1}_{r}\big)$$

Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

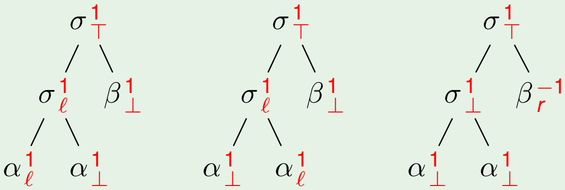
Note

Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Weighted example

Example

Non-zero weighted run



Recognized weighted language

$$A(\sigma(t_1, t_2)) = |t_1|_\alpha - |t_2|_\beta$$

Note

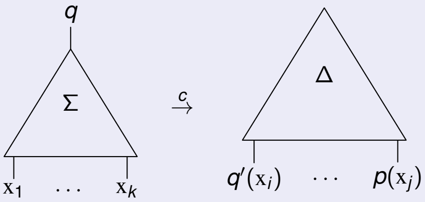
Support $\mathrm{supp}(A) = \{\sigma(t_1, t_2) \mid |t_1|_\alpha \neq |t_2|_\beta\}$ is not recognizable!
(i.e., language of non-zero weighted trees)

Contents

Syntax

Definition (Arnold, Dauchet 1976, Graehl, Knight 2004)

Weighted extended top-down tree transducer (WXTT)
$M = (Q, \Sigma, \Delta, I, R)$ with finitely many rules



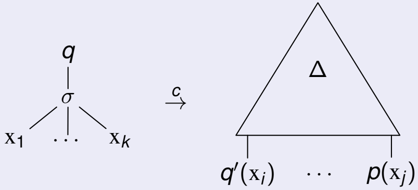
- states $q, q', p \in Q$
- variable indices $i, j \in \{1, \dots, k\}$

[Arnold, Dauchet: *Bi-transductions de forêts.* Proc. ICALP 1976]
[Graehl, Knight: *Training tree transducers.* Proc. NAACL 2004]

Syntax

Definition (Rounds 1970, Thatcher 1970)

Weighted top-down tree transducer (WTT) if all rules



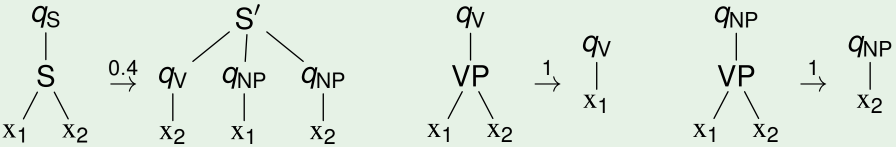
[Rounds: *Mappings and grammars on trees.* Math. Syst. Theory, 1970]

[Thatcher: *Generalized sequential machine maps.* J. Comput. Syst. Sci., 1970]

Semantics

Example

States $\{q_S, q_V, q_{NP}\}$ of which only q_S has non-zero initial weight

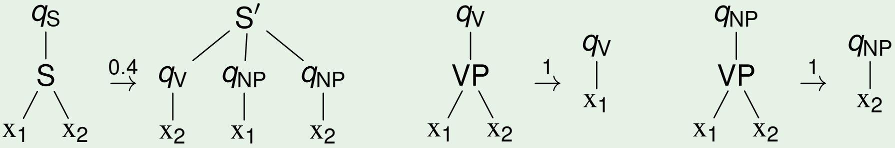


Derivation

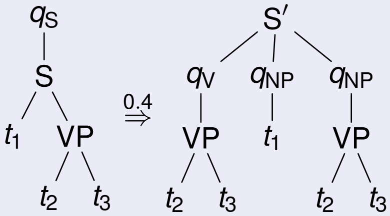
Semantics

Example

States $\{q_S, q_V, q_{NP}\}$ of which only q_S has non-zero initial weight



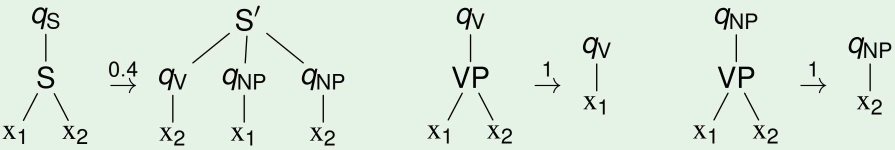
Derivation



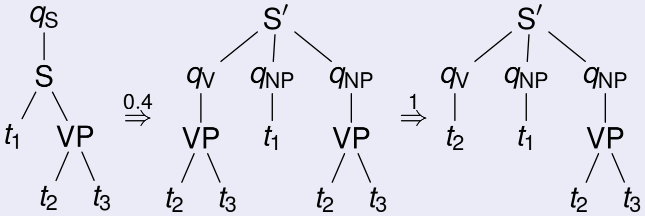
Semantics

Example

States $\{q_S, q_V, q_{NP}\}$ of which only q_S has non-zero initial weight



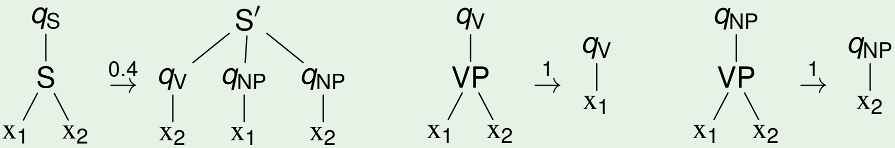
Derivation



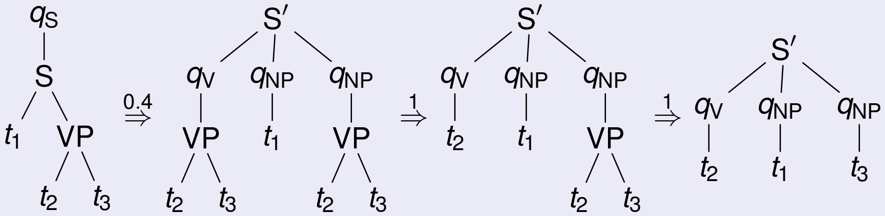
Semantics

Example

States $\{q_S, q_V, q_{NP}\}$ of which only q_S has non-zero initial weight



Derivation



Semantics

Definition

Computed transformation ($t \in T_\Sigma$ and $u \in T_\Delta$):

$$M(t, u) = \sum_{\substack{q \in Q \\ q(t) \overset{c_1}{\Rightarrow} \cdots \overset{c_n}{\Rightarrow} u \\ \text{left-most derivation}}} I(q) \cdot c_1 \cdot \ldots \cdot c_n$$

Contents

Preservation of recognizability

Definition (Forward application)

$M\colon T_\Sigma \times T_\Delta \to C$ and $A\colon T_\Sigma \to C$

$$[M(A)](u) = \sum_{t \in T_\Sigma} A(t) \cdot M(t, u)$$

Approach

1. Input (or output) product followed by projection
2. Direct construction

Preservation of recognizability

Definition (Forward application)

$M\colon T_\Sigma \times T_\Delta \to C$ and $A\colon T_\Sigma \to C$

$$[M(A)](u) = \sum_{t \in T_\Sigma} A(t) \cdot M(t, u)$$

Approach

1. Input (or output) product followed by projection
2. Direct construction

Input product + projection

Definition (Forward application)

$M\colon T_\Sigma \times T_\Delta \to C$ and $A\colon T_\Sigma \to C$

$$[M(A)](u) = \sum_{t \in T_\Sigma} A(t) \cdot M(t, u)$$

Definition (Input product)

Input product of WTA A and WXTT M is WXTT ${}_AM$ with

$${}_AM(t, u) = A(t) \cdot M(t, u)$$

Definition (Range projection)

WXTT M

$$[\mathrm{ran}(M)](u) = \sum_{t \in T_\Sigma} M(t, u)$$

Input product + projection

Definition (Forward application)

$M \colon T_\Sigma \times T_\Delta \to C$ and $A \colon T_\Sigma \to C$

$$[M(A)](u) = \sum_{t \in T_\Sigma} A(t) \cdot M(t, u)$$

Definition (Input product)

Input product of WTA A and WXTT M is WXTT ${}_A M$ with

$${}_A M(t, u) = A(t) \cdot M(t, u)$$

Definition (Range projection)

WXTT M

$$[\mathrm{ran}(M)](u) = \sum_{t \in T_\Sigma} M(t, u)$$

Input product + projection

Definition (Forward application)

$M\colon T_\Sigma \times T_\Delta \to C$ and $A\colon T_\Sigma \to C$

$$M(A) = \mathrm{ran}({}_A M)$$

Definition (Input product)

Input product of WTA A and WXTT M is WXTT ${}_A M$ with

$${}_A M(t, u) = A(t) \cdot M(t, u)$$

Definition (Range projection)

WXTT M

$$[\mathrm{ran}(M)](u) = \sum_{t \in T_\Sigma} M(t, u)$$

Product + projection

positive

- two simple generic constructions
  - BAR-HILLEL construction
  - projection
- reusable
- explain most known cases

Product + projection

positive

- two simple generic constructions
  - BAR-HILLEL construction
  - projection
- reusable
- explain most known cases

negative

- requirements of two constructions
- inefficiencies

Product + projection

Requirement

model	input product	range projection	output product	domain projection
In-XTOP	✓	✓	✓	✓
l-XTOP	✗	✓/✗	✓	✓
XTOP	✗	✗	✓	✗
In-MBOT				
l-MBOT				
MBOT				
In-STSSG				

Conclusion

Nondeletion essential for input product!

Product + projection

Requirement

model	input product	range projection	output product	domain projection
ln-XTOP	✓	✓	✓	✓
l-XTOP	✗	✓/✗	✓	✓
XTOP	✗	✗	✓	✗
ln-MBOT	✓	✗	✓	✓
l-MBOT	✓	✗	✓	✓
MBOT	✗	✗	✓	✗
ln-STSSG				

Conclusion

Nondeletion essential for input product!

Product + projection

Requirement

model	input product	range projection	output product	domain projection
In-XTOP	✓	✓	✓	✓
I-XTOP	✗	✓/✗	✓	✓
XTOP	✗	✗	✓	✗
In-MBOT	✓	✗	✓	✓
I-MBOT	✓	✗	✓	✓
MBOT	✗	✗	✓	✗
In-STSSG	✓	✗	✓	✗

Conclusion

Nondeletion essential for input product!

Product + projection

Requirement

model	input product		range projection	output product		domain projection
ln-XTOP	✓	(✓)	✓	✓	(✓)	✓
l-XTOP	✗	(✓/✗)	✓/✗	✓	(✓)	✓
XTOP	✗	(✗)	✗	✓	(✗)	✗
ln-MBOT	✓	(✗)	✗	✓	(✓)	✓
l-MBOT	✓	(✗)	✗	✓	(✓)	✓
MBOT	✗	(✗)	✗	✓	(✗)	✗
ln-STSSG	✓	(✗)	✗	✓	(✗)	✗

Conclusion

Nondeletion essential for input product!

Product + projection

Requirement

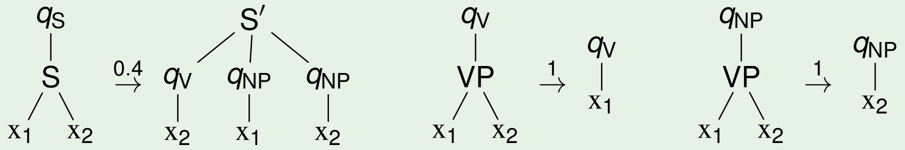
model	input product	range projection	output product	domain projection
ln-XTOP	✓	✓	✓	✓
l-XTOP	✗	✓/✗	✓	✓
XTOP	✗	✗	✓	✗
ln-MBOT	✓	✗	✓	✓
l-MBOT	✓	✗	✓	✓
MBOT	✗	✗	✓	✗
ln-STSSG	✓	✗	✓	✗

Conclusion

Nondeletion essential for input product!

Nondeletion

Example



nondeleting   linear   linear

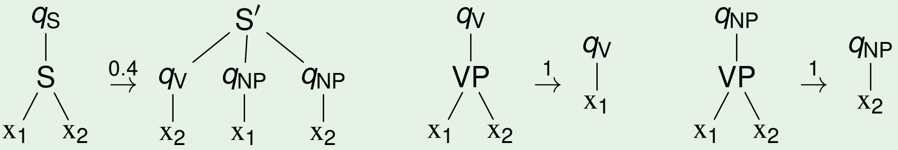
Definition

WXTT M is

- nondeleting if $\mathrm{var}(l) = \mathrm{var}(r)$ for all rules $l \to r$
- linear if no variable appears twice in r for all rules $l \to r$

Nondeletion

Example



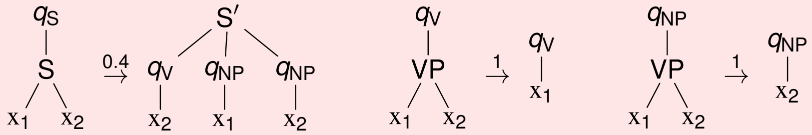
nondeleting deletes x_2 deletes x_1

Definition

- all-copies nondeleting = nondeleting
  = every copy of an input subtree is fully explored
- some-copy nondeleting
  = one copy of each input subtree is fully explored

Nondeletion

Example

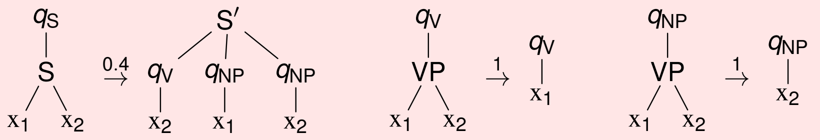


is not some-copy nondeleting

Example (Derivation)

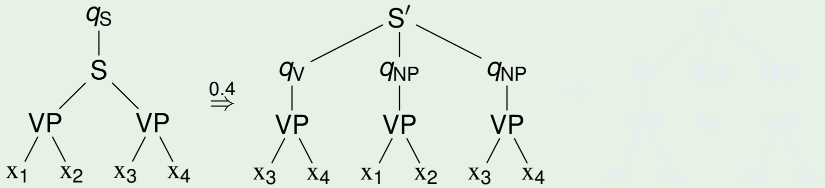
Nondeletion

Example



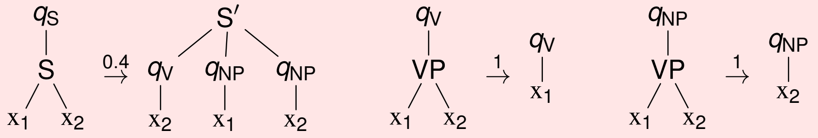
is not some-copy nondeleting

Example (Derivation)



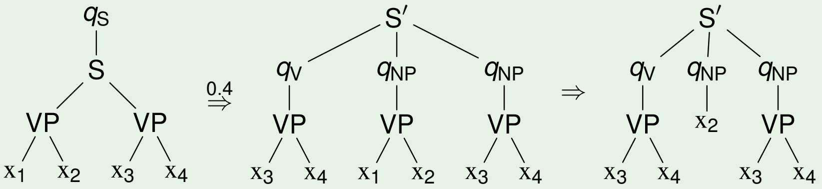
Nondeletion

Example



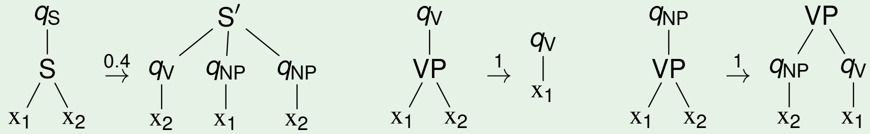
is not some-copy nondeleting

Example (Derivation)



Nondeletion

Example

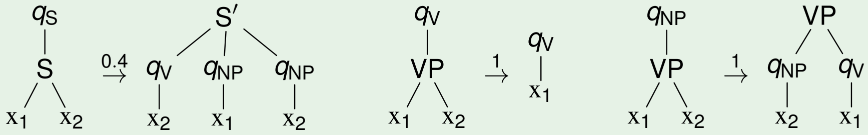


can be some-copy nondeleting

Example (Derivation)

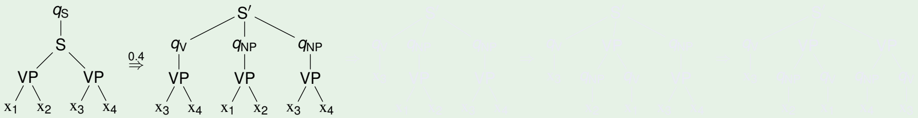
Nondeletion

Example



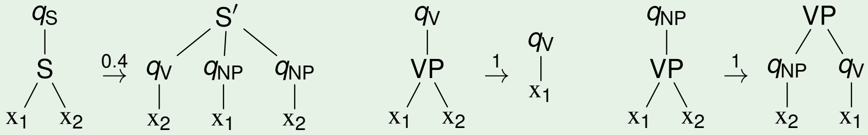
can be some-copy nondeleting

Example (Derivation)



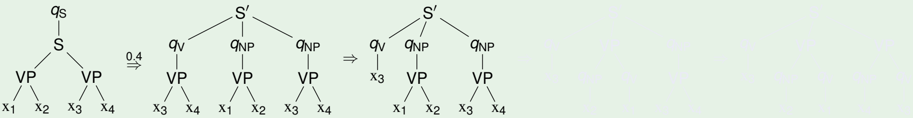
Nondeletion

Example



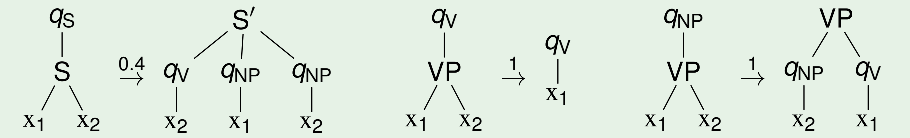
can be some-copy nondeleting

Example (Derivation)



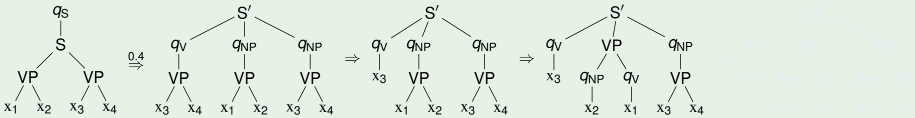
Nondeletion

Example



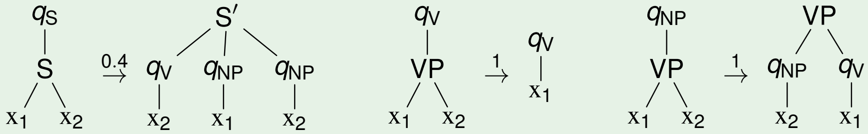
can be some-copy nondeleting

Example (Derivation)



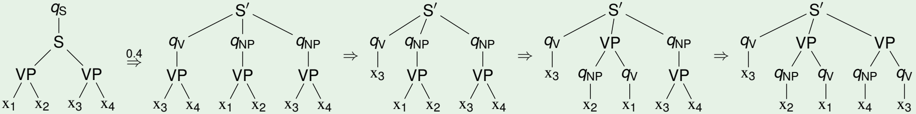
Nondeletion

Example



can be some-copy nondeleting

Example (Derivation)

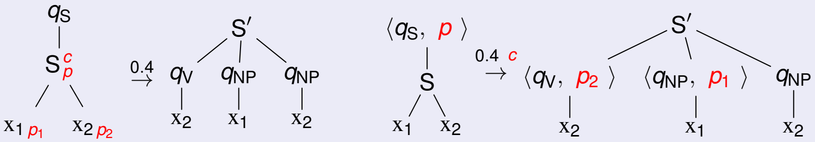


Scenario 1

Theorem (Engelfriet 1977)

For nondeleting WXTT M and WTA A we can construct ${}_A M$

Proof.



original rules

constructed rule

- for original nondeleting rules construct new rules
- mark one state for each variable; one possibility
- $x_2\,a \quad x_1\,b \quad x_2\,d \;\rightarrow\; \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_2\,d$

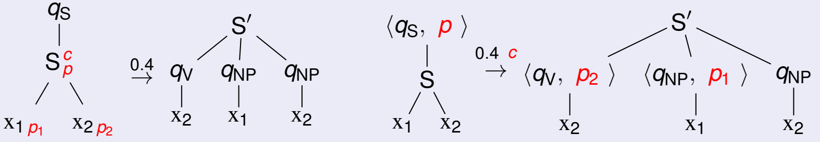
□

Scenario 1

Theorem (ENGELFRIET 1977)

For nondeleting WXTT M and WTA A we can construct ${}_AM$

Proof.



original rules

constructed rule

- for original nondeleting rules construct new rules
- mark one state for each variable; one possibility
- $x_{2\,a} \quad x_{1\,b} \quad x_{2\,d} \quad \to \quad \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_{2\,d}$

□

Scenario 2

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over idempotent semiring we can construct ${}_AM$

Proof.

- for original nondeleting rules construct new rules
- mark one state for each variable; all possibilities

$$x_{2}{}_{a} \quad x_{1}{}_{b} \quad x_{2}{}_{d} \quad \to \quad \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_{2}{}_{d} \quad \Big| \quad x_{2}{}_{a} \quad \boxed{x_1}^{f}_{b} \quad \boxed{x_2}^{e}_{d}$$

- at least one exploration will succeed (somy-copy nondeletion)
- $aebfd + abfde = abdef$ if several succeed (idempotency)

□

Scenario 2

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over idempotent semiring we can construct ${}_AM$

Proof.

- for original nondeleting rules construct new rules
- mark one state for each variable; all possibilities

$$x_{2\,a} \quad x_{1\,b} \quad x_{2\,d} \quad \to \quad \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_{2\,d} \quad \Big| \quad x_{2\,a} \quad \boxed{x_1}^{f}_{b} \quad \boxed{x_2}^{e}_{d}$$

- at least one exploration will succeed (somy-copy nondeletion)
- $aebfd + abfde = abdef$ if several succeed (idempotency)

□

Scenario 2

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over idempotent semiring we can construct ${}_A M$

Proof.

- for original nondeleting rules construct new rules
- mark one state for each variable; all possibilities

$$x_{2\,a} \quad x_{1\,b} \quad x_{2\,d} \quad \to \quad \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_{2\,d} \quad \mid \quad x_{2\,a} \quad \boxed{x_1}^{f}_{b} \quad \boxed{x_2}^{e}_{d}$$

- at least one exploration will succeed (somy-copy nondeletion)
- *aebfd* $+$ *abfde* $=$ *abdef* if several succeed (idempotency)

$\square$

Scenario 3

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over ring we can construct ${}_A M$

Proof.

- for original nondeleting rules construct several new rules
- mark states according to elimination scheme
- $x_{2a} \quad x_{1b} \quad x_{2d} \quad \to$

$$\boxed{x_2}^{e}_{a} \; \boxed{x_1}^{f}_{b} \; x_{2d} \;\Big|\; x_{2a} \; \boxed{x_1}^{f}_{b} \; \boxed{x_2}^{e}_{d} \;\Big|\; \boxed{x_2}^{e}_{a} \; \boxed{x_1}^{f}_{b} \; \boxed{x_2}^{-1}_{d}$$

- at least one exploration will succeed

$\square$

Scenario 3

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over ring we can construct $_A M$

Proof.

- for original nondeleting rules construct several new rules
- mark states according to elimination scheme
- $x_{2\,a} \quad x_{1\,b} \quad x_{2\,d} \quad \to$

$$\boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad x_{2\,d} \quad \Big| \quad x_{2\,a} \quad \boxed{x_1}^{f}_{b} \quad \boxed{x_2}^{e}_{d} \quad \Big| \quad \boxed{x_2}^{e}_{a} \quad \boxed{x_1}^{f}_{b} \quad \boxed{x_2}^{-1}_{d}$$

- at least one exploration will succeed

□

Scenario 3

Theorem ($\sim$ 2010)

For some-copy nondeleting WXTT M and WTA A over ring we can construct ${}_A M$

Proof.

- if several succeed, then

$$\boxed{x_2}_a^e \quad \boxed{x_1}_b^f \quad x_{2\,d} \quad \Big| \quad x_{2\,a} \quad \boxed{x_1}_b^f \quad \boxed{x_2}_d^e \quad \Big| \quad \boxed{x_2}_a^e \quad \boxed{x_1}_b^f \quad \boxed{x_2}_d^{-1}$$

$aebfd$	0	0
0	$abfde$	0
$aebfd$	$abfde$	$-aebfd$

$\square$

Elimination schemes

Question

Do elimination schemes exist?

Answer

	001	010	100	011	101	110	111	$\sum$
	$+$	$+$	$+$	$-$	$-$	$-$	$+$	
001	a	0	0	0	0	0	0	a
010	0	a	0	0	0	0	0	a
100	0	0	a	0	0	0	0	a
011	a	a	0	$-a$	0	0	0	a
101	a	0	a	0	$-a$	0	0	a
110	0	a	a	0	0	$-a$	0	a
111	a	a	a	$-a$	$-a$	$-a$	a	a

Elimination schemes

Question

Do elimination schemes exist?

Answer

	001	010	100	011	101	110	111	$\sum$
	$+$	$+$	$+$	$-$	$-$	$-$	$+$	
001	a	0	0	0	0	0	0	a
010	0	a	0	0	0	0	0	a
100	0	0	a	0	0	0	0	a
011	a	a	0	$-a$	0	0	0	a
101	a	0	a	0	$-a$	0	0	a
110	0	a	a	0	0	$-a$	0	a
111	a	a	a	$-a$	$-a$	$-a$	a	a

Elimination schemes

Question

Do elimination schemes exist?

Answer

	001	010	100	011	101	110	111	$\sum$
	$+$	$+$	$+$	$-$	$-$	$-$	$+$	
001	a	0	0	0	0	0	0	a
010	0	a	0	0	0	0	0	a
100	0	0	a	0	0	0	0	a
011	a	a	0	$-a$	0	0	0	a
101	a	0	a	0	$-a$	0	0	a
110	0	a	a	0	0	$-a$	0	a
111	a	a	a	$-a$	$-a$	$-a$	a	a

Direct construction

Applicability

- here only l-XTOP (product + projection fails)

Failure

- input product fails
  because it cannot attach weights to deleted subtrees
- but range projection disregards input trees

Solution

- assign aggregate weight to transitions deleting subtrees

Direct construction

Applicability

- here only l-XTOP (product + projection fails)

Failure

- input product fails
  because it cannot attach weights to deleted subtrees
- but range projection disregards input trees

Solution

- assign aggregate weight to transitions deleting subtrees

Direct construction

Applicability

- here only l-XTOP (product + projection fails)

Failure

- input product fails
  because it cannot attach weights to deleted subtrees
- but range projection disregards input trees

Solution

- assign aggregate weight to transitions deleting subtrees

Direct construction

Applicability

- here only l-XTOP (product + projection fails)

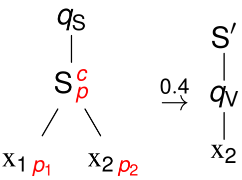
Failure

- input product fails
  because it cannot attach weights to deleted subtrees
- but range projection disregards input trees

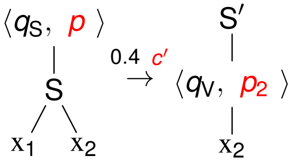
Solution

- assign aggregate weight to transitions deleting subtrees

Bonus scenario



original rules



constructed rule

where $c' = c \cdot \mathrm{in}(p_1)$

Inside weight of p

$$\mathrm{in}(p) = \sum_{\substack{t \in T_\Sigma \\ r \text{ run on } t \\ \mathrm{root}(r) = p}} \mathrm{wt}(r)$$

Bonus scenario

Theorem

For linear WXTT M and WTA A we can construct ${}_{A}M$
if inside weights of A can be computed

Computation of inside weights

- trivial in BOOLEAN semiring
- typically simple in extremal semirings (VITERBI algorithms)
- possible in $\mathbb{N}$ (deciding finiteness of support)

→ possible in many interesting cases

- approximation possible for $\mathbb{R}$ (NEWTON method)

Bonus scenario

Theorem

For linear WXTT M and WTA A we can construct ${}_A M$
if inside weights of A can be computed

Computation of inside weights

- trivial in BOOLEAN semiring
- typically simple in extremal semirings (VITERBI algorithms)
- possible in $\mathbb{N}$ (deciding finiteness of support)

→ possible in many interesting cases

- approximation possible for $\mathbb{R}$ (NEWTON method)

Bonus scenario

Theorem

For linear WXTT M and WTA A we can construct ${}_A M$
if inside weights of A can be computed

Computation of inside weights

- trivial in BOOLEAN semiring
- typically simple in extremal semirings (VITERBI algorithms)
- possible in $\mathbb{N}$ (deciding finiteness of support)

$\rightarrow$ possible in many interesting cases

- approximation possible for $\mathbb{R}$ (NEWTON method)

Contents

Overview

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗ (✓)	✓
XTOP	✗	✗ (✓)
ln-MBOT	✗	✓
l-MBOT	✗	✓(✓)
MBOT	✗	✗ (✓)
ln-STSSG	✗	✗

Limitation

- no coverage of unweighted failures
- only backward application of XTOP!
  (same phenomenon for MBOT)

Overview

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗ (✓)	✓
XTOP	✗	✗ (✓)
ln-MBOT	✗	✓
l-MBOT	✗	✓(✓)
MBOT	✗	✗ (✓)
ln-STSSG	✗	✗

Limitation

- no coverage of unweighted failures
- only backward application of XTOP!
  (same phenomenon for MBOT)

Overview

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗ (✓)	✓
XTOP	✗	✗ (✓)
ln-MBOT	✗	✓
l-MBOT	✗	✓(✓)
MBOT	✗	✗ (✓)
ln-STSSG	✗	✗

Limitation

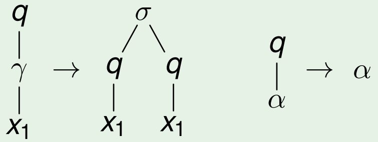
- no coverage of unweighted failures
- only backward application of XTOP!
  (same phenomenon for MBOT)

Counterexample

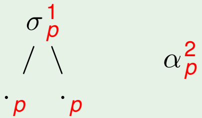
WXTT M

WTA A

Example



Example

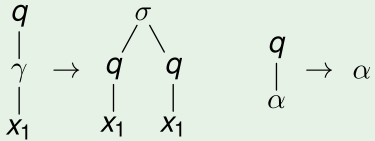


Counterexample

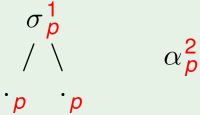
WXTT M

WTA A

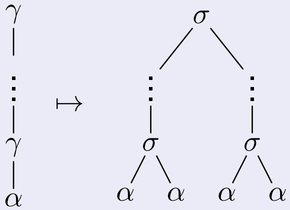
Example



Example



Transformation

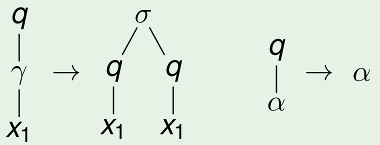


Counterexample

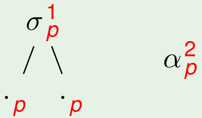
WXTT M

WTA A

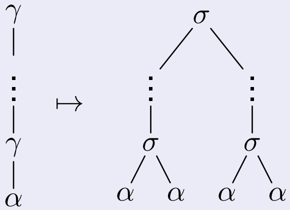
Example



Example



Transformation

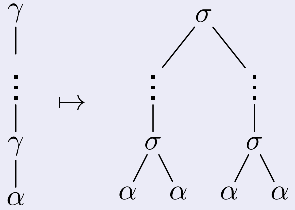


Weighted tree language

$$A(u) = 2^{|u|_\alpha}$$

Counterexample

Transformation M



$$|u|_\alpha = 2^{|t|_\gamma}$$

Weighted tree language A

$$A(u) = 2^{|u|_\alpha}$$

Backward application

$$[M^{-1}(A)](t) = 2^{(2^{|t|_\gamma})}$$

Theorem

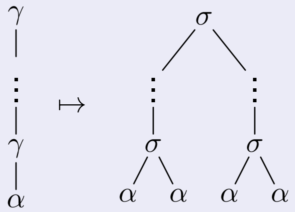
For every WTA A over $\mathbb{N}$ there exists $n \in \mathbb{N}$ such that $\forall t \in T_\Sigma$

$$A(t) \leq n^{|t|+1}$$

[Fülöp, ∼, Vogler: *Weighted extended tree transducers.* Fundam. Inform. 2011]

Counterexample

Transformation M



$|u|_\alpha = 2^{|t|_\gamma}$

Weighted tree language A

$$A(u) = 2^{|u|_\alpha}$$

Backward application

$$[M^{-1}(A)](t) = 2^{(2^{|t|_\gamma})}$$

Theorem

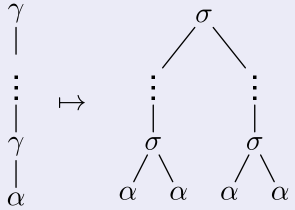
For every WTA A over $\mathbb{N}$ there exists $n \in \mathbb{N}$ such that $\forall t \in T_\Sigma$

$$A(t) \leq n^{|t|+1}$$

[FÜLÖP, ∼, VOGLER: *Weighted extended tree transducers.* Fundam. Inform. 2011]

Counterexample

Transformation M



$|u|_\alpha = 2^{|t|_\gamma}$

Weighted tree language A

$$A(u) = 2^{|u|_\alpha}$$

Backward application

$$[M^{-1}(A)](t) = 2^{(2^{|t|_\gamma})}$$

Theorem

For every WTA A over $\mathbb{N}$ there exists $n \in \mathbb{N}$ such that $\forall t \in T_\Sigma$

$$A(t) \leq n^{|t|+1}$$

[FÜLÖP, ∼, VOGLER: *Weighted extended tree transducers.* Fundam. Inform. 2011]

Overview

model	$M(L)$ recognizable?	$M^{-1}(L)$ recognizable?
ln-XTOP	✓	✓
l-XTOP	✓/✗ (✓)	✓
XTOP	✗	✗ (✓)
ln-MBOT	✗	✓
l-MBOT	✗	✓(✓)
MBOT	✗	✗ (✓)
ln-STSSG	✗	✗

That's all, folks!

Thank you for your attention!

References

- Arnold, Dauchet: *Bi-transductions de forêts.* Proc. ICALP 1976
- Berstel, Reutenauer: *Recognizable formal power series on trees.* Theor. Comput. Sci. 18, 1982
- Engelfriet: *Top-down tree transducers with regular look-ahead.* Math. Systems Theory 10, 1977
- Engelfriet, Fülöp, Vogler: *Bottom-up and top-down tree series transformations.* J. Autom. Lang. Comb. 7, 2002
- Fülöp, Maletti, Vogler: *Weighted extended tree transducers.* Fundam. Inform. 112, 2011
- Graehl, Knight: *Training tree transducers.* Proc. HLT-NAACL 2004
- Maletti, Satta: *Parsing algorithms based on tree automata.* Proc. IWPT 2009
- Maletti: *Input products for weighted extended top-down tree transducers.* Proc. DLT 2010
- Maletti: *An alternative to synchronous tree substitution grammars.* Natural Language Engineering 17, 2011
- May, Knight: Tiburon — *a weighted tree automata toolkit.* Proc. CIAA 2006
- Rounds: *Mappings and grammars on trees.* Math. Systems Theory 4, 1970
- Thatcher *Generalized^2 sequential machine maps.* J. Comput. System Sci. 4, 1970