Tree-Series-to-Tree-Series Transformations

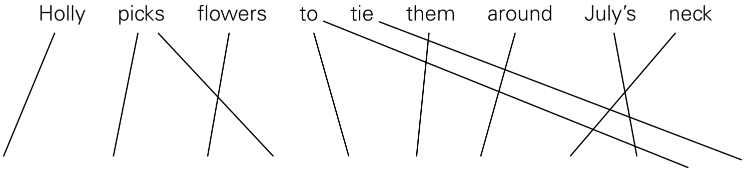
Andreas Maletti

International Computer Science Institute
Berkeley, CA, USA

maletti@icsi.berkeley.edu

San Francisco — July 21, 2008

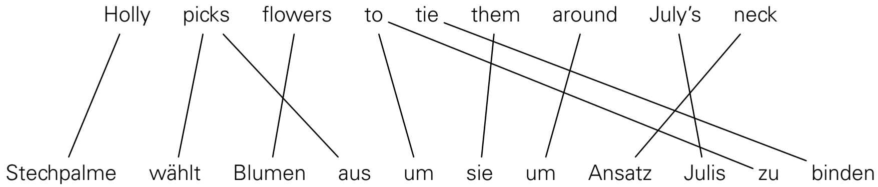
Classic Approach (simplified)



Steps

1. Select scheme
2. Translate words individually
3. Check sequence (bigrams, trigrams, etc.)

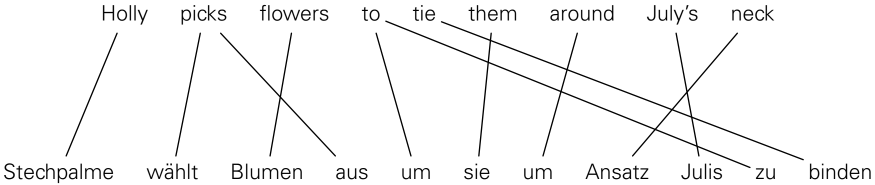
Classic Approach (simplified)



Steps

1. Select scheme
2. Translate words individually
3. Check sequence (bigrams, trigrams, etc.)

Classic Approach (simplified)

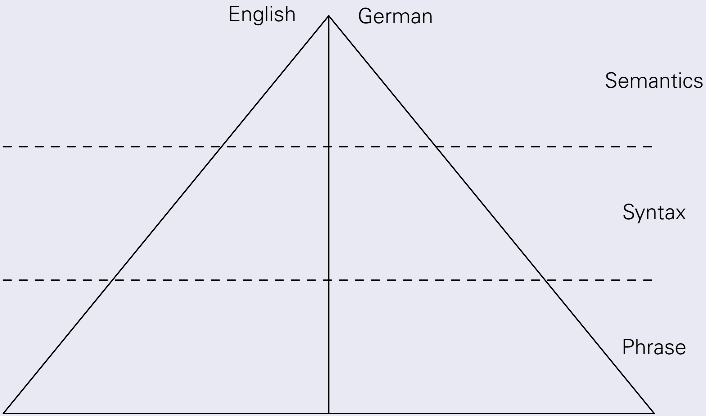


Steps

1. Select scheme
2. Translate words individually
3. Check sequence (bigrams, trigrams, etc.)

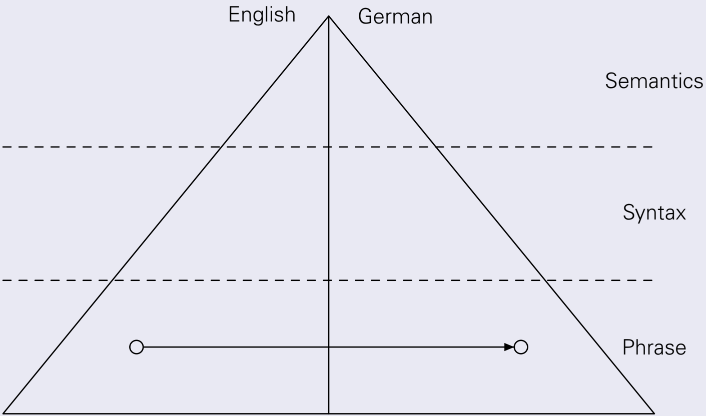
Syntax-based Approach

Overview



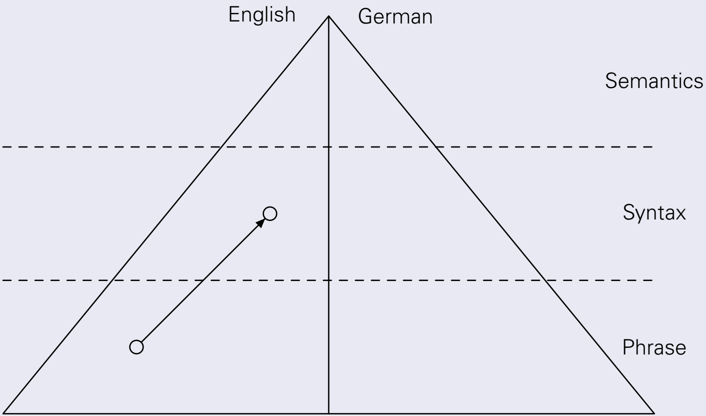
Syntax-based Approach

Overview



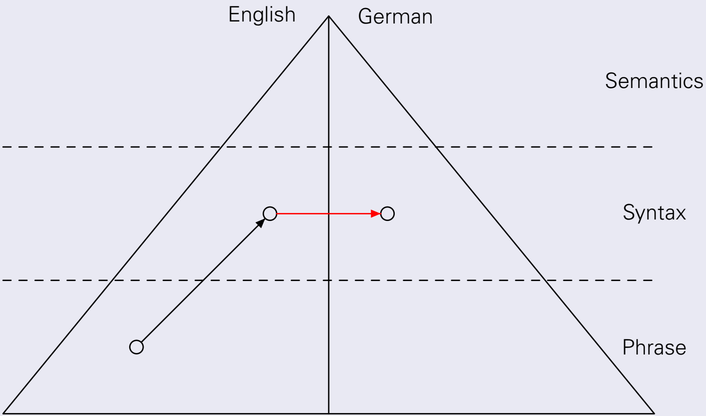
Syntax-based Approach

Overview



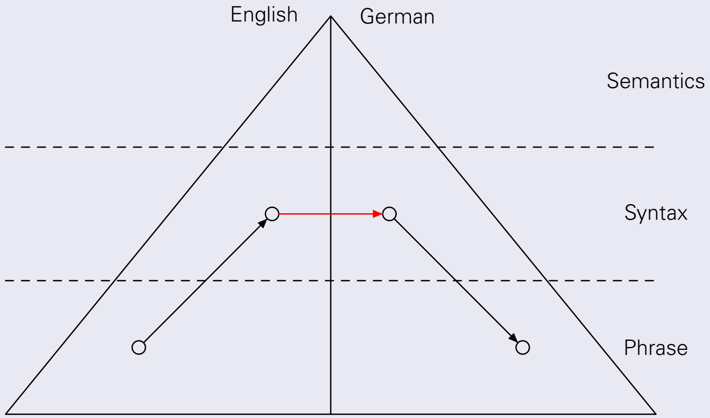
Syntax-based Approach

Overview

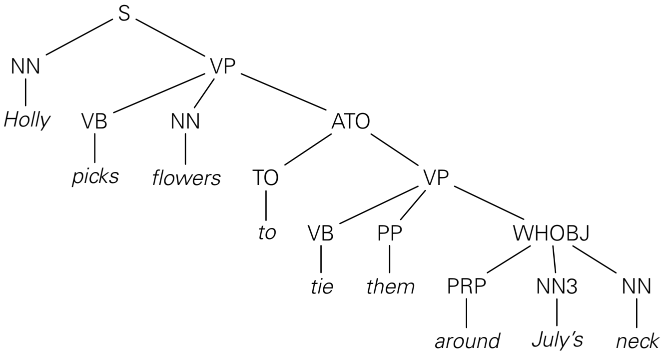


Syntax-based Approach

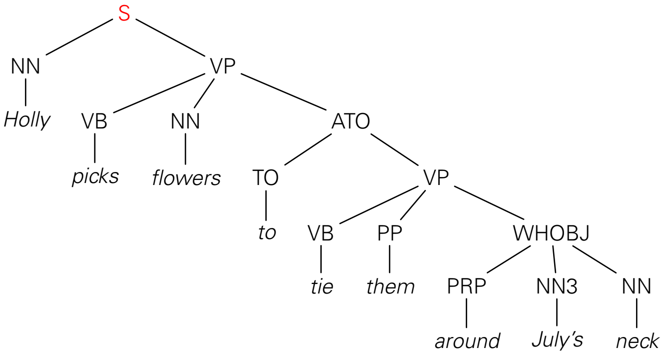
Overview



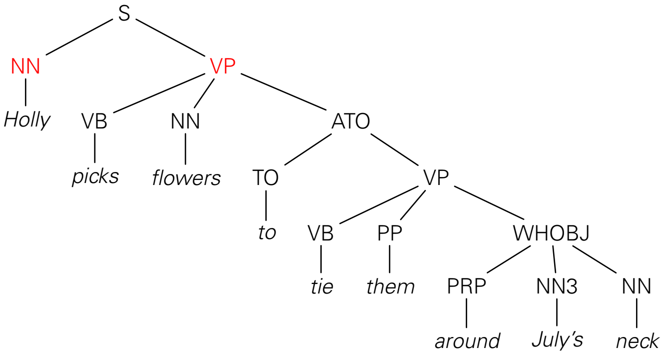
Syntactic Analysis



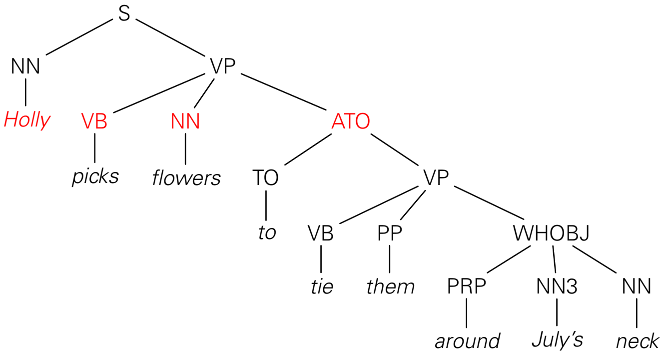
Translation



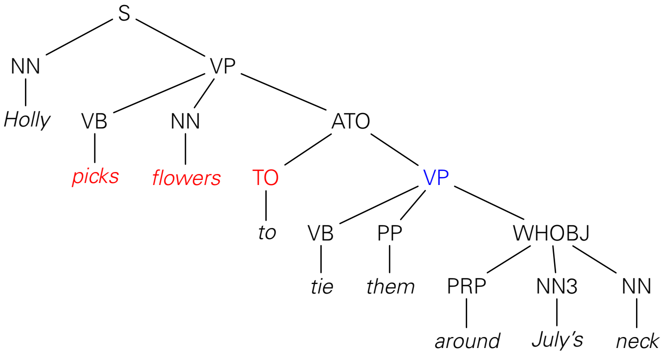
Translation



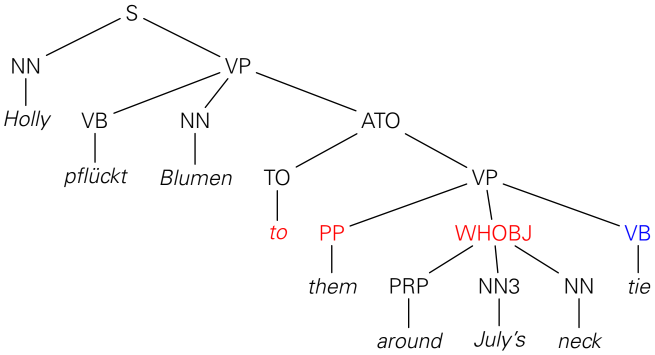
Translation



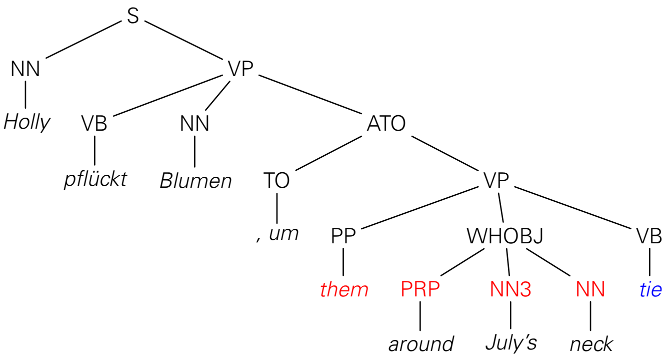
Translation



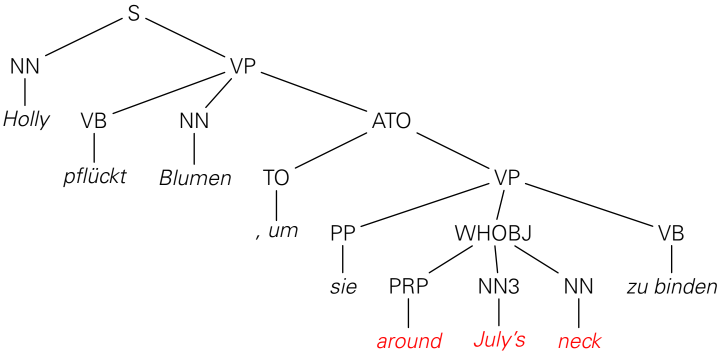
Translation



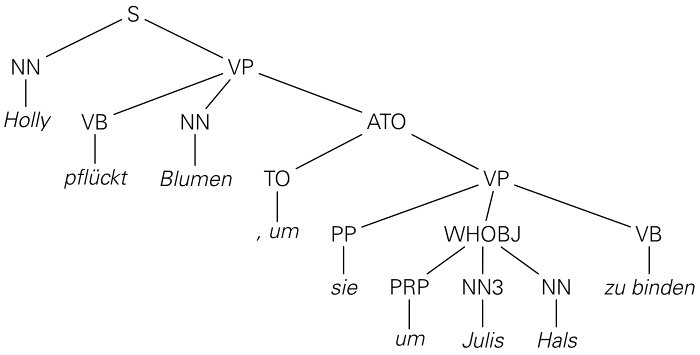
Translation



Translation



Translation



Adding Weights

- Hard decision (yes/no) replaced by soft decision
- Each translation then has a score
- Yields ranking on the alternatives

Problems

- Harder to train (but nowadays there is enough data)
- Destroys nice properties of the tree transducer framework

Addressed here

Well-definedness problem

Adding Weights

- Hard decision (yes/no) replaced by soft decision
- Each translation then has a score
- Yields ranking on the alternatives

Problems

- Harder to train (but nowadays there is enough data)
- Destroys nice properties of the tree transducer framework

Addressed here

Well-definedness problem

Adding Weights

- Hard decision (yes/no) replaced by soft decision
- Each translation then has a score
- Yields ranking on the alternatives

Problems

- Harder to train (but nowadays there is enough data)
- Destroys nice properties of the tree transducer framework

Addressed here

Well-definedness problem

Adding Weights

- Hard decision (yes/no) replaced by soft decision
- Each translation then has a score
- Yields ranking on the alternatives

Problems

- Harder to train (but nowadays there is enough data)
- Destroys nice properties of the tree transducer framework

Addressed here

Well-definedness problem

Table of Contents

Some Basics

- semiring $\mathcal{A} = (A, +, \cdot, 0, 1)$ is a "ring without subtraction"
- $A\langle\langle T_\Delta(X)\rangle\rangle$: set of all maps $T_\Delta(X) \to A$
- $A\langle T_\Delta(X)\rangle$: 0 almost everywhere maps of $A\langle\langle T_\Delta(X)\rangle\rangle$

Some Basics

- semiring $\mathcal{A} = (A, +, \cdot, 0, 1)$ is a "ring without subtraction"
- $A\langle\langle T_\Delta(X)\rangle\rangle$: set of all maps $T_\Delta(X) \to A$
- $A\langle T_\Delta(X)\rangle$: 0 almost everywhere maps of $A\langle\langle T_\Delta(X)\rangle\rangle$

Some Basics

- semiring $\mathcal{A} = (A, +, \cdot, 0, 1)$ is a “ring without subtraction”
- $A\langle\langle T_\Delta(X)\rangle\rangle$: set of all maps $T_\Delta(X) \to A$
- $A\langle T_\Delta(X)\rangle$: 0 almost everywhere maps of $A\langle\langle T_\Delta(X)\rangle\rangle$

Syntax

Definition

Tree series transducer is tuple $(Q, \Sigma, \Delta, \mathcal{A}, F, \mu)$

- Q finite set of states
- Σ and Δ ranked alphabets of input und output symbols
- $\mathcal{A} = (A, +, \cdot, 0, 1)$ commutative semiring
- $F \subseteq Q$ final states
- $\mu = (\mu_k)_{k \in \mathbb{N}}$ with $\mu_k \colon \Sigma_k \to A\langle\langle T_\Delta(X)\rangle\rangle^{Q \times Q(X_k)^*}$

such that

1. $\mu_k(\sigma)_{q,w} \in A\langle T_\Delta(X_{|w|})\rangle$
2. $\mu_k(\sigma)$ is $\widetilde{0}$ almost everywhere

Syntax

Definition

Tree series transducer is tuple $(Q, \Sigma, \Delta, \mathcal{A}, F, \mu)$

- Q finite set of states
- Σ and Δ ranked alphabets of input und output symbols
- $\mathcal{A} = (A, +, \cdot, 0, 1)$ commutative semiring
- $F \subseteq Q$ final states
- $\mu = (\mu_k)_{k \in \mathbb{N}}$ with $\mu_k : \Sigma_k \to A\langle\langle T_\Delta(X)\rangle\rangle^{Q \times Q(X_k)^*}$

such that

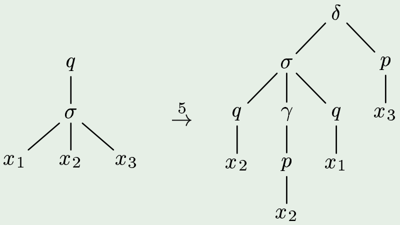
1. $\mu_k(\sigma)_{q,w} \in A\langle T_\Delta(X_{|w|})\rangle$
2. $\mu_k(\sigma)$ is $\widetilde{0}$ almost everywhere

Syntax: Top-down vs. Bottom-up

Definition

- top-down if $\mu_k(\sigma)_{q,w}$ is linear and nondeleting in $X_{|w|}$
- bottom-up if $\mu_k(\sigma)$ is nonzero only at $(q, q_1(x_1) \cdots q_k(x_k))$

Example (Top-down)



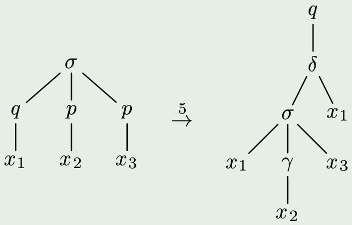
$$(\mu_3(\sigma)_{q,q(x_2)p(x_2)q(x_1)p(x_3)}, \delta(\sigma(x_1, \gamma(x_2), x_3), x_4)) = 5$$

Syntax: Top-down vs. Bottom-up

Definition

- top-down if $\mu_k(\sigma)_{q,w}$ is linear and nondeleting in $X_{|w|}$
- bottom-up if $\mu_k(\sigma)$ is nonzero only at $(q, q_1(x_1) \cdots q_k(x_k))$

Example (Bottom-up)



$$(\mu_3(\sigma)_{q,q(x_1)p(x_2)p(x_3)}, \delta(\sigma(x_1, \gamma(x_2), x_3), x_1)) = 5$$

Substitution of Tree Series

Definition

$\psi, \psi_1, \ldots, \psi_n \in A\langle\langle T_\Delta(X)\rangle\rangle$

$$\psi \leftarrow (\psi_1, \ldots, \psi_n) = \sum_{t, t_1, \ldots, t_n \in T_\Delta(X)} \left((\psi, t) \cdot \prod_{i=1}^{n} (\psi_i, t_i) \right) t[t_1, \ldots, t_n]$$

Semantics

Definition

Define tree evaluation as:

$$h_\mu(\sigma(t_1, \ldots, t_k))_q = \sum_{w \in Q(X_k)^*} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(t_{i_1})_{q_1}, \ldots, h_\mu(t_{i_n})_{q_n})$$

where $w = q_1(x_{i_1}) \cdots q_n(x_{i_n})$

Definition

Tree-to-tree-series transformation

$$\tau_M(t) = \sum_{q \in F} h_\mu(t)_q$$

Semantics

Definition

Define tree evaluation as:

$$h_\mu(\sigma(t_1, \dots, t_k))_q = \sum_{w \in Q(X_k)^*} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(t_{i_1})_{q_1}, \dots, h_\mu(t_{i_n})_{q_n})$$

where $w = q_1(x_{i_1}) \cdots q_n(x_{i_n})$

Definition

Tree-to-tree-series transformation

$$\tau_M(t) = \sum_{q \in F} h_\mu(t)_q$$

Semantics

Definition

Define tree evaluation as:

$$h_\mu(\sigma(t_1, \dots, t_k))_q = \sum_{w \in Q(X_k)^*} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(t_{i_1})_{q_1}, \dots, h_\mu(t_{i_n})_{q_n})$$

where $w = q_1(x_{i_1}) \cdots q_n(x_{i_n})$

Definition

Tree-series-to-tree-series transformation

$$\tau_M(\psi) = \sum_{t \in T_\Sigma} (\psi, t) \cdot \tau_M(t)$$

Well-definedness

Problem

When is $\sum_{t \in T_\Sigma} (\psi, t) \cdot \tau_M(t)$ well-defined?

Answer

Always in complete semirings.

Rebuke

Most complete semirings are unpractical.

Well-definedness

Problem

When is $\sum_{t \in T_\Sigma} (\psi, t) \cdot \tau_M(t)$ well-defined?

Answer

Always in complete semirings.

Rebuke

Most complete semirings are unpractical.

Well-definedness

Problem

When is $\sum_{t \in T_\Sigma} (\psi, t) \cdot \tau_M(t)$ well-defined?

Answer

Always in complete semirings.

Rebuke

Most complete semirings are unpractical.

Table of Contents

Our Approach

Answer

It is well-defined if every output tree can be produced only from finitely many input trees

Without weights

It is well-defined if $\tau_M^{-1}(u)$ is finite

Our Approach

Answer

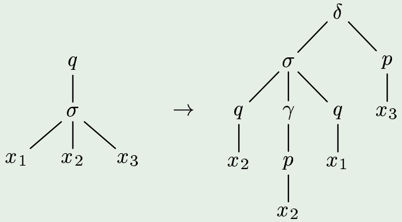
It is well-defined if every output tree can be produced only from finitely many input trees

Without weights

It is well-defined if $\tau_M^{-1}(u)$ is finite

Top-down Case

Example (Nondeleting)

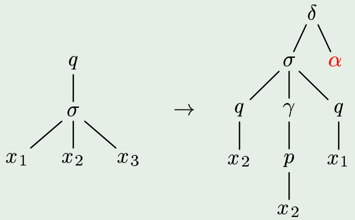


Theorem

If a trim top-down tree transducer is deleting, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Top-down Case

Example (Deleting)

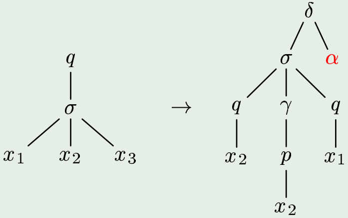


Theorem

If a trim top-down tree transducer is deleting, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Top-down Case

Example (Deleting)



Theorem

If a trim top-down tree transducer is deleting, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Top-down Case (cont'd)

Example (Replication)

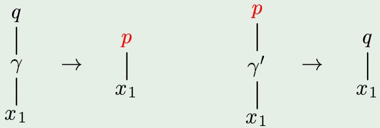
$$q(\gamma(x_1)) \to q(x_1)$$

Definition

Any state that can be reached from itself without producing output is called replicating.

Top-down Case (cont'd)

Example (Replication)



Definition

Any state that can be reached from itself without producing output is called replicating.

Top-down Case (cont'd)

Example (Replication)

$q(\gamma(x_1)) \to p(x_1)$ $\qquad$ $p(\gamma'(x_1)) \to q(x_1)$

Definition

Any state that can be reached from itself without producing output is called replicating.

Top-down Case (cont'd)

Theorem

For a trim top-down tree transducer M, TFAE:

(i) $\tau_M^{-1}(u)$ *is finite for all* $u \in T_\Delta$

(ii) M *is nondeleting and has no replicating states*

Proof.

Using a size argument for the output trees. □

Remarks

- (ii) characterizes well-definedness over positive semirings
- (ii) yields well-definedness of the ts-ts transformation in arbitrary semirings

Top-down Case (cont'd)

Theorem

For a trim top-down tree transducer M, TFAE:

(i) $\tau_M^{-1}(u)$ *is finite for all* $u \in T_\Delta$

(ii) M *is nondeleting and has no replicating states*

Proof.

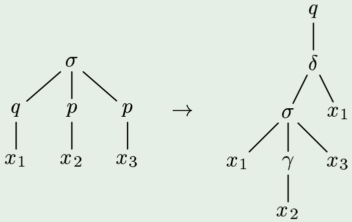
Using a size argument for the output trees. □

Remarks

- (ii) characterizes well-definedness over positive semirings
- (ii) yields well-definedness of the ts-ts transformation in arbitrary semirings

Bottom-up Case

Example (Nondeleting)

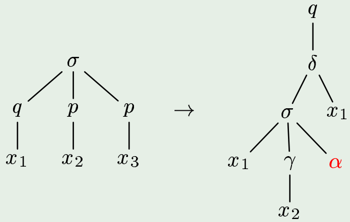


Theorem

If a trim bottom-up tree transducer deletes at a state that can accept infinitely many trees, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Bottom-up Case

Example (Deleting)

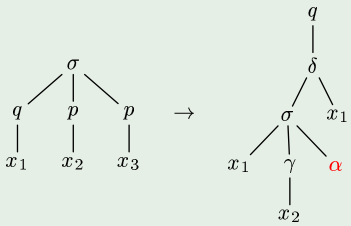


Theorem

If a trim bottom-up tree transducer deletes at a state that can accept infinitely many trees, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Bottom-up Case

Example (Deleting)

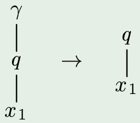


Theorem

If a trim bottom-up tree transducer deletes at a state that can accept infinitely many trees, then $\tau_M^{-1}(u)$ is infinite for some output tree u.

Bottom-up Case (con't)

Example (Replication)



Definition

- Replicating state defined as imagined
- Infinite state accepts infinitely many input trees

Bottom-up Case (con't)

Example (Replication)

$$\gamma(q(x_1)) \to p(x_1) \qquad \gamma'(p(x_1)) \to q(x_1)$$

Definition

- Replicating state defined as imagined
- Infinite state accepts infinitely many input trees

Bottom-up Case (con't)

Example (Replication)

$\gamma(q(x_1)) \to p(x_1)$ $\qquad$ $\gamma'(p(x_1)) \to q(x_1)$

Definition

- Replicating state defined as imagined
- Infinite state accepts infinitely many input trees

Bottom-up Case (cont'd)

Theorem

For a trim bottom-up tree transducer M, TFAE:

(i) $\tau_M^{-1}(u)$ *is finite for all* $u \in T_\Delta$

(ii) *M does not delete at an infinite state and has no replicating states*

Proof.

Using again a size argument for the output trees. □

Remarks

- (ii) characterizes well-definedness over positive semirings
- (ii) yields well-definedness of the ts-ts transformation in arbitrary semirings

Bottom-up Case (cont'd)

Theorem

For a trim bottom-up tree transducer M, TFAE:

(i) $\tau_M^{-1}(u)$ *is finite for all* $u \in T_\Delta$

(ii) *M does not delete at an infinite state and has no replicating states*

Proof.

Using again a size argument for the output trees. □

Remarks

- (ii) characterizes well-definedness over positive semirings
- (ii) yields well-definedness of the ts-ts transformation in arbitrary semirings

References

- Engelfriet, Fülöp, Vogler: Bottom-up and Top-down Tree Series Transformations. *J. Automata, Languages and Combinatorics* 7, 2002
- Gécseg, Steinby: *Tree Automata*. Akadémiai Kiadó, Budapest 1984
- Engelfriet: Bottom-up and top-down tree transformations — a comparison. *Math. Systems Theory* 9(3), 1975

Thank you for your attention!