Tree Transducers

Niko Paltzer

Seminar Formal Grammars WS 06/07
Advisor: Marco Kuhlmann

Programming Systems Lab

Outline

Trees & Tree Transducers

Derivations & State-sequences

Copying Normal Form

Intercalation Lemma

Back
Close

Trees

Definition

Trees are defined over a ranked alphabet Σ.

$$t \in T_\Sigma \text{ if } \begin{cases} t \in \Sigma_0 \\ t = \sigma(t_1 \ldots t_n) \quad \sigma \in \Sigma_n, t_i \in T_\Sigma \end{cases}$$

Trees

Definition

Trees are defined over a ranked alphabet Σ.

$$t \in T_\Sigma \text{ if } \begin{cases} t \in \Sigma_0 \\ t = \sigma(t_1 \ldots t_n) & \sigma \in \Sigma_n, t_i \in T_\Sigma \end{cases}$$

Example

$$\Sigma_2 = \{\sigma\}$$
$$\Sigma_1 = \{\tau\}$$
$$\Sigma_0 = \{\delta\}$$

$$\Sigma = \Sigma_2 \cup \Sigma_1 \cup \Sigma_0$$

Trees

Definition

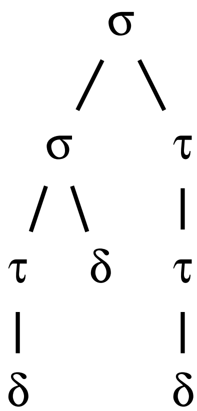
Trees are defined over a ranked alphabet Σ.

$$t \in T_\Sigma \text{ if } \begin{cases} t \in \Sigma_0 \\ t = \sigma(t_1 \ldots t_n) & \sigma \in \Sigma_n, t_i \in T_\Sigma \end{cases}$$

Example

$\Sigma_2 = \{\sigma\}$
$\Sigma_1 = \{\tau\}$
$\Sigma_0 = \{\delta\}$

$\Sigma = \Sigma_2 \cup \Sigma_1 \cup \Sigma_0$



Back
Close

Tree Transducers

Definition

$M = (Q, \Sigma, \Delta, q_0, R)$

Q	finite set of states
Σ	ranked input alphabet
Δ	ranked output alphabet
$q_0 \in Q$	initial state
R	finite set of rules

Tree Transducers

Definition

$M = (Q, \Sigma, \Delta, q_0, R)$

Q	finite set of states
Σ	ranked input alphabet
Δ	ranked output alphabet
$q_0 \in Q$	initial state
R	finite set of rules

Rule Format

$q(\sigma(x_1 \ldots x_n)) \to \tau(q_1(x_{i_1}) \ldots q_k(x_{i_k}))$

x_j variables
$\sigma \in \Sigma_n$
$\tau \in \Delta_k$
$q, q_1, \ldots, q_k \in Q$

Tree Transducers

Definition

$M = (Q, \Sigma, \Delta, q_0, R)$

Q	finite set of states
Σ	ranked input alphabet
Δ	ranked output alphabet
$q_0 \in Q$	initial state
R	finite set of rules

Rule Format

$q(\sigma(x_1 \ldots x_n)) \to \tau(q_1(x_{i_1}) \ldots q_k(x_{i_k}))$

x_j variables
$\sigma \in \Sigma_n$
$\tau \in \Delta_k$
$q, q_1, \ldots, q_k \in Q$

M deterministic $\Leftrightarrow$ left-hand sides are disjoint

Example

Tree Transducer

$$M = (\{q\}, \{\sigma, \tau, \delta\}, \{\beta, \gamma\}, q, R)$$

$$\begin{aligned} q(\sigma(xy)) &\rightarrow \beta(q(x)q(y)) \\ q(\tau(x)) &\rightarrow \beta(q(x)q(x)) \\ q(\delta) &\rightarrow \gamma \end{aligned}$$

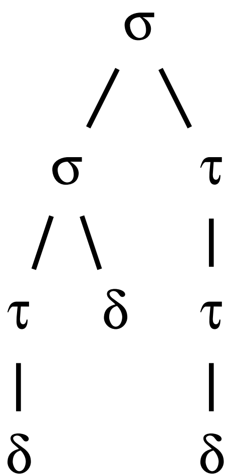
Example

Tree Transducer

$$M = (\{q\}, \{\sigma, \tau, \delta\}, \{\beta, \gamma\}, q, R)$$

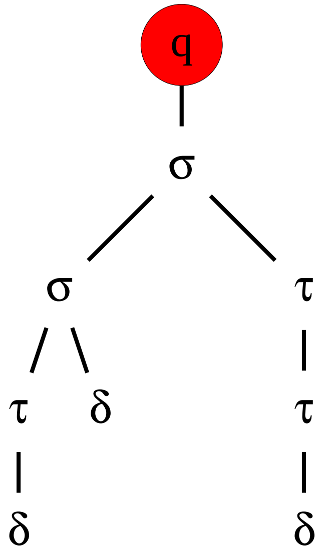
$$\begin{aligned} q(\sigma(xy)) &\rightarrow \beta(q(x)q(y)) \\ q(\tau(x)) &\rightarrow \beta(q(x)q(x)) \\ q(\delta) &\rightarrow \gamma \end{aligned}$$

Input Tree



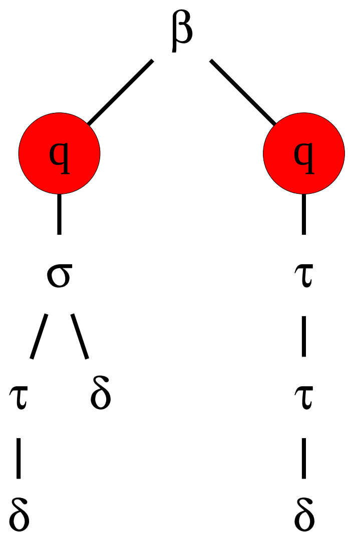
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



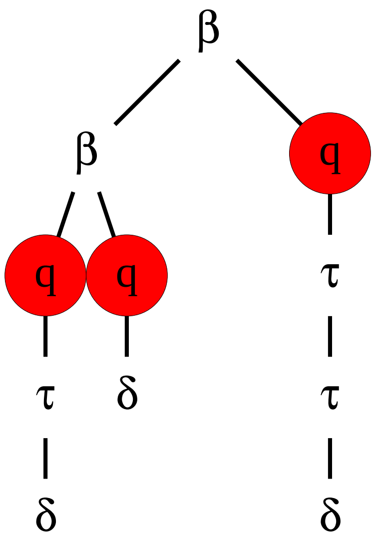
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



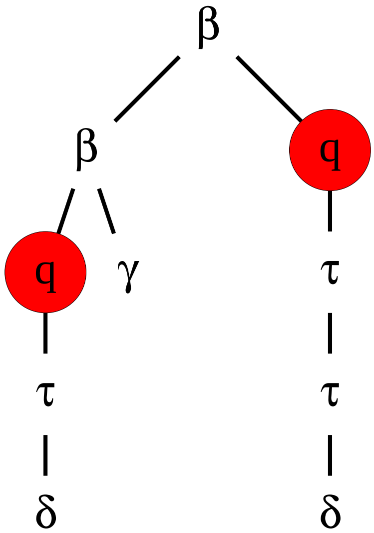
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



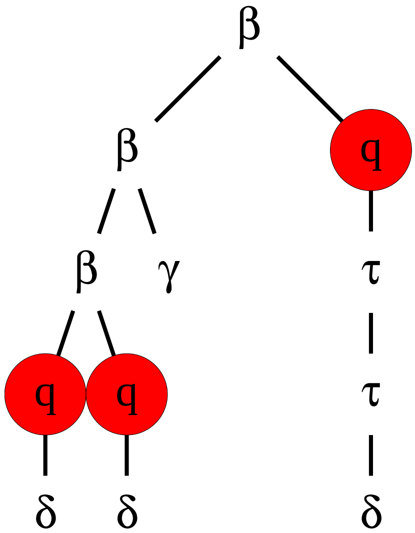
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



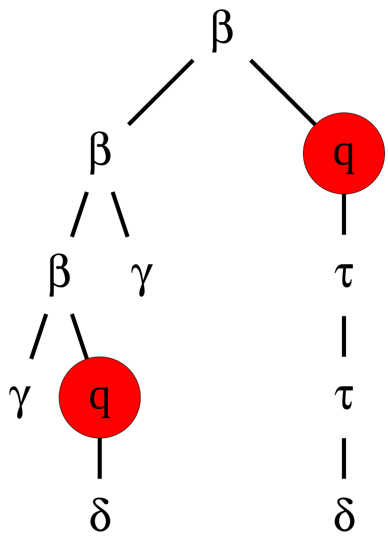
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



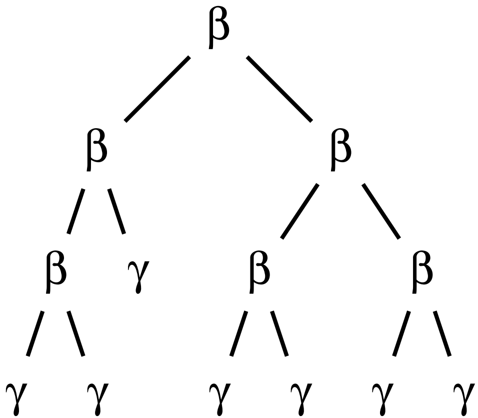
Example - Derivation

$$q(\sigma(xy)) \rightarrow \beta(q(x)q(y))$$
$$q(\tau(x)) \rightarrow \beta(q(x)q(x))$$
$$q(\delta) \rightarrow \gamma$$



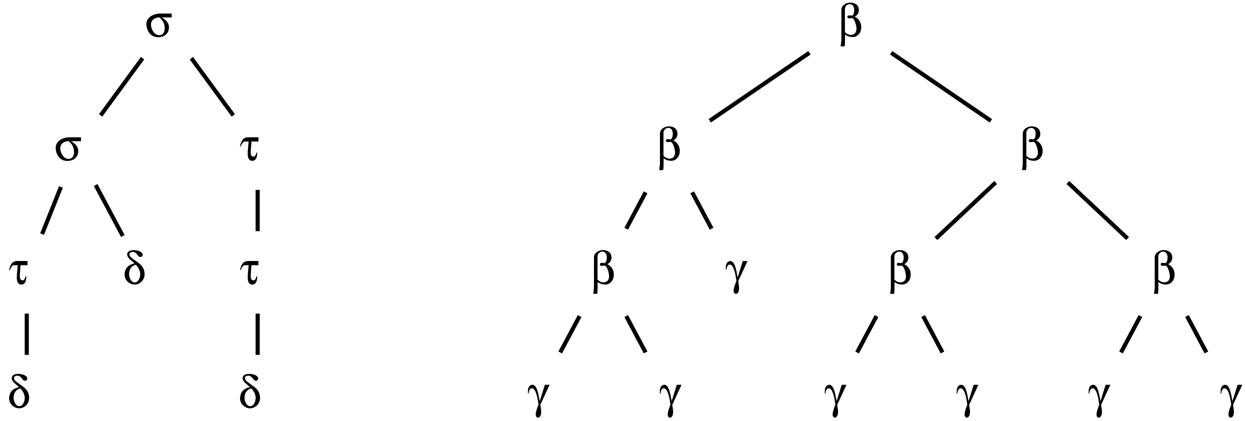
Example - Derivation

$$
\begin{aligned}
q(\sigma(xy)) &\rightarrow \beta(q(x)q(y)) \\
q(\tau(x)) &\rightarrow \beta(q(x)q(x)) \\
q(\delta) &\rightarrow \gamma
\end{aligned}
$$

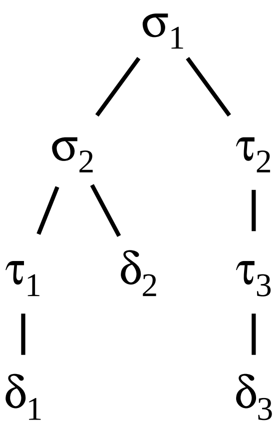


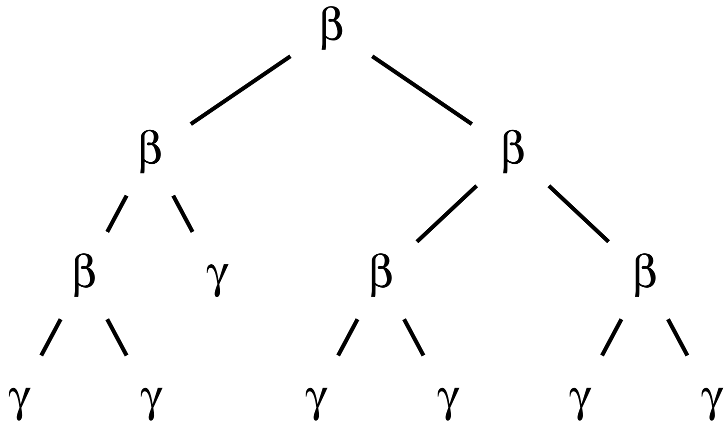
Back
Close

State-sequence

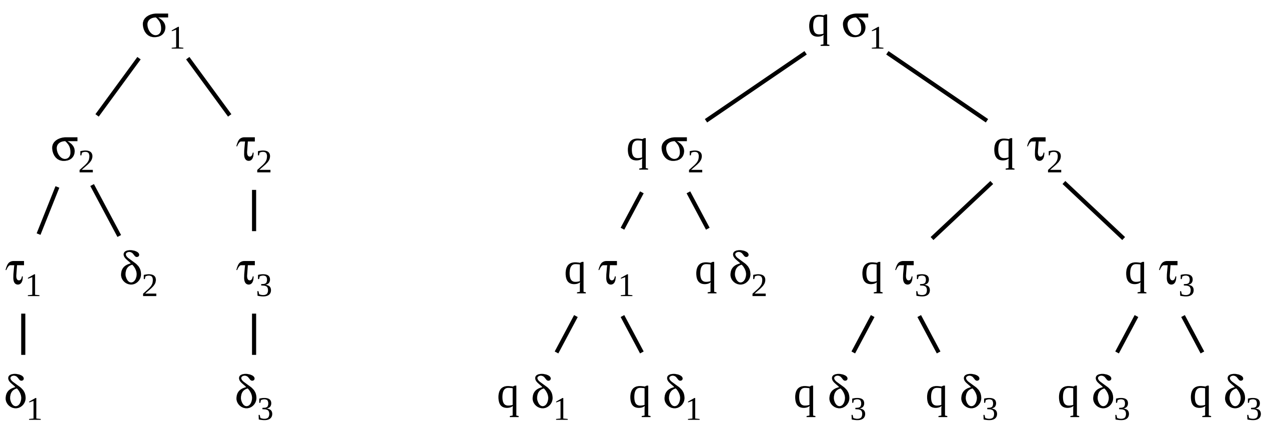


State-sequence

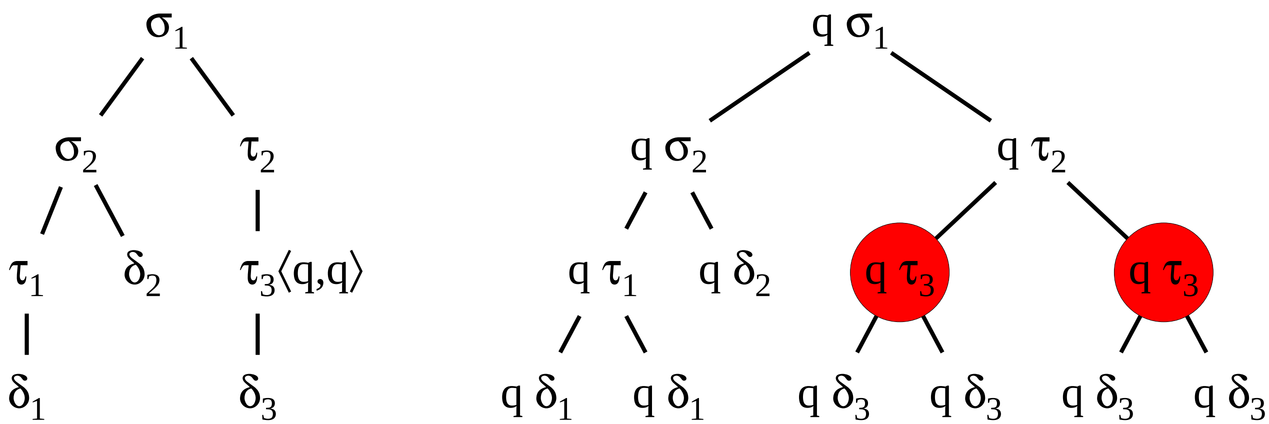




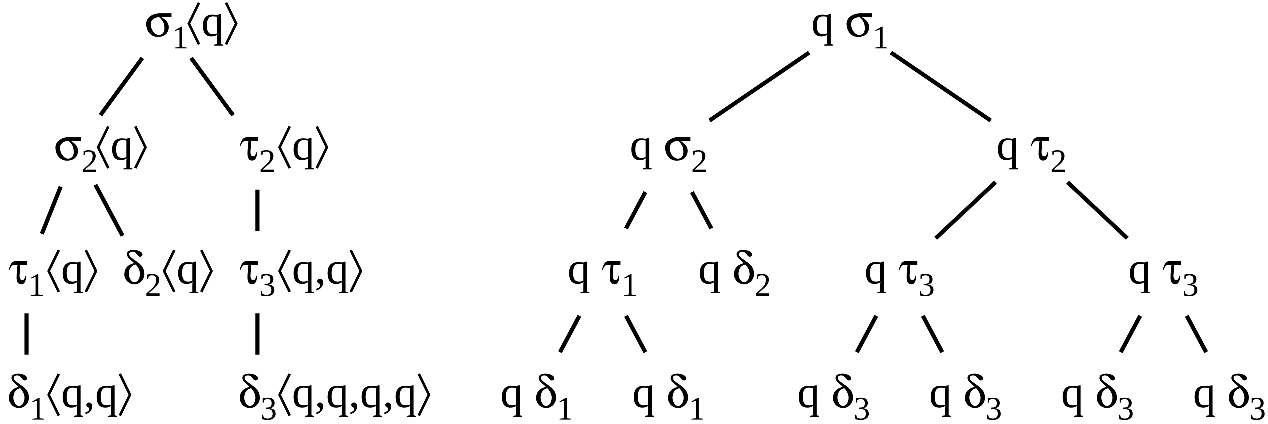
State-sequence



State-sequence



State-sequence



Copying

Copying-bound

A derivation has copying-bound k if all state-sequences have at most length k.

A transducer M has copying-bound k if for all possible output trees there exists a derivation with copying-bound k.

Copying

Copying-bound

A derivation has copying-bound k if all state-sequences have at most length k.

A transducer M has copying-bound k if for all possible output trees there exists a derivation with copying-bound k.

Finite Copying

A transducer M is finite copying if it has some copying-bound $k \in \mathbb{N}$.

$DT_{fc(k)}$ is the class of deterministic tree transducers with copying-bound k.

Copying-bound

Dynamic

The definition of the copying-bound is based on dynamic properties.

Copying-bound

Dynamic

The definition of the copying-bound is based on dynamic properties.

Static

There exists a possibility to check the copying-bound statically.

Copying Normal Form

Idea

If each state-sequence consists of different states, the number of states is an upper copying-bound.

This is only possible if the transducer is finite copying.

Copying Normal Form

Idea

If each state-sequence consists of different states, the number of states is an upper copying-bound.

This is only possible if the transducer is finite copying.

Algorithm (found in [vV96])

1. encode state-sequences into new states
2. copy transducer rules for additionally introduced states
3. repeat till fixpoint is reached

Copying Normal Form - Example

$$q(\tau(x)) \to \beta(r(x)s(x))$$
$$q(\delta) \to \gamma$$

$$r(\tau(x)) \to \beta(s(x)s(x))$$
$$r(\delta) \to \gamma$$

$$s(\tau(x)) \to \alpha(s(x))$$
$$s(\delta) \to \gamma$$

Copying Normal Form - Example

$q(\tau(x)) \rightarrow \beta(r(x)s(x))$
$q(\delta) \rightarrow \gamma$

$r(\tau(x)) \rightarrow \beta(s(x)s(x))$
$r(\delta) \rightarrow \gamma$

$s(\tau(x)) \rightarrow \alpha(s(x))$
$s(\delta) \rightarrow \gamma$

$\langle \overline{q} \rangle(\tau(x)) \rightarrow \beta(\langle \overline{r} s \rangle(x) \langle r \overline{s} \rangle(x))$

Copying Normal Form - Example

$q(\tau(x)) \to \beta(r(x)s(x))$ $\qquad$ $\langle\overline{q}\rangle(\tau(x)) \to \beta(\langle\overline{r}s\rangle(x)\langle r\overline{s}\rangle(x))$

$q(\delta) \to \gamma$

$r(\tau(x)) \to \beta(s(x)s(x))$ $\qquad$ $\langle\overline{r}s\rangle$

$r(\delta) \to \gamma$

$s(\tau(x)) \to \alpha(s(x))$ $\qquad$ $\langle r\overline{s}\rangle$

$s(\delta) \to \gamma$

Copying Normal Form - Example

$$q(\tau(x)) \to \beta(r(x)s(x)) \qquad \langle\overline{q}\rangle(\tau(x)) \to \beta(\langle\overline{r}s\rangle(x)\langle r\overline{s}\rangle(x))$$
$$q(\delta) \to \gamma \qquad \langle\overline{q}\rangle(\delta) \to \gamma$$

$$r(\tau(x)) \to \beta(s(x)s(x)) \qquad \langle\overline{r}s\rangle$$
$$r(\delta) \to \gamma$$

$$s(\tau(x)) \to \alpha(s(x)) \qquad \langle r\overline{s}\rangle$$
$$s(\delta) \to \gamma$$

Copying Normal Form - Example

$$q(\tau(x)) \longrightarrow \beta(r(x)s(x))$$
$$q(\delta) \longrightarrow \gamma$$

$$r(\tau(x)) \longrightarrow \beta(s(x)s(x))$$
$$r(\delta) \longrightarrow \gamma$$

$$s(\tau(x)) \longrightarrow \alpha(s(x))$$
$$s(\delta) \longrightarrow \gamma$$

$$\langle\overline{q}\rangle(\tau(x)) \longrightarrow \beta(\langle\overline{r}s\rangle(x)\langle r\overline{s}\rangle(x))$$
$$\langle\overline{q}\rangle(\delta) \longrightarrow \gamma$$

$$\langle\overline{r}s\rangle(\tau(x)) \longrightarrow \beta(\langle\overline{s}ss\rangle(x)\langle s\overline{s}s\rangle(x))$$

$$\langle r\overline{s}\rangle$$

Copying Normal Form - Example

$q(\tau(x)) \rightarrow \beta(r(x)s(x))$
$q(\delta) \rightarrow \gamma$

$r(\tau(x)) \rightarrow \beta(s(x)s(x))$
$r(\delta) \rightarrow \gamma$

$s(\tau(x)) \rightarrow \alpha(s(x))$
$s(\delta) \rightarrow \gamma$

$\langle \overline{q} \rangle(\tau(x)) \rightarrow \beta(\langle \overline{r}s \rangle(x)\langle r\overline{s} \rangle(x))$
$\langle \overline{q} \rangle(\delta) \rightarrow \gamma$

$\langle \overline{r}s \rangle(\tau(x)) \rightarrow \beta(\langle \overline{s}ss \rangle(x)\langle s\overline{s}s \rangle(x))$

$\langle r\overline{s} \rangle$

$\langle \overline{s}ss \rangle$

$\langle s\overline{s}s \rangle$

Back
Close

Copying Normal Form - Example

$$q(\tau(x)) \longrightarrow \beta(r(x)s(x))$$
$$q(\delta) \longrightarrow \gamma$$

$$r(\tau(x)) \longrightarrow \beta(s(x)s(x))$$
$$r(\delta) \longrightarrow \gamma$$

$$s(\tau(x)) \longrightarrow \alpha(s(x))$$
$$s(\delta) \longrightarrow \gamma$$

$$\langle \overline{q} \rangle(\tau(x)) \longrightarrow \beta(\langle \overline{r} s \rangle(x)\langle r \overline{s} \rangle(x))$$
$$\langle \overline{q} \rangle(\delta) \longrightarrow \gamma$$

$$\langle \overline{r} s \rangle(\tau(x)) \longrightarrow \beta(\langle \overline{s} s s \rangle(x)\langle s \overline{s} s \rangle(x))$$
$$\langle \overline{r} s \rangle(\delta) \longrightarrow \gamma$$

$$\langle r \overline{s} \rangle$$

$$\langle \overline{s} s s \rangle$$

$$\langle s \overline{s} s \rangle$$

Copying Normal Form - Example

$$q(\tau(x)) \longrightarrow \beta(r(x)s(x))$$
$$q(\delta) \longrightarrow \gamma$$

$$r(\tau(x)) \longrightarrow \beta(s(x)s(x))$$
$$r(\delta) \longrightarrow \gamma$$

$$s(\tau(x)) \longrightarrow \alpha(s(x))$$
$$s(\delta) \longrightarrow \gamma$$

$$\langle\overline{q}\rangle(\tau(x)) \longrightarrow \beta(\langle\overline{r}s\rangle(x)\langle r\overline{s}\rangle(x))$$
$$\langle\overline{q}\rangle(\delta) \longrightarrow \gamma$$

$$\langle\overline{r}s\rangle(\tau(x)) \longrightarrow \beta(\langle\overline{s}ss\rangle(x)\langle s\overline{s}s\rangle(x))$$
$$\langle\overline{r}s\rangle(\delta) \longrightarrow \gamma$$

$$\langle r\overline{s}\rangle(\tau(x)) \longrightarrow \alpha(\langle ss\overline{s}\rangle(x))$$

$$\langle\overline{s}ss\rangle$$

$$\langle s\overline{s}s\rangle$$

Back
Close

Copying Normal Form - Example

$$q(\tau(x)) \to \beta(r(x)s(x))$$
$$q(\delta) \to \gamma$$

$$r(\tau(x)) \to \beta(s(x)s(x))$$
$$r(\delta) \to \gamma$$

$$s(\tau(x)) \to \alpha(s(x))$$
$$s(\delta) \to \gamma$$

$$\langle \overline{q} \rangle(\tau(x)) \to \beta(\langle \overline{r}s \rangle(x)\langle r\overline{s} \rangle(x))$$
$$\langle \overline{q} \rangle(\delta) \to \gamma$$

$$\langle \overline{r}s \rangle(\tau(x)) \to \beta(\langle \overline{s}ss \rangle(x)\langle s\overline{s}s \rangle(x))$$
$$\langle \overline{r}s \rangle(\delta) \to \gamma$$

$$\langle r\overline{s} \rangle(\tau(x)) \to \alpha(\langle ss\overline{s} \rangle(x))$$

$$\langle \overline{s}ss \rangle$$

$$\langle s\overline{s}s \rangle$$

$$\langle ss\overline{s} \rangle$$

Copying Normal Form - Example

$$q(\tau(x)) \longrightarrow \beta(r(x)s(x))$$
$$q(\delta) \longrightarrow \gamma$$

$$r(\tau(x)) \longrightarrow \beta(s(x)s(x))$$
$$r(\delta) \longrightarrow \gamma$$

$$s(\tau(x)) \longrightarrow \alpha(s(x))$$
$$s(\delta) \longrightarrow \gamma$$

$$\langle \overline{q} \rangle(\tau(x)) \longrightarrow \beta(\langle \overline{r} s \rangle(x)\langle r \overline{s} \rangle(x))$$
$$\langle \overline{q} \rangle(\delta) \longrightarrow \gamma$$

$$\langle \overline{r} s \rangle(\tau(x)) \longrightarrow \beta(\langle \overline{s} s s \rangle(x)\langle s \overline{s} s \rangle(x))$$
$$\langle \overline{r} s \rangle(\delta) \longrightarrow \gamma$$

$$\langle r \overline{s} \rangle(\tau(x)) \longrightarrow \alpha(\langle s s \overline{s} \rangle(x))$$
$$\langle r \overline{s} \rangle(\delta) \longrightarrow \gamma$$

$$\langle \overline{s} s s \rangle(\tau(x)) \longrightarrow \alpha(\langle \overline{s} s s \rangle(x))$$
$$\langle \overline{s} s s \rangle(\delta) \longrightarrow \gamma$$

$$\langle s \overline{s} s \rangle(\tau(x)) \longrightarrow \alpha(\langle s \overline{s} s \rangle(x))$$
$$\langle s \overline{s} s \rangle(\delta) \longrightarrow \gamma$$

$$\langle s s \overline{s} \rangle(\tau(x)) \longrightarrow \alpha(\langle s s \overline{s} \rangle(x))$$
$$\langle s s \overline{s} \rangle(\delta) \longrightarrow \gamma$$

Yield Language

Definition

Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a tree transducer.

Then

$$y\mathcal{L}(M) = \{yield(t') \mid q_0(t) \Rightarrow^* t' \text{ for some } t \in T_\Sigma\}$$

is the *yield language* of M.

Yield Language

Definition

Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a tree transducer.

Then

$$y\mathcal{L}(M) = \{yield(t') \mid q_0(t) \Rightarrow^* t' \text{ for some } t \in T_\Sigma\}$$

is the *yield language* of M.

And

$$y\mathcal{L}(\mathcal{C}) = \{y\mathcal{L}(M) \mid M \in \mathcal{C}\}$$

is the *set of yield languages* for some class of tree transducers $\mathcal{C}$.

Yield Language

Definition

Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a tree transducer.

Then

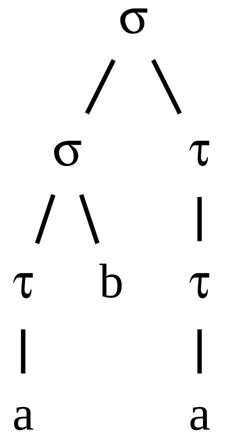
$$y\mathcal{L}(M) = \{yield(t') \mid q_0(t) \Rightarrow^* t' \text{ for some } t \in T_\Sigma\}$$

is the *yield language* of M.

And

$$y\mathcal{L}(\mathcal{C}) = \{y\mathcal{L}(M) \mid M \in \mathcal{C}\}$$

is the *set of yield languages* for some class of tree transducers $\mathcal{C}$.



aba

Language Hierarchy

Claim

$y\mathcal{L}(DT_{fc(k-1)}) \subsetneq y\mathcal{L}(DT_{fc(k)})$ for $k > 1$

Language Hierarchy

Claim

$y\mathcal{L}(DT_{fc(k-1)}) \subsetneq y\mathcal{L}(DT_{fc(k)})$ for $k > 1$

Proof

Let $L_k = \{a_1^n a_2^n \dots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$.

Show $L_k \in y\mathcal{L}(DT_{fc(k)}) \backslash y\mathcal{L}(DT_{fc(k-1)})$, i.e.

show $L_k \in y\mathcal{L}(DT_{fc(k)})$ by construction and

$L_k \notin y\mathcal{L}(DT_{fc(k-1)})$ with the intercalation lemma.

Intercalation Lemma

This is a weak form of the intercalation lemma from [ERS80]:

Let $k > 0$ be an integer.

$\forall L \in y\mathcal{L}(DT_{fc(k)}).\ \exists p.\ \forall z \in L$ with $|z| \geq p$.

Intercalation Lemma

This is a weak form of the intercalation lemma from [ERS80]:

Let $k > 0$ be an integer.

$\forall L \in y\mathcal{L}(DT_{fc(k)}).\ \exists p.\ \forall z \in L$ with $|z| \geq p$.
$\exists z_1, \ldots, z_s$ and $x_1, \ldots, x_{s+1}$ $(1 \leq s \leq k)$ such that

1. $z = x_1 z_1 x_2 z_2 \ldots x_s z_s x_{s+1}$

Intercalation Lemma

This is a weak form of the intercalation lemma from [ERS80]:

Let $k > 0$ be an integer.

$\forall L \in y\mathcal{L}(DT_{fc(k)}).\ \exists p.\ \forall z \in L$ with $|z| \geq p.$
$\exists z_1, \ldots, z_s$ and $x_1, \ldots, x_{s+1}$ $(1 \leq s \leq k)$ such that

1. $z = x_1 z_1 x_2 z_2 \ldots x_s z_s x_{s+1}$
2. $0 < |z_i| \leq p$ for $1 \leq i \leq s$

Intercalation Lemma

This is a weak form of the intercalation lemma from [ERS80]:

Let $k > 0$ be an integer.

$\forall L \in y\mathcal{L}(DT_{fc(k)}).\ \exists p.\ \forall z \in L$ with $|z| \geq p.$
$\exists z_1, \ldots, z_s$ and $x_1, \ldots, x_{s+1}$ $(1 \leq s \leq k)$ such that

1. $z = x_1 z_1 x_2 z_2 \ldots x_s z_s x_{s+1}$
2. $0 < |z_i| \leq p$ for $1 \leq i \leq s$
3. $\forall n \in \mathbb{N}.\ \exists v_1, \ldots, v_s$ such that
   (a) $v = x_1 v_1 x_2 v_2 \ldots x_s v_s x_{s+1}$
   (b) $v \in L$
   (c) $|v| \geq n$
   (d) $alph(v_i) = alph(z_i)$ for $1 \leq i \leq s$

Intercalation Lemma - Application

Claim

Let $L_k = \{a_1^n a_2^n \ldots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$, then $L_k \notin y\mathcal{L}(DT_{fc(k-1)})$

Proof by Contradiction

Assume $L_k \in y\mathcal{L}(DT_{fc(k-1)})$. Let $z = a_1^p a_2^p \ldots a_{2k}^p \in L_k$.

Then $z = x_1 z_1 x_2 z_2 \ldots x_s z_s x_{s+1}$ with $s \leq k-1$ and $0 < |z_i| \leq p$.

Intercalation Lemma - Application

Claim

Let $L_k = \{a_1^n a_2^n \dots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$, then $L_k \notin y\mathcal{L}(DT_{fc(k-1)})$

Proof by Contradiction

Assume $L_k \in y\mathcal{L}(DT_{fc(k-1)})$. Let $z = a_1^p a_2^p \dots a_{2k}^p \in L_k$.

Then $z = x_1 z_1 x_2 z_2 \dots x_s z_s x_{s+1}$ with $s \leq k-1$ and $0 < |z_i| \leq p$.

$\Rightarrow$ At least 1 and at most $2k-2$ different a_i occur in $z_1, \dots, z_s$.

Intercalation Lemma - Application

Claim

Let $L_k = \{a_1^n a_2^n \dots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$, then $L_k \notin y\mathcal{L}(DT_{fc(k-1)})$

Proof by Contradiction

Assume $L_k \in y\mathcal{L}(DT_{fc(k-1)})$. Let $z = a_1^p a_2^p \dots a_{2k}^p \in L_k$.

Then $z = x_1 z_1 x_2 z_2 \dots x_s z_s x_{s+1}$ with $s \leq k-1$ and $0 < |z_i| \leq p$.

$\Rightarrow$ At least 1 and at most $2k-2$ different a_i occur in $z_1, \dots, z_s$.

$\Rightarrow$ There is some a_j that does not occur in any z_i.

Intercalation Lemma - Application

Claim

Let $L_k = \{a_1^n a_2^n \ldots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$, then $L_k \notin y\mathcal{L}(DT_{fc(k-1)})$

Proof by Contradiction

Assume $L_k \in y\mathcal{L}(DT_{fc(k-1)})$. Let $z = a_1^p a_2^p \ldots a_{2k}^p \in L_k$.

Then $z = x_1 z_1 x_2 z_2 \ldots x_s z_s x_{s+1}$ with $s \leq k-1$ and $0 < |z_i| \leq p$.

$\Rightarrow$ At least 1 and at most $2k-2$ different a_i occur in $z_1, \ldots, z_s$.

$\Rightarrow$ There is some a_j that does not occur in any z_i.

$\Rightarrow$ When "intercalating", the number of a_j remains p.

Summary

Tree Transducers:

Derivations & State-sequences

Copying Normal Form:

Introduction & Algorithm

Intercalation Lemma:

Introduction & Application

References

[ERS80] Joost Engelfriet, Grzegorz Rozenberg, and Giora Slutzki. *Tree Transducers, L Systems, and Two-Way Machines*. Journal of Computer and System Sciences 20, 1980.

[Man04] Sebastian Maneth. *Models of Tree Translation*. IPA Dissertation Series, 2004.

[Per75] C. Raymond Perrault. *Intercalation theorems for tree transducer languages*. STOC '75: Proceedings of seventh annual ACM symposium on Theory of computing, 1975.

[vV96] Niké van Vugt. *Generalized Context-Free Grammars*. Master's Thesis, Universiteit Leiden, 1996.

Language Hierarchy - Proof

Claim

Let $L_k = \{a_1^n a_2^n \dots a_{2k}^n \mid n \in \mathbb{N}\}$ for $k > 1$, then $L_k \in y\mathcal{L}(DT_{fc(k)})$.

Proof by Construction

$M = (\{q_0, \dots, q_k, r_1, \dots, r_{2k}\}, \{a, b, c, d\}, \{a, b, c, a_1, \dots, a_{2k}\}, q_0, R)$ and R containing the following rules for $1 \leq i \leq k$:

$$
\begin{aligned}
q_0(a(x)) &\longrightarrow a(q_1(x) \dots q_k(x)) \\
q_i(b(xyz)) &\longrightarrow b(r_{2i-1}(x) q_i(y) r_{2i}(z)) \\
q_i(c(xy)) &\longrightarrow c(r_{2i-1}(x) r_{2i}(y)) \\
r_i(d) &\longrightarrow a_i
\end{aligned}
$$

Back
Close