Multi Context-Free Tree Grammars and Multi-component Tree Adjoining Grammars

Joost Engelfriet[1] Andreas Maletti[2]

[1] LIACS, Universiteit Leiden, Leiden, The Netherlands

[2] Institute of Computer Science, UNIVERSITÄT LEIPZIG, Leipzig, Germany
maletti@informatik.uni-leipzig.de

Bordeaux, France

Motivation

Definition

Context-free grammar (N, Σ, S, R) is in Greibach normal form
if each rule $\rho \in R \setminus \{S \to \varepsilon\}$ is of the form $\rho = A \to \sigma A_1 \cdots A_n$
with $\sigma \in \Sigma$ and $A, A_1, \ldots, A_n \in N$

Motivation

Definition

Context-free grammar (N, Σ, S, R) is in Greibach normal form
if each rule $\rho \in R \setminus \{S \to \varepsilon\}$ is of the form $\rho = A \to \sigma A_1 \cdots A_n$
with $\sigma \in \Sigma$ and $A, A_1, \ldots, A_n \in N$

Theorem [Greibach 1965]

Every CFG can be turned into an equivalent CFG in Greibach normal form

Motivation

Definition

CFG (N, Σ, S, R) is **lexicalized** if $\mathrm{occ}_\Sigma(r) \neq \emptyset$
for each rule $(A \to r) \in R \setminus \{S \to \varepsilon\}$

Motivation

Definition

CFG (N, Σ, S, R) is lexicalized if $\mathrm{occ}_{\Sigma}(r) \neq \emptyset$
for each rule $(A \to r) \in R \setminus \{S \to \varepsilon\}$

- CFG in Greibach normal form is lexicalized

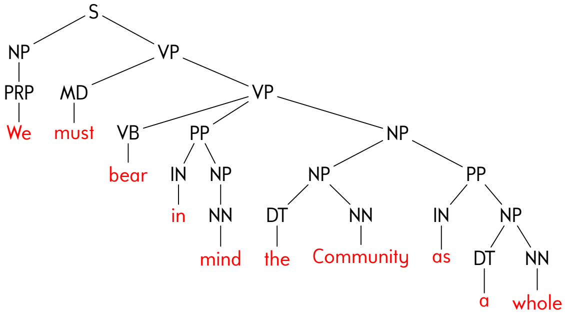
Motivation

Definition

CFG (N, Σ, S, R) is lexicalized if $\mathrm{occ}_\Sigma(r) \neq \emptyset$
for each rule $(A \to r) \in R \setminus \{S \to \varepsilon\}$

- CFG in Greibach normal form is lexicalized
- lexicographers (linguists) love lexicalized grammars
- occurrence of lexical element in a rule is called anchor

Motivation



- linguists nowadays care more about the parse tree than the membership of its yield in the (string) language
- modern grammar formalisms generate tree <u>and</u> string languages

Motivation

Definition

For two tree grammars G and G', of which G' is lexicalized,

- G' weakly lexicalizes G if $\mathrm{yield}(L(G')) = \mathrm{yield}(L(G))$
- G' strongly lexicalizes G if $L(G') = L(G)$

Motivation

Definition

For two tree grammars G and G', of which G' is lexicalized,

- G' weakly lexicalizes G if $\mathrm{yield}(L(G')) = \mathrm{yield}(L(G))$
- G' strongly lexicalizes G if $L(G') = L(G)$

- <u>tree</u> language preserved under <u>strong</u> lexicalization
- <u>string</u> language preserved under <u>weak</u> lexicalization

Motivation

Definition

For two tree grammars G and G', of which G' is lexicalized,

- G' **weakly lexicalizes** G if $\mathrm{yield}(L(G')) = \mathrm{yield}(L(G))$
- G' **strongly lexicalizes** G if $L(G') = L(G)$

- tree language preserved under strong lexicalization
- string language preserved under weak lexicalization
- lifted to classes $\mathcal{C}$ and $\mathcal{C}'$ as usual
  $\mathcal{C}'$-grammars strongly lexicalize $\mathcal{C}$-grammars if for every $G \in \mathcal{C}$ there exists a lexicalized $G' \in \mathcal{C}'$ such that $L(G') = L(G)$

Motivation

Some results:

- CFGs (local tree grammars) weakly lexicalize themselves [Greibach 1965]
- Tree adjoining grammars (TAGs) strongly lexicalize CFGs [Joshi, Schabes 1997]

Motivation

Some results:

- CFGs (local tree grammars) weakly lexicalize themselves [Greibach 1965]
- Tree adjoining grammars (TAGs) strongly lexicalize CFGs [Joshi, Schabes 1997]
- TAGs strongly lexicalize themselves [Joshi, Schabes 1997]

Motivation

Some results:

- CFGs (local tree grammars) weakly lexicalize themselves
  [Greibach 1965]
- Tree adjoining grammars (TAGs) strongly lexicalize CFGs
  [Joshi, Schabes 1997]
- ~~TAGs strongly lexicalize themselves~~
  [Joshi, Schabes 1997]
- TAGs do not strongly lexicalize themselves
  [Kuhlmann, Satta 2012]

Motivation

Some results:

- CFGs (local tree grammars) weakly lexicalize themselves
  [Greibach 1965]
- Tree adjoining grammars (TAGs) strongly lexicalize CFGs
  [Joshi, Schabes 1997]
- ~~TAGs strongly lexicalize themselves~~
  [Joshi, Schabes 1997]
- TAGs do not strongly lexicalize themselves
  [Kuhlmann, Satta 2012]
- Context-free tree grammars (CFTGs)
  strongly lexicalize TAGs and themselves
  [Maletti, Engelfriet 2013]

Contents

1. Motivation

2. Main notion

3. Lexicalization

4. Expressive Power

Main notion

Definition [Engelfriet, Maneth 1998; Kanazawa 2010]

Multiple context-free tree grammar (MCFTG) $G = (N, B, \Sigma, S, R)$

- finite totally ordered ranked alphabet N (nonterminals)
- partition $B \subseteq \mathcal{P}(N)$ of N (big nonterminals)
- finite ranked alphabet Σ (terminals)
- $S \in N^{(0)}$ with $\{S\} \in B$ (initial big nonterminal)
- finite set R of rules of the form $\overline{A} \to \overline{r}$ with $\overline{A} \in B$ and N-linear forest $\overline{r} \in C_{N \cup \Sigma}(X)^+$ such that $\mathrm{rk}^+(\overline{r}) = \mathrm{rk}^+(\overline{A})$ and B saturates $\mathrm{occ}_N(\overline{r})$

Main notion

Definition [Engelfriet, Maneth 1998; Kanazawa 2010]

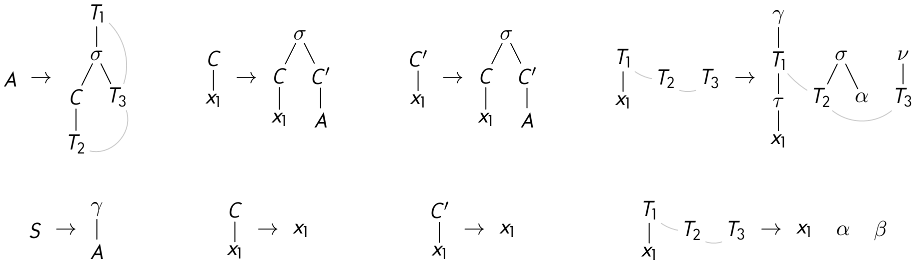
Multiple context-free tree grammar (MCFTG) $G = (N, B, \Sigma, S, R)$

- finite totally ordered ranked alphabet N (nonterminals)
- partition $B \subseteq \mathcal{P}(N)$ of N (big nonterminals)
- finite ranked alphabet Σ (terminals)
- $S \in N^{(0)}$ with $\{S\} \in B$ (initial big nonterminal)
- finite set R of rules of the form $\overline{A} \to \overline{r}$ with $\overline{A} \in B$ and N-linear forest $\overline{r} \in C_{N \cup \Sigma}(X)^{+}$ such that $\mathrm{rk}^{+}(\overline{r}) = \mathrm{rk}^{+}(\overline{A})$ and B saturates $\mathrm{occ}_N(\overline{r})$

- MCFTGs generalize (linear, nondeleting) CFTGs to multiple components
- multiple components synchronously applied to "synchronized" nonterminal occurrences

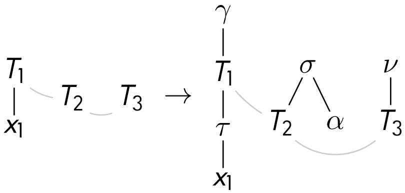
Main notion

Nonterminals S, A, C, C', T_1, T_2, T_3:



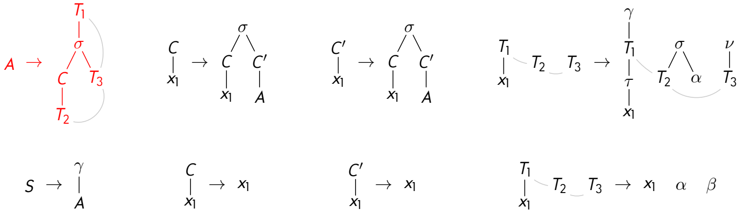
(nonterminals that constitute a big nonterminal connected by splines)

Main notion



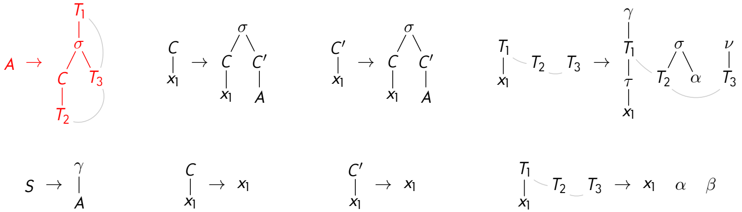
- nonterminals T_1, T_2, T_3 with $T_1 < T_2 < T_3$, terminals $\{\gamma, \tau, \sigma, \alpha, \nu\}$
- big nonterminal in lhs and rhs: $\{T_1, T_2, T_3\}$ of ranks $1, 0, 0$
- 3 corresponding rhs contexts with $1, 0, 0$ variables

Main notion

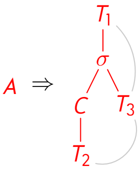


Derivation:

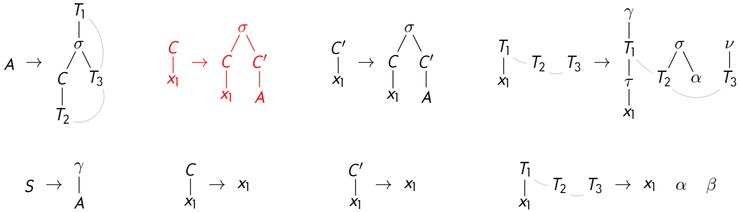
A


$A \to$
$C \to$
$C' \to$
$S \to$
$\to x_1$
$\to x_1$
$\to x_1 \quad \alpha \quad \beta$

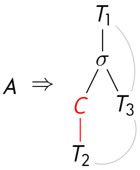
Derivation:


$A \Rightarrow$

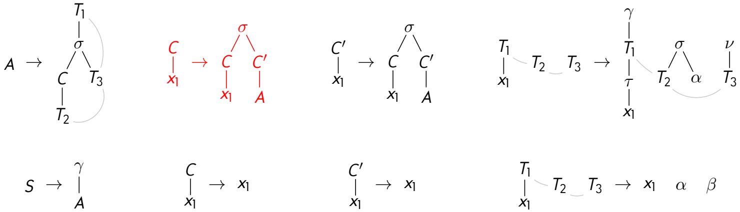
Main notion



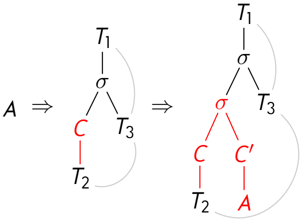
Derivation:



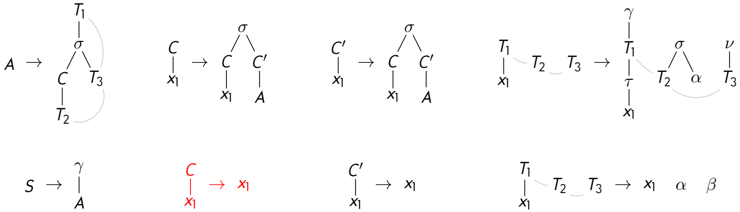
Main notion



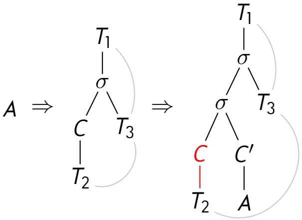
Derivation:



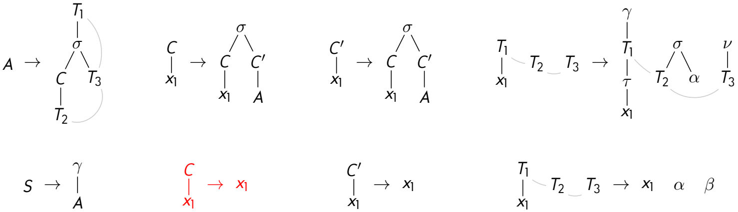
Main notion



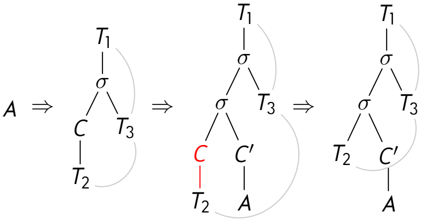
Derivation:



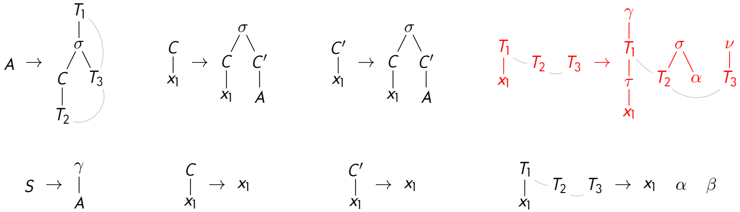
Main notion



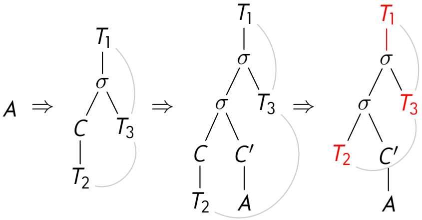
Derivation:



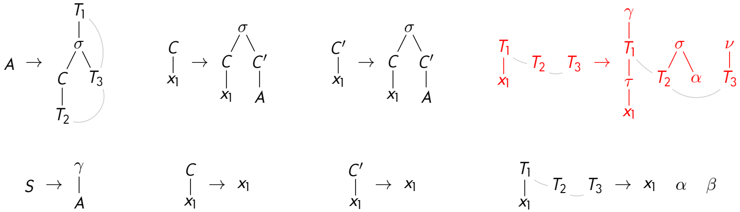
Main notion



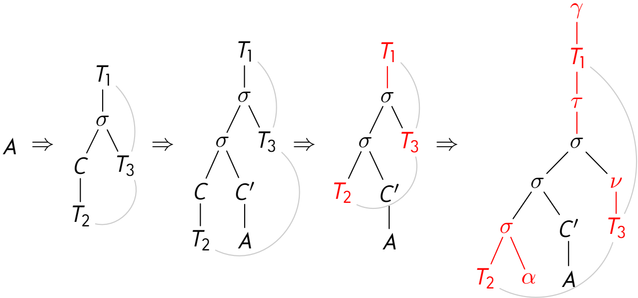
Derivation:



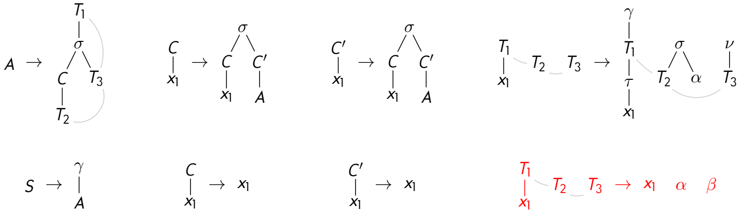
Main notion



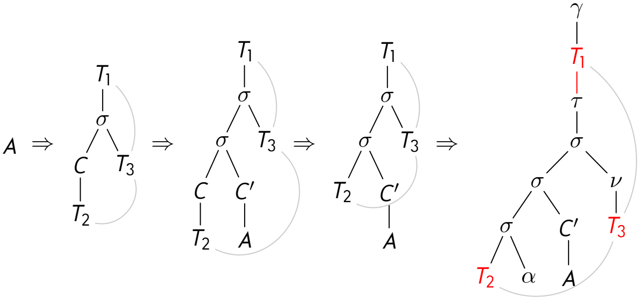
Derivation:



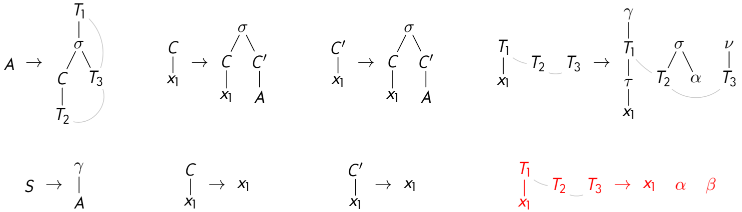
Main notion



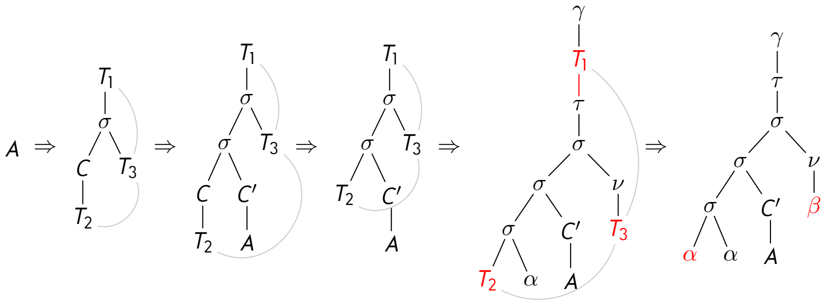
Derivation:



Main notion



Derivation:



Main notion

Definition

The tree language generated by the MCFTG $G = (N, B, \Sigma, S, R)$ is

$$L(G) = \{t \in T_\Sigma \mid S \Rightarrow^* t\}$$

Contents

1 Motivation

2 Main notion

3 Lexicalization

4 Expressive Power

Lexicalization

Definition

Tree language $L \subseteq T_\Sigma$ has finite ambiguity if for every $w \in (\Sigma^{(0)})^*$

$$\{t \in L \mid \text{yield}(t) = w\} \quad \text{is finite}$$

Lexicalization

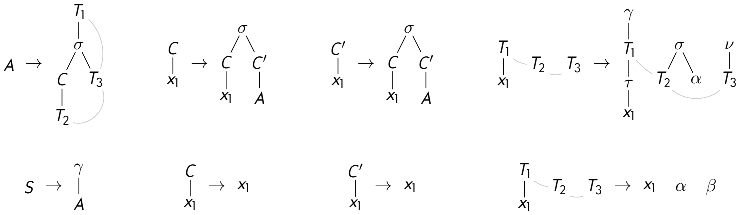
Definition

Tree language $L \subseteq T_\Sigma$ has finite ambiguity if for every $w \in (\Sigma^{(0)})^*$

$$\{t \in L \mid \mathrm{yield}(t) = w\} \qquad \text{is finite}$$

- every string w has finitely many "parses" in L
  (i.e., finitely many tree representations that have w as yield)
- property of the language, not the grammar
  (not to be confused with the similarly named notions for grammars)

MCFTG G:



$L(G)$ has finite ambiguity

Lexicalization

Definition

MCFTG (N, B, Σ, S, R) is lexicalized if $\mathrm{occ}_{\Sigma^{(0)}}(\bar{r}) \neq \emptyset$ for every $\bar{A} \to \bar{r} \in R$

Lexicalization

Definition

MCFTG (N, B, Σ, S, R) is lexicalized if $\mathrm{occ}_{\Sigma^{(0)}}(\bar{r}) \neq \emptyset$ for every $\bar{A} \to \bar{r} \in R$

- each rule contains an anchor (from $\Sigma^{(0)}$)
- lexicalized MCFTGs generate tree languages with finite ambiguity

Lexicalization

Theorem [MCFTGs strongly lexicalize themselves]

For every MCFTG G it is decidable whether $L(G)$ has finite ambiguity

Lexicalization

Theorem [MCFTGs strongly lexicalize themselves]

For every MCFTG G it is decidable whether $L(G)$ has finite ambiguity and if so an equivalent lexicalized MCFTG can be constructed.

Lexicalization

Theorem [MCFTGs strongly lexicalize themselves]

For every MCFTG G it is decidable whether $L(G)$ has finite ambiguity and if so an equivalent lexicalized MCFTG can be constructed.

- multiplicity remains the same
  (multiplicity = maximal cardinality of big nonterminals)
- width increases at most by 1
  (width = maximal rank of nonterminals)
- derivation trees are even related by means of linear deterministic top-down tree transducers with regular look-ahead

Lexicalization

Lexicalization approach:

- normalize terminal rules to contain at least 2 anchors

Lexicalization

Lexicalization approach:

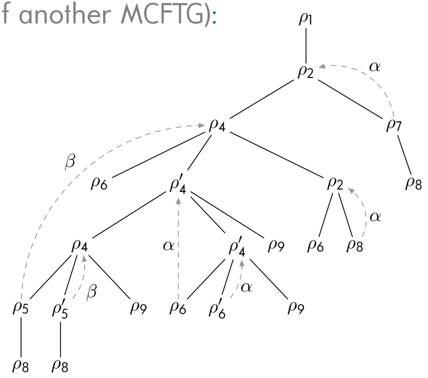
- normalize terminal rules to contain at least 2 anchors
- normalize unary rules to contain at least 1 anchor

Lexicalization

Lexicalization approach:

- normalize terminal rules to contain at least 2 anchors
- normalize unary rules to contain at least 1 anchor
- guess-and-verify strategy for remaining rules

  Derivation tree (of another MCFTG):



Lexicalization

- Extraction (verification) of lexical symbol

Original rule:

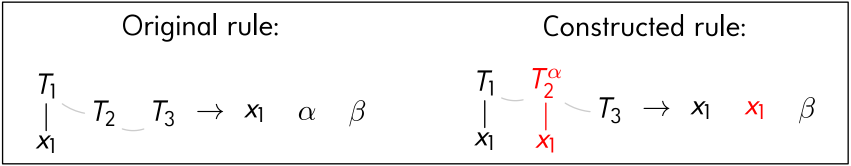
$$\begin{array}{c} T_1 \\ | \\ x_1 \end{array} \; T_2 \; T_3 \;\rightarrow\; x_1 \quad \alpha \quad \beta$$

Constructed rule:

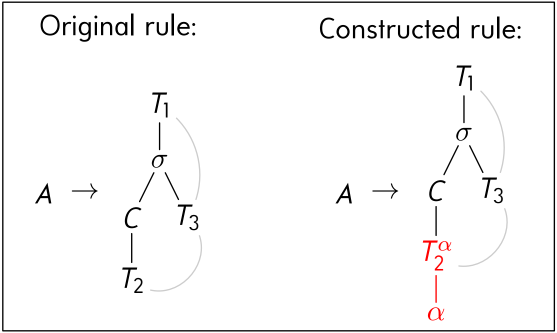
$$\begin{array}{c} T_1 \\ | \\ x_1 \end{array} \; \begin{array}{c} \color{red}{T_2^{\alpha}} \\ \color{red}{|} \\ \color{red}{x_1} \end{array} \; T_3 \;\rightarrow\; x_1 \quad \color{red}{x_1} \quad \beta$$

Lexicalization

- Extraction (verification) of lexical symbol



- Guess of lexical symbol (lexicalizing the rule)



Contents

1. Motivation
2. Main notion
3. Lexicalization
4. Expressive Power

Definition

- Context $c \in C_{N \cup \Sigma}(X_k)$ with k variables is footed
  if $k = 0$ or there is a subtree of the form $\sigma(x_1, \ldots, x_k)$

Expressive Power

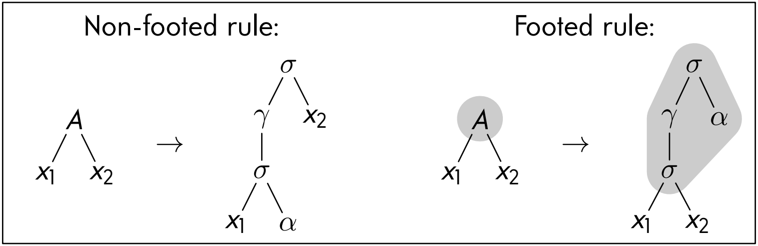
Definition

- Context $c \in C_{N \cup \Sigma}(X_k)$ with k variables is footed
  if $k = 0$ or there is a subtree of the form $\sigma(x_1, \ldots, x_k)$
- Rule $\overline{A} \to \overline{r}$ is footed if all contexts in $\overline{r}$ are footed

Expressive Power

Definition

- Context $c \in C_{N \cup \Sigma}(X_k)$ with k variables is footed
  if $k = 0$ or there is a subtree of the form $\sigma(x_1, \ldots, x_k)$
- Rule $\overline{A} \to \overline{r}$ is footed if all contexts in $\overline{r}$ are footed
- MCFTG (N, B, Σ, S, R) is a multi-component tree adjoining grammar (MC-TAG) if all the rules of R are footed.



Expressive Power

Theorem

For every MCFTG G there exists an equivalent MC-TAG G'

- footed normal form for MCFTGs
- footed CFTGs as expressive as TAGs [Kepser, Rogers 2011]

Expressive Power

Theorem

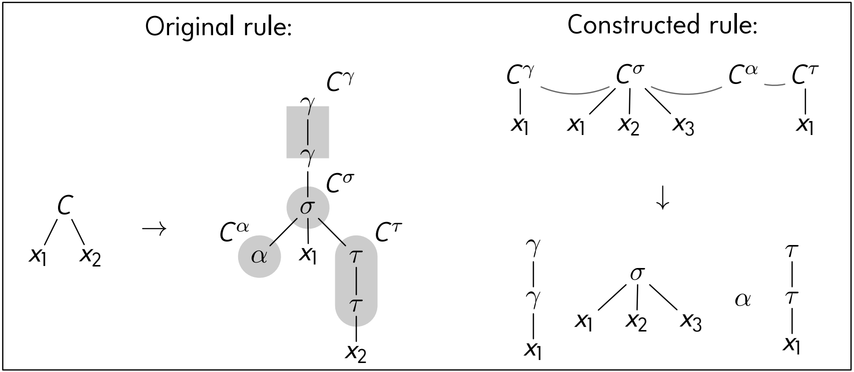
For every MCFTG G there exists an equivalent MC-TAG G'

- footed normal form for MCFTGs
- footed CFTGs as expressive as TAGs [Kepser, Rogers 2011]
- result also true for strict MC-TAG
  (our notion of MC-TAG is essentially "non-strict MC-TAG")
- if MCFTG G lexicalized, then so is MC-TAG G'

Expressive Power

Proof idea:

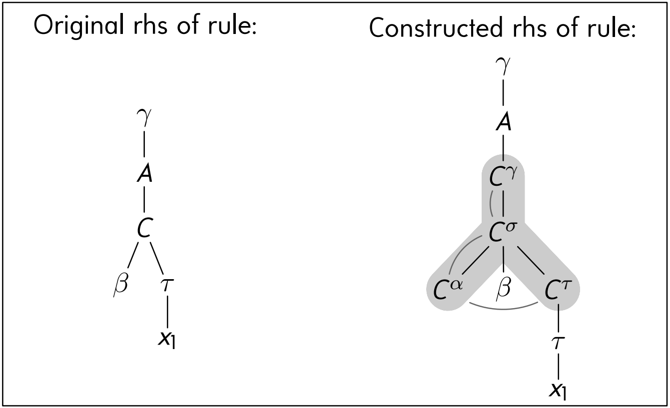
- Decompose context into footed contexts:



Expressive Power

Proof idea:

- Adjust "calls" appropriately:



Expressive Power

Corollary [MC-TAGs strongly lexicalize themselves]

For every MC-TAG G it is decidable whether $L(G)$ has finite ambiguity and if so an equivalent lexicalized MC-TAG can be constructed.

Key points:

- MCFTGs and MC-TAGs equally expressive
- both allow strong lexicalization

Key points:

- MCFTGs and MC-TAGs equally expressive
- both allow strong lexicalization

Thank you for your attention.

Full version available on ▸ arXiv