

Multiple Context-free Grammars

Course 1: Motivations and formal definition

Sylvain Salvati

INRIA Bordeaux Sud-Ouest

ESSLLI 2011

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

- Evidence against context-freeness of natural languages
- Joshi's notion of mildly context sensitivity
- Multiple context free languages

The convergence of mildly context sensitive formalisms

- Hyperedge replacement grammars
- Second order ACGs
- The convergence

Some other virtues of MCFGs for NLP

- Derivation trees, Algebra and Compositional semantics
- Recognizability algorithmic and linguistic descriptions
- Probability distributions

Course outline

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

- Evidence against context-freeness of natural languages

- Joshi's notion of mildly context sensitivity

- Multiple context free languages

The convergence of mildly context sensitive formalisms

- Hyperedge replacement grammars

- Second order ACGs

- The convergence

Some other virtues of MCFGs for NLP

- Derivation trees, Algebra and Compositional semantics

- Recognizability algorithmic and linguistic descriptions

- Probability distributions

Course outline

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages

Joshi's notion of mildly context sensitivity

Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars

Second order ACGs

The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics

Recognizability algorithmic and linguistic descriptions

Probability distributions

Course outline

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Context-freeness of natural languages



Gazdar, Pullum 1982:

Whether non-context-free characteristics can be found in the stringset of some natural language remains an open question, just as it was a quarter century ago.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Verb-object cross-serial dependencies in subordinate clauses

English:

that Charles lets Mary help Peter teach John to swim

German:

daß der Karl die Maria dem Peter den Hans schwimmen lehren helfen läßt

Dutch:

dat Karel Marie Piet Jan laat helpen leren zwemmen

Swiss German:

dass de Karl d'Maria em Peter de Hans laat hëlfe lärne schwüme

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Verb-object cross-serial dependencies in subordinate clauses

English:

that Charles lets Mary help Peter teach John to swim



German:

daß der Karl die Maria dem Peter den Hans schwimmen lehren helfen läßt



Dutch:

dat Karel Marie Piet Jan laat helpen leren zwemmen

Swiss German:

dass de Karl d'Maria em Peter de Hans laat hëlfe lärne schwüme

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Verb-object cross-serial dependencies in subordinate clauses

English:

that Charles **lets** **Mary** **help** **Peter** **teach** **John** to swim

German:

daß der Karl **die Maria** **dem Peter** **den Hans** schwimmen **lehren** **helfen** **läßt**

Dutch:

dat Karel **Marie** **Piet** **Jan** **laat** **helpen** **leren** zwemmen



Swiss German:

dass de Karl **d'Maria** **em Peter** **de Hans** **laat** **hölfe** **lärne** schwüme



- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Evidence against context-freeness of natural languages

Non-context-freeness of natural languages



Shieber 1985:

This Swiss German sentence is grammatical if and only if
 $k = m$ and $l = n$:

De Jan säit, dass mer (d'chind)^k (em Hans)^l es huus haend wele laa^m hälfeⁿ aastriiche.
Jan said that we have wanted (to let the children)^m (help Hans)ⁿ paint the house.

$$\{a^m b^n c^m d^n \mid m, n \geq 1\} \notin CFL$$

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages

Joshi's notion of mildly context sensitivity

Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars

Second order ACGs

The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics

Recognizability algorithmic and linguistic descriptions

Probability distributions

Course outline

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Mildly Context-Sensitive Languages



Joshi 1985:

Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural description.

or: How linguistically adequate are Tree Adjoining Languages?

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Mildly Context-Sensitive Languages



Joshi 1985:

Since the late 1970s there has been vigorous activity in constructing highly constrained grammatical systems by eliminating the transformational component either totally or partially. There is increasing recognition of the fact that the entire range of dependencies that transformational grammars in their various incarnations have tried to account for can be captured satisfactorily by classes of rules that are *nontransformational* and at the same time highly constrained in terms of the classes of grammars and languages they define.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Three formal criteria for an informal notion

- ▶ Limited cross-serial dependencies
- ▶ Constant growth property
- ▶ Polynomial time parsing

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Three formal criteria for an informal notion

- ▶ Limited cross-serial dependencies
- ▶ Constant growth property
- ▶ Polynomial time parsing

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

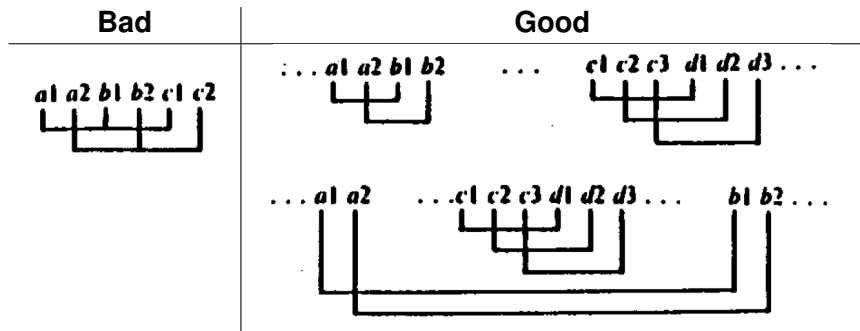
Three formal criteria for an informal notion

- ▶ Limited cross-serial dependencies
- ▶ Constant growth property
- ▶ Polynomial time parsing

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Limited cross-serial dependencies

Dependencies: an informal notion.



- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Further formalization of limited cross-serial dependencies

- Joshi, Vijay Shanker, Weir (1991)



in MCSL; 2) languages in MCSL can be parsed in polynomial time; 3) MCSGs capture only certain kinds of dependencies, e.g., nested dependencies and certain limited kinds of crossing dependencies (e.g., in the subordinate clause constructions in Dutch or some variations of them, but perhaps not in the so-called MIX (or Bach) language, which consists of equal numbers of a's, b's, and c's in any order 4) languages in MCSL have constant growth property, i.e., if the strings of a language

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Further formalization of limited cross-serial dependencies

- Groenink (PhD dissertation): *finite pumpability*



Definition: Finite pumpability. A language L is FINITELY PUMPABLE if there is a constant c_0 such that for any $w \in L$ with $|w| > c_0$, there are a finite number k and strings u_0, \dots, u_k and v_1, \dots, v_k such that $w = u_0 v_1 u_1 v_2 u_2 \cdots u_{k-1} v_k u_k$, and for each i , $1 \leq |v_i| < c_0$, and for any $p \geq 0$, $u_0 v_1^p u_1 v_2^p u_2 \cdots u_{k-1} v_k^p u_k \in L$.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Further formalization of limited cross-serial dependencies



- Kallmeyer (course Düsseldorf): *finite copying*

Cross-Serial Dependencies

The second property (limited amount of cross-serial dependencies) is a little unclear. It can be taken to mean the following:

There is an $n \geq 2$ such that $\{w^k \mid w \in T^*\} \in \mathcal{L}$ for all $k \leq n$.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Constant growth property



► Weir (PhD dissertation):

L is **constant growth** if there is a constant c_0 and a finite set of constants C such that for all $w \in L$ where $|w| > c_0$ there is a $w' \in L$ such that $|w| = |w'| + c$ for some $c \in C$.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity and constant growth property

Joshi:



6.3.2. Constant growth property

This property is connected with the so-called semilinear property, a property also possessed by context-free languages. It can be shown that languages of TAGs also have this property, but I will not give the proof here (Joshi and Yokomori, 1983). Rather, I will give an informal discussion in terms of the constant growth property.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity and constant growth property



Weir:

This mathematical property is intended to be an approximate characterization of the linguistic intuition that sentences of any natural language are built from a finite set of clauses of bounded structure using certain simple linear operations. While the constant growth property is too weak to capture this intuition fully, since it refers only to strings lengths, it represents an interesting attempt to formalize a linguistic intuition. The slightly stronger property of semilinearity may come closer, but is still only an approximation of what is really intended.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity

Definition

A set of vectors \mathcal{V} of \mathbb{N}^k is *linear* iff there are two finite sets \mathcal{C} and \mathcal{P} of elements of \mathbb{N}^k such that every \mathbf{v} in \mathcal{V} is of the form

$$\mathbf{c} + \sum_{\mathbf{w} \in \mathcal{P}} \lambda_{\mathbf{w}} \cdot \mathbf{w}$$

for some \mathbf{c} in \mathcal{C} .

Definition

A set of vectors \mathcal{V} is *semilinear* iff it is a finite union of linear sets

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity

Definition

A set of vectors \mathcal{V} of \mathbb{N}^k is *linear* iff there are two finite sets \mathcal{C} and \mathcal{P} of elements of \mathbb{N}^k such that every \mathbf{v} in \mathcal{V} is of the form

$$\mathbf{c} + \sum_{\mathbf{w} \in \mathcal{P}} \lambda_{\mathbf{w}} \cdot \mathbf{w}$$

for some \mathbf{c} in \mathcal{C} .

Definition

A set of vectors \mathcal{V} is *semilinear* iff it is a finite union of linear sets

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity

Definition

If \mathcal{L} is included in Σ^* , and $|\Sigma| = k$, \mathcal{L} is *semilinear* iff for every monoid homomorphism:

$$\varphi : (\Sigma^*, \cdot) \rightarrow (\mathbb{N}^k, +)$$

$\varphi(\mathcal{L})$ is a semilinear set.

Theorem

A language is semilinear iff it is letter equivalent to a regular language.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Semilinearity

Definition

If \mathcal{L} is included in Σ^* , and $|\Sigma| = k$, \mathcal{L} is *semilinear* iff for every monoid homomorphism:

$$\varphi : (\Sigma^*, \cdot) \rightarrow (\mathbb{N}^k, +)$$

$\varphi(\mathcal{L})$ is a semilinear set.

Theorem

A language is semilinear iff it is letter equivalent to a regular language.

Polynomial parsing

Two kinds of problems:

- ▶ Universal membership/parsing problem:

INPUT	a grammar G and a word w
OUTPUT	yes iff w belongs to the language generated by G ; no otherwise.

- ▶ Membership/parsing problem:

INPUT	a word w
OUTPUT	yes iff w belongs to the language generated by the <i>fixed</i> grammar G ; no otherwise.

Polynomial parsing

Two kinds of problems:

- Universal membership/parsing problem:

INPUT	a grammar G and a word w
OUTPUT	yes iff w belongs to the language generated by G ; no otherwise.

- Membership/parsing problem:

INPUT	a word w
OUTPUT	yes iff w belongs to the language generated by the <i>fixed</i> grammar G ; no otherwise.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Polynomial parsing

Two kinds of problems:

- ▶ Universal membership/parsing problem: this problem is related to a class of presentations of languages.
- ▶ Membership/parsing problem: this problem is related to the language itself.

Languages defined by Monadic Second Order formulae

Universal Membership	Membership
PSPACE-complete	LOGSPACE

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Polynomial parsing

Two kinds of problems:

- ▶ Universal membership/parsing problem: this problem is related to a class of presentations of languages.
- ▶ Membership/parsing problem: this problem is related to the language itself.

Languages defined by Monadic Second Order formulae

Universal Membership	Membership
PSPACE-complete	LOGSPACE

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Joshi's notion of mildly context sensitivity

Polynomial parsing

Two kinds of problems:

- ▶ Universal membership/parsing problem: this problem is related to a class of presentations of languages.
- ▶ Membership/parsing problem: this problem is related to the language itself.

Languages defined by Monadic Second Order formulae

Universal Membership	Membership
PSPACE-complete	LOGSPACE

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Joshi's notion of mildly context sensitivity

Polynomial parsing

Polynomial parsing = polynomial membership problem

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity

Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

Original paper

Fundamental Study

On multiple context-free grammars*

Hiroyuki Seki

*Department of Information and Computer Sciences, Faculty of Engineering Science,
Osaka University, Toyonaka, Osaka 560, Japan*

Takashi Matsumura

Nomura Research Institute, Ltd., Tyu-o-ku, Tokyo 104, Japan

Mamoru Fujii

College of General Education, Osaka University, Toyonaka, Osaka, 560, Japan

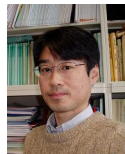
Tadao Kasami

*Department of Information and Computer Sciences, Faculty of Engineering Science,
Osaka University, Toyonaka, Osaka 560, Japan*

Communicated by M. Takahashi

Received July 1989

Revised January 1990



A generalization of context-free grammars

Rule of a context free grammar:

$$A \rightarrow w_1 B_1 \dots w_n B_n w_{n+1}$$

with A, B_1, \dots, B_n non-terminals and w_1, \dots, w_{n+1} string of terminals.

A bottom-up view:

$$A(w_1 x_1 \dots w_n x_n w_{n+1}) \leftarrow B_1(x_1), \dots, B_n(x_n)$$

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

A generalization of context-free grammars

Rule of a context free grammar:

$$A \rightarrow w_1 B_1 \dots w_n B_n w_{n+1}$$

with A, B_1, \dots, B_n non-terminals and w_1, \dots, w_{n+1} string of terminals.

A bottom-up view:

$$A(w_1 x_1 \dots w_n x_n w_{n+1}) \leftarrow B_1(x_1), \dots, B_n(x_n)$$

A generalization of context-free grammars

Replace strings by tuple of strings:

$$B(s_1, \dots, s_m) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

- ▶ the strings s_i are made of terminals and of the variables x_j^i ,
- ▶ the variables x_j^i are pairwise distinct (otherwise we get Groenink's Literal Movement Grammars),
- ▶ each variable x_j^i has at most one occurrence in the string $s_1 \dots s_m$ (otherwise we get Parallel Multiple Context-Free Grammars).

A generalization of context-free grammars

Replace strings by tuple of strings:

$$B(s_1, \dots, s_m) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

- ▶ the strings s_i are made of terminals and of the variables x_j^i ,
- ▶ the variables x_j^i are pairwise distinct (otherwise we get Groenink's Literal Movement Grammars),
- ▶ each variable x_j^i has at most one occurrence in the string $s_1 \dots s_m$ (otherwise we get Parallel Multiple Context-Free Grammars).

A generalization of context-free grammars

Replace strings by tuple of strings:

$$B(s_1, \dots, s_m) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

- ▶ the strings s_i are made of terminals and of the variables x_j^i ,
- ▶ the variables x_j^i are pairwise distinct (otherwise we get Groenink's Literal Movement Grammars),
- ▶ each variable x_j^i has at most one occurrence in the string $s_1 \dots s_m$ (otherwise we get Parallel Multiple Context-Free Grammars).

A generalization of context-free grammars

Replace strings by tuple of strings:

$$B(s_1, \dots, s_m) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

- ▶ the strings s_i are made of terminals and of the variables x_j^i ,
- ▶ the variables x_j^i are pairwise distinct (otherwise we get Groenink's Literal Movement Grammars),
- ▶ each variable x_j^i has at most one occurrence in the string $s_1 \dots s_m$ (otherwise we get Parallel Multiple Context-Free Grammars).

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

A generalisation of context-free grammars

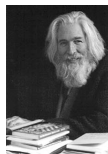
Replace strings by tuple of strings:

$$A(s_1, \dots, s_m) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

This presentation is due to Kanazawa



It is inspired by Smullyan's extended formal systems



Formal definition

A m -MCFG(r) is a 4-tuple (N, T, P, S) such that:

- ▶ N is a ranked alphabet of *non-terminals* of max. rank m .
- ▶ T is an alphabet of *terminals*
- ▶ P is a set of rules of the form:

$$A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

where:

- ▶ A is a non-terminal of rank k , B_i is non-terminal of rank k_i , $n \leq r$,
- ▶ the variables x_j^i are pairwise distinct,
- ▶ the strings s_i are in $(T \cup X)^*$ with $X = \bigcup_{i=1}^n \bigcup_{j=1}^{k_i} \{x_j^i\}$,
- ▶ each variable x_j^i has at most one occurrence in $s_1 \dots s_k$
- ▶ S is a non-terminal of rank 1, *the starting symbol*.

Formal definition

A m -MCFG(r) is a 4-tuple (N, T, P, S) such that:

- ▶ N is a ranked alphabet of *non-terminals* of max. rank m .
- ▶ T is an alphabet of *terminals*
- ▶ P is a set of rules of the form:

$$A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

where:

- ▶ A is a non-terminal of rank k , B_i is non-terminal of rank k_i , $n \leq r$,
- ▶ the variables x_j^i are pairwise distinct,
- ▶ the strings s_i are in $(T \cup X)^*$ with $X = \bigcup_{i=1}^n \bigcup_{j=1}^{k_i} \{x_j^i\}$,
- ▶ each variable x_j^i has at most one occurrence in $s_1 \dots s_k$
- ▶ S is a non-terminal of rank 1, *the starting symbol*.

Formal definition

A m -MCFG(r) is a 4-tuple (N, T, P, S) such that:

- ▶ N is a ranked alphabet of *non-terminals* of max. rank m .
- ▶ T is an alphabet of *terminals*
- ▶ P is a set of rules of the form:

$$A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

where:

- ▶ A is a non-terminal of rank k , B_i is non-terminal of rank k_i , $n \leq r$,
 - ▶ the variables x_j^i are pairwise distinct,
 - ▶ the strings s_i are in $(T \cup X)^*$ with $X = \bigcup_{i=1}^n \bigcup_{j=1}^{k_i} \{x_j^i\}$,
 - ▶ each variable x_j^i has at most one occurrence in $s_1 \dots s_k$
- ▶ S is a non-terminal of rank 1, *the starting symbol*.

Formal definition

A m -MCFG(r) is a 4-tuple (N, T, P, S) such that:

- ▶ N is a ranked alphabet of *non-terminals* of max. rank m .
- ▶ T is an alphabet of *terminals*
- ▶ P is a set of rules of the form:

$$A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$$

where:

- ▶ A is a non-terminal of rank k , B_i is non-terminal of rank k_i , $n \leq r$,
- ▶ the variables x_j^i are pairwise distinct,
- ▶ the strings s_i are in $(T \cup X)^*$ with $X = \bigcup_{i=1}^n \bigcup_{j=1}^{k_i} \{x_j^i\}$,
- ▶ each variable x_j^i has at most one occurrence in $s_1 \dots s_k$
- ▶ S is a non-terminal of rank 1, *the starting symbol*.

The language generated by an MCFG

Given an MCFG $G = (N, T, P, S)$, if the following conditions holds:

- ▶ $B_1(s_1^1, \dots, s_{k_1}^1), \dots, B_n(s_1^n, \dots, s_{k_n}^n)$ are derivable,
- ▶ $A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$ is a rule in P

then $A(t_1, \dots, t_k)$ with $t_i = s_i[x_j^i \leftarrow s_j^i]_{i \in [1;n], j \in [1;k_i]}$ is derivable.

The language defined by G , $L(G)$ is:

$$\{w \mid S(w) \text{ is derivable}\}$$

The language generated by an MCFG

Given an MCFG $G = (N, T, P, S)$, if the following conditions holds:

- ▶ $B_1(s_1^1, \dots, s_{k_1}^1), \dots, B_n(s_1^n, \dots, s_{k_n}^n)$ are derivable,
- ▶ $A(s_1, \dots, s_k) \leftarrow B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_n(x_1^n, \dots, x_{k_n}^n)$ is a rule in P

then $A(t_1, \dots, t_k)$ with $t_i = s_i[x_j^i \leftarrow s_j^i]_{i \in [1;n], j \in [1;k_i]}$ is derivable.

The language define by G , $L(G)$ is:

$$\{w \mid S(w) \text{ is derivable}\}$$

An example

$$S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2)$$

$$P(ax_1, bx_2) \leftarrow P(x_1, x_2)$$

$$P(\epsilon, \epsilon) \leftarrow$$

$$Q(cx_1, dx_2) \leftarrow Q(x_1, x_2)$$

$$Q(\epsilon, \epsilon) \leftarrow$$

$$S(a^n c^m b^n d^m) \leftarrow P(a^n, b^n), Q(c^m, d^m)$$

The language is: $\{a^n c^m b^n d^m \mid n \in \mathbb{N} \wedge m \in \mathbb{N}\}$

An example

$$S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2)$$

$$P(ax_1, bx_2) \leftarrow P(x_1, x_2)$$

$$P(\epsilon, \epsilon) \leftarrow$$

$$Q(cx_1, dx_2) \leftarrow Q(x_1, x_2)$$

$$Q(\epsilon, \epsilon) \leftarrow$$

$$S(a^n c^m b^n d^m) \leftarrow P(a^n, b^n), Q(c^m, d^m)$$

The language is: $\{a^n c^m b^n d^m \mid n \in \mathbb{N} \wedge m \in \mathbb{N}\}$

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

MCFG and mild context sensitivity: folklore

- Joshi, Vijay Shanker, Weir (1991)



precisely defined in Weir (1988). MCTAGs also belong to MCSGs and are in fact equivalent to LCFRs, discussed in Section 6.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

MCFG and mild context sensitivity: folklore



Groenink:

The class of mildly context-sensitive languages is often confused with the class of languages described by the **TAG** formalism; this is because the concept of mild context-sensitivity appears almost uniquely in the **TAG** literature, which often fails to mention stronger formalisms; it is more appropriate to say that **MCS** is most adequately approached by linear **MCFG**.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

Is folklore correct? Cross serial dependencies.

- Joshi, Vijay Shanker, Weir (1991)



in MCSL; 2) languages in MCSL can be parsed in polynomial time; 3) MCSGs capture only certain kinds of dependencies, e.g., nested dependencies and certain limited kinds of crossing dependencies (e.g., in the subordinate clause constructions in Dutch or some variations of them, but perhaps not in the so-called MIX (or Bach) language, which consists of equal numbers of a's, b's, and c's in any order 4) languages in MCSL have constant growth property, i.e., if the strings of a language

MIX is a 2-MCFL(2)

Is folklore correct? Cross serial dependencies.

- Groenink (PhD dissertation): *finite pumpability*



8-8 definition: Finite pumpability. A language L is FINITELY PUMPABLE if there is a constant c_0 such that for any $w \in L$ with $|w| > c_0$, there are a finite number k and strings u_0, \dots, u_k and v_1, \dots, v_k such that $w = u_0 v_1 u_1 v_2 u_2 \cdots u_{k-1} v_k u_k$, and for each i , $1 \leq |v_i| < c_0$, and for any $p \geq 0$, $u_0 v_1^p u_1 v_2^p u_2 \cdots u_{k-1} v_k^p u_k \in L$.

There is a 3-MCFL which is not finitely pumpable.

Is folklore correct? Cross serial dependencies.



- Kallmeyer (course Düsseldorf): *finite copying*

Cross-Serial Dependencies

The second property (limited amount of cross-serial dependencies) is a little unclear. It can be taken to mean the following:

There is an $n \geq 2$ such that $\{w^k \mid w \in T^*\} \in \mathcal{L}$ for all $k \leq n$.

For every $k \in \mathbb{N}$, $\{w^k \mid w \in T^*\}$ is a k -MCFL.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Multiple context free languages

Is folklore correct? Constant growth property.

- ▶ Vijay Shanker, Weir, Joshi (1987)



Theorem

The language defined by an MCFG is semilinear.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
 - └ Multiple context free languages

Is folklore correct? Polynomial parsing.

	Membership	Universal Membership
CFG	LOGCFL-complete	P-complete
TAG	LOGCFL-complete	P-complete
m -MCFG(r)	LOGCFL-complete	P-complete
m -MCFG	LOGCFL-complete	NP-complete (when $m \geq 2$)
MCFG	LOGCFL-complete	PSPACE-complete/ EXPTIME-complete

Satta 1992, Kaji, Nakanishi, Seki and Kasami 1992.

- └ Mildly context sensitive formalism and Multiple Context Free Languages
- └ Multiple context free languages

MCFG and mild context sensitivity

- ▶ either MCFG do not enter within the class mildly context sensitive languages,
- ▶ or we need to make the notion of *limited cross-serial dependencies* more precise.

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages

Joshi's notion of mildly context sensitivity

Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars

Second order ACGs

The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics

Recognizability algorithmic and linguistic descriptions

Probability distributions

Course outline

- └ The convergence of mildly context sensitive formalisms
- └ Hyperedge replacement grammars

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

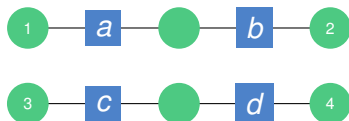
Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

- └ The convergence of mildly context sensitive formalisms
 - └ Hyperedge replacement grammars

Representing tuples of strings as hypergraphs

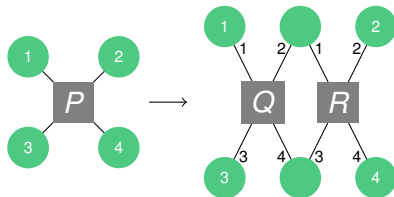
(ab, cd)



- └ The convergence of mildly context sensitive formalisms
- └ Hyperedge replacement grammars

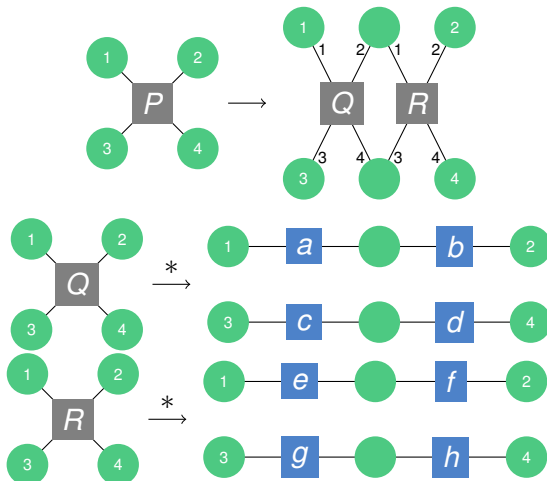
Representing MCFG rules as HR rules

The rule $P(x_1y_1, x_2y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:



Representing MCFG rules as HR rules

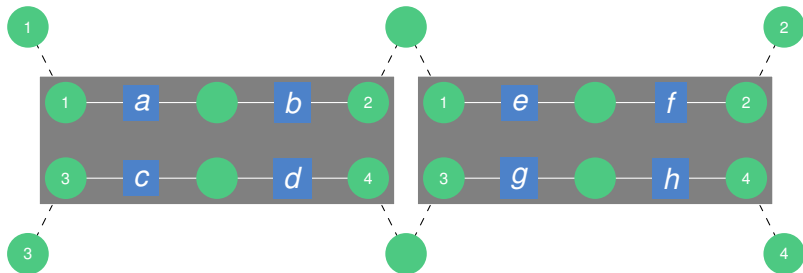
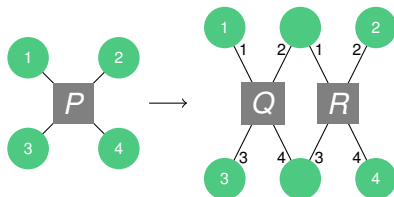
The rule $P(x_1y_1, x_2y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:



- └ The convergence of mildly context sensitive formalisms
 - └ Hyperedge replacement grammars

Representing MCFG rules as HR rules

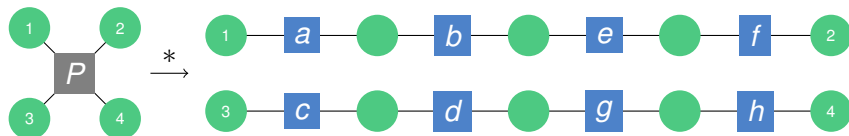
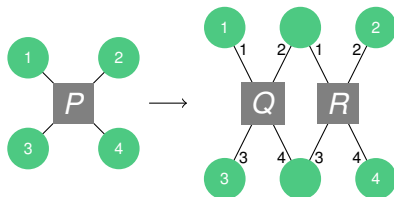
The rule $P(x_1y_1, x_2y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:



- └ The convergence of mildly context sensitive formalisms
- └ Hyperedge replacement grammars

Representing MCFG rules as HR rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:



- └ The convergence of mildly context sensitive formalisms
- └ Hyperedge replacement grammars

String languages of hyperedge replacement grammars

- Engelfriet, Heyker (1991)



$P \longrightarrow \text{Any hypergraph}$

$$\text{MCFG} = \text{STR}(\text{HR})$$

- └ The convergence of mildly context sensitive formalisms
- └ Second order ACGs

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

- └ The convergence of mildly context sensitive formalisms
- └ Second order ACGs

Tuples of strings as λ -terms

(ab, cd)

$\lambda t.t(\lambda z.a(bz))(\lambda z.c(dz))$
 $/ab/$ $/cd/$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$M[X \leftarrow \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))]$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))[X \leftarrow \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))]$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. \textcolor{red}{X}(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))[\textcolor{red}{X} \leftarrow \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))]$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. (\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))(\lambda z. a(bz))(\lambda z. c(dz))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. (\lambda \mathbf{x_1} x_2. Y(\lambda y_1 y_2. t(\lambda z. \mathbf{x_1}(y_1 z))(\lambda z. x_2(y_2 z))))(\lambda z. \mathbf{a(bz))}(\lambda z. c(dz))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. (\lambda x_2. Y(\lambda y_1 y_2. t(\lambda z. (\lambda z. a(bz))(y_1 z))(\lambda z. x_2(y_2 z))))(\lambda z. c(dz))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. (\lambda x_2. Y(\lambda y_1 y_2. t(\lambda z. (\lambda z. a(bz))(y_1 z))(\lambda z. x_2(y_2 z))))(\lambda z. c(dz))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. (\lambda x_2. Y(\lambda y_1 y_2. t(\lambda z. (a(b(\textcolor{red}{y}_1 z))))(\lambda z. x_2(y_2 z))))(\lambda z. c(d z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. (\lambda x_2. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. x_2(y_2 z))))(\lambda z. c(d z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z)))))(\lambda z. (\lambda z. c(d z))(y_2 z)))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z)))))(\lambda z. (\lambda z. c(d z))(y_2 z)))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d y_2 z)))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z)))))[Y \leftarrow \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))]$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z)))))[Y \leftarrow \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))]$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z))))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z)))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. (\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z))))) (\lambda z. e(f z)) (\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. (\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(d(y_2 z)))))(\lambda z. e(f z))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z)))(\lambda z. (c(d y_2 z))))))$$

$$\lambda t. \lambda y_2. t(\lambda z. (a(b((\lambda z. e(f z)) z)))(\lambda z. (c(d(y_2 z)))(\lambda z. g(h z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. \lambda y_2. t(\lambda z. (a(b((\lambda z. e(f z)) z))))(\lambda z. (c(d(y_2 z))))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. \lambda y_2. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d(y_2 z))))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. \lambda y_2. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d(y_2 z))))(\lambda z. g(h z))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d((\lambda z. g(h z)) z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d((\lambda z. g(h z)) z))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d(g(h z))))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(b z))(\lambda z. c(d z))$$

$$Y = \lambda t. t(\lambda z. e(f z))(\lambda z. g(h z))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. t(\lambda z. (a(b(e(f z)))))(\lambda z. (c(d(g(h z))))))$$

Representing MCFG rules as λ -CFG rules

The rule $P(x_1 y_1, x_2 y_2) \leftarrow Q(x_1, x_2), R(y_1, y_2)$:

$$P(M) \leftarrow Q(X), P(Y)$$

with $M = \lambda t. X(\lambda x_1 x_2. Y(\lambda y_1 y_2. t(\lambda z. x_1(y_1 z))(\lambda z. x_2(y_2 z))))$

$$X = \lambda t. t(\lambda z. a(bz))(\lambda z. c(dz))$$

$$Y = \lambda t. t(\lambda z. e(fz))(\lambda z. g(hz))$$

$$\lambda t. Y(\lambda y_1 y_2. t(\lambda z. (a(b(y_1 z))))(\lambda z. (c(dy_2 z))))$$

$$\lambda t. t(\lambda z. (a(b(e(fz)))))(\lambda z. (c(d(g(hz))))))$$

$$X \approx (/ab/, /cd/)$$

$$Y \approx (/ef/, /gh/)$$

$$M[X \leftarrow (/ab/, /cd/), Y \leftarrow (/ef/, /gh/)] \approx (/abef/, /cdgh/)$$

String languages of λ -CFG

- ▶ de Groote, Pogodalla (2004)

$$\text{MCFG} \subseteq \text{STR}(\lambda\text{-CFG})$$

- ▶ S. (2006)

$$\text{STR}(\lambda\text{-CFG}) \subseteq \text{MCFG}$$



String languages of λ -CFG



- ▶ de Groote, Pogodalla (2004)

$$\text{MCFG} \subseteq \text{STR}(\lambda\text{-CFG})$$

- ▶ S. (2006)

$$\text{STR}(\lambda\text{-CFG}) \subseteq \text{MCFG}$$

- └ The convergence of mildly context sensitive formalisms
 - └ The convergence

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs

The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

- └ The convergence of mildly context sensitive formalisms
 - └ The convergence

The convergence

- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

- └ The convergence of mildly context sensitive formalisms
 - └ The convergence

The convergence

- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

- └ The convergence of mildly context sensitive formalisms
 - └ The convergence

The convergence

- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

- └ The convergence of mildly context sensitive formalisms
 - └ The convergence

The convergence

- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

The convergence

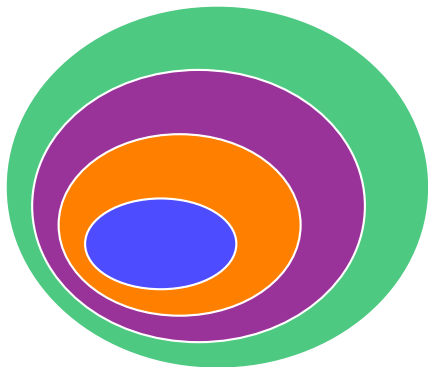
- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

The convergence

- ▶ Linear Context Free Rewriting Systems, OUT(Deterministic Tree Walking Transducer), STR(Hyperedge Replacement Grammars) (Engelfriet, Heyker 1991, Weir 1992)
- ▶ Minimalist Grammars (Michaelis 2001, Harkema 2001)
- ▶ Multi-Component Tree Adjoining Grammars (Vijay Shanker, Weir, Joshi 1987)
- ▶ Local Unordered Scattered Context Grammar (Rambow, Satta 1994)
- ▶ yield(OUT(finite copying tree transducers)) (Engelfriet, Rozenberg, Slutzki 1980)
- ▶ second order string Abstract Categorical Grammars (de Groote, Pogodalla 2004, S. 06)

- └ The convergence of mildly context sensitive formalisms
- └ The convergence

The convergences



$$\begin{aligned} MCTAG &= MCFG = HR \\ &= OUT(DTWT) \\ &= yDT_{fc}(REGT) = LUSCG = MG \\ &= \lambda\text{-CFL}_{lin}(4) = \lambda\text{-CFL}_{lin} \end{aligned}$$

$$\begin{aligned} MCFG_{wn} &= IO_{nd} = OI_{nd} \\ &= CCFG = \lambda\text{-CFL}_{lin}(3) \\ TAG &= LIG = CCG = HG \\ 2\text{-}MCFG_{wn} \end{aligned}$$

$$CFL = 1\text{-}MCFL = \lambda\text{-CFL}_{lin}(2)$$

- └ The convergence of mildly context sensitive formalisms
- └ The convergence

On convergence of formalisms

- Joshi, Vijay Shanker, Weir (1991)



When two formalisms based on apparently completely different ideas turn out to be equivalent, there is a possibility that we are getting a handle on some fundamental properties of the objects that these formalisms were designed to describe. When more than two distinct formalisms turn out to be equivalent, the possibility is even greater. In fact, a deeper understanding of the relationships between these formalisms is obtained if we look at the derivation structures (related to structural descriptions) provided by each formalism. A first attempt to capture the closeness of some of

► Chomsky (2004)

- ▶ Chomsky (2004)

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages

Joshi's notion of mildly context sensitivity

Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars

Second order ACGs

The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics

Recognizability algorithmic and linguistic descriptions

Probability distributions

Course outline

- └ Some other virtues of MCFGs for NLP
 - └ Derivation trees, Algebra and Compositional semantics

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

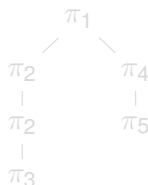
- ▶ de Saussure



L'expression simple sera algébrique ou ne sera pas.
... a linguistic description "sera algébrique ou ne sera pas"
 [Newmeyer 1986]

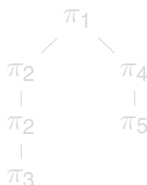
Rules and free algebra

$\pi_1 :$	$Q \rightarrow P \rightarrow S$	$S(x_1 y_1 x_2 y_2) \leftarrow$	$P(x_1, x_2), Q(y_1, y_2)$
$\pi_2 :$	$P \rightarrow P$	$P(ax_1, bx_2) \leftarrow$	$P(x_1, x_2)$
$\pi_3 :$	P	$P(\epsilon, \epsilon) \leftarrow$	
$\pi_4 :$	$Q \rightarrow Q$	$Q(cx_1, dx_2) \leftarrow$	$Q(x_1, x_2)$
$\pi_5 :$	Q	$Q(\epsilon, \epsilon) \leftarrow$	



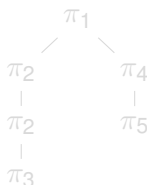
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



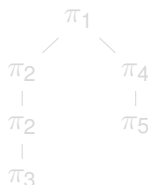
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



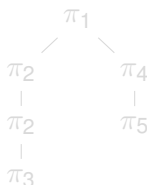
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



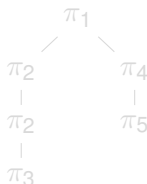
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & \left| \begin{array}{lcl} S(x_1 y_1 x_2 y_2) & \leftarrow & P(x_1, x_2), Q(y_1, y_2) \\ \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\ \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\ \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\ \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow \end{array} \right.
 \end{array}$$



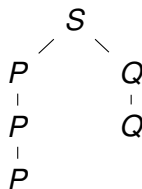
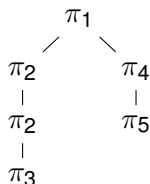
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & \left| \begin{array}{lcl} S(x_1 y_1 x_2 y_2) & \leftarrow & P(x_1, x_2), Q(y_1, y_2) \\ \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\ \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\ \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\ \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow \end{array} \right.
 \end{array}$$



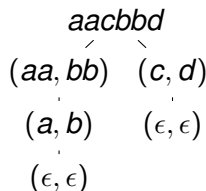
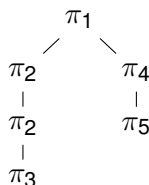
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



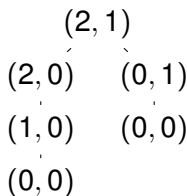
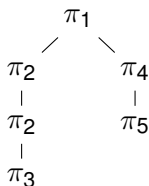
Rules and free algebra

$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



Rules and free algebra

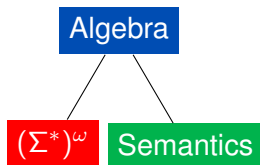
$$\begin{array}{lcl}
 \pi_1 : & Q \rightarrow P \rightarrow S & S(x_1 y_1 x_2 y_2) \leftarrow P(x_1, x_2), Q(y_1, y_2) \\
 \pi_2 : & P \rightarrow P & P(ax_1, bx_2) \leftarrow P(x_1, x_2) \\
 \pi_3 : & P & P(\epsilon, \epsilon) \leftarrow \\
 \pi_4 : & Q \rightarrow Q & Q(cx_1, dx_2) \leftarrow Q(x_1, x_2) \\
 \pi_5 : & Q & Q(\epsilon, \epsilon) \leftarrow
 \end{array}$$



MCFG and as algebras

MCFG =

- ▶ multi-sorted algebra
- ▶ each sort is interpreted in $(\Sigma^*)^k$ for k in \mathbb{N} ,
- ▶ each rule $\pi : B_1 \rightarrow \dots B_n \rightarrow A$ is an appropriate operation over $(\Sigma^*)^{k_1} \rightarrow \dots \rightarrow (\Sigma^*)^{k_n} \rightarrow (\Sigma^*)^k$
- ▶ other interpretations of the underlying free algebra can provide compositional semantics for the languages.



- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

Finite state automata for trees

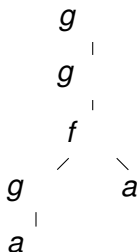
- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



$$\begin{aligned}
 A &= a \\
 G_1 X &= gX \\
 G_2 X &= gX \\
 G_3 X &= gX \\
 FXY &= fxy
 \end{aligned}$$

Finite state automata for trees

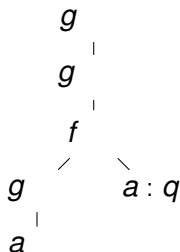
- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



$$\begin{aligned}
 A &= a \\
 G_1 x &= gx \\
 G_2 x &= gx \\
 G_3 x &= gx \\
 Fxy &= fxy
 \end{aligned}$$

Finite state automata for trees

- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$

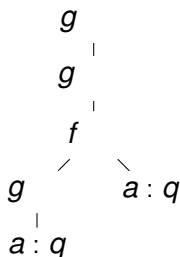


A

$$\begin{aligned}
 A &= a \\
 G_1 X &= g X \\
 G_2 X &= g X \\
 G_3 X &= g X \\
 F X Y &= f X Y
 \end{aligned}$$

Finite state automata for trees

- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



A

A

$$A = a$$

$$G_1 X = gX$$

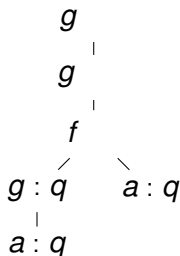
$$G_2 X = gX$$

$$G_3 X = gX$$

$$FXY = fxy$$

Finite state automata for trees

- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



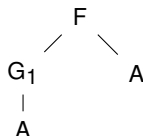
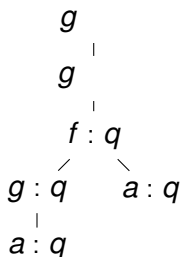
G_1
|
 A

A

$A = a$
 $G_1 X = gX$
 $G_2 X = gX$
 $G_3 X = gX$
 $FXY = fxy$

Finite state automata for trees

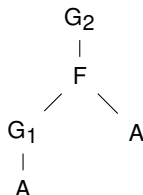
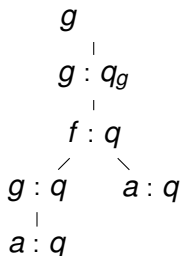
- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



$$\begin{aligned}
 A &= a \\
 G_1 x &= gx \\
 G_2 x &= gx \\
 G_3 x &= gx \\
 Fxy &= fxy
 \end{aligned}$$

Finite state automata for trees

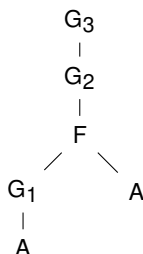
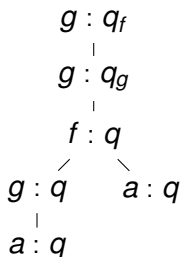
- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$



$$\begin{aligned}
 A &= a \\
 G_1 x &= g x \\
 G_2 x &= g x \\
 G_3 x &= g x \\
 F x y &= f x y
 \end{aligned}$$

Finite state automata for trees

- ▶ $f : o \rightarrow o \rightarrow o, g : o \rightarrow o, a : o,$
- ▶ STATES = $\{q; q_g; q_f\}$, FINAL STATES = $\{q_f\}$,
- ▶ $a \rightarrow q, g(q) \rightarrow q, g(q) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q, q) \rightarrow q,$
- ▶ $A : q, G_1 : q \rightarrow q, G_2 : q \rightarrow q_g, G_3 : q_g \rightarrow q_f,$
 $F : q \rightarrow q \rightarrow q.$

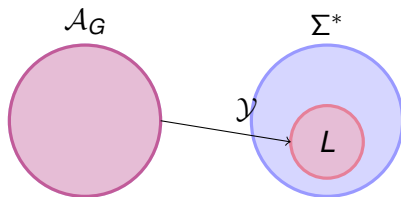


$$\begin{aligned}
 A &= a \\
 G_1 X &= gX \\
 G_2 X &= gX \\
 G_3 X &= gX \\
 FXY &= fxy
 \end{aligned}$$

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Recognizability and MCFG

Given an MCFG G and a recognizable tree language T over its rule trees:

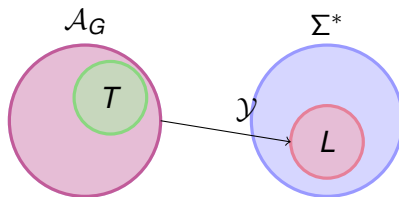


MCFGs are closed under recognizable constraints on the rule tree.

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Recognizability and MCFG

Given an MCFG G and a recognizable tree language T over its rule trees:

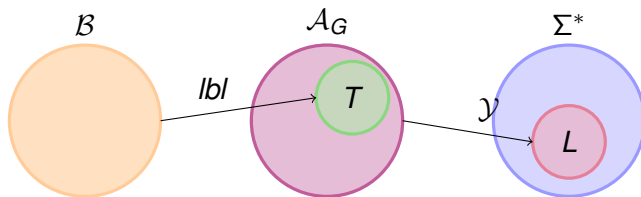


MCFGs are closed under recognizable constraints on the rule tree.

- Some other virtues of MCFGs for NLP
 - Recognizability algorithmic and linguistic descriptions

Recognizability and MCFG

Given an MCFG G and a recognizable tree language T over its rule trees:

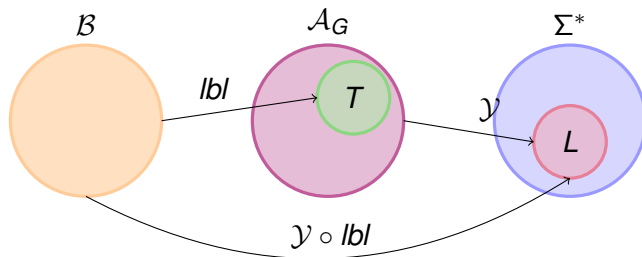


MCFGs are closed under recognizable constraints on the rule tree.

- Some other virtues of MCFGs for NLP
 - Recognizability algorithmic and linguistic descriptions

Recognizability and MCFG

Given an MCFG G and a recognizable tree language T over its rule trees:



MCFGs are closed under recognizable constraints on the rule tree.

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Logical characterization of recognizability

- ▶ Recognizable sets of trees are closed under boolean operations
- ▶ For describing finite ranked trees of arity at most k , the *weak Monadic Second-Order Logic* captures exactly recognizability (Doner, Thatcher and Wright).
- ▶ A substantial part of GB can be formalized using MSOL (Rogers 1995),
- ▶ Constraints on the syntactic structures of Blache's property grammars are MSOL-definable (Duchiez, Prost 2009).

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Logical characterization of recognizability

- ▶ Recognizable sets of trees are closed under boolean operations
- ▶ For describing finite ranked trees of arity at most k , the *weak Monadic Second-Order Logic* captures exactly recognizability (Doner, Thatcher and Wright).
- ▶ A substantial part of GB can be formalized using MSOL (Rogers 1995),
- ▶ Constraints on the syntactic structures of Blache's property grammars are MSOL-definable (Duchiez, Prost 2009).

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Logical characterization of recognizability

- ▶ Recognizable sets of trees are closed under boolean operations
- ▶ For describing finite ranked trees of arity at most k , the *weak Monadic Second-Order Logic* captures exactly recognizability (Doner, Thatcher and Wright).
- ▶ A substantial part of GB can be formalized using MSOL (Rogers 1995),
- ▶ Constraints on the syntactic structures of Blache's property grammars are MSOL-definable (Duchiez, Prost 2009).

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

Logical characterization of recognizability

- ▶ Recognizable sets of trees are closed under boolean operations
- ▶ For describing finite ranked trees of arity at most k , the *weak Monadic Second-Order Logic* captures exactly recognizability (Doner, Thatcher and Wright).
- ▶ A substantial part of GB can be formalized using MSOL (Rogers 1995),
- ▶ Constraints on the syntactic structures of Blache's property grammars are MSOL-definable (Duchiez, Prost 2009).

- └ Some other virtues of MCFGs for NLP
 - └ Recognizability algorithmic and linguistic descriptions

A Two-Step Approach to Non-Contextfreeness

- ▶ Michaelis, Mönnich, Morawietz (2000)



The main result of this paper is a description of non context-free phenomena equivalently in terms of (a) regular tree languages (to express the recursive properties) and regular string languages (to establish the intended linguistic relations) and (b) transductions definable in monadic second-order (MSO) logic which are defined on a domain of finite trees that is characterized as the model set of a closed MSO formula. In fact, we shall not deal directly with the two types

- └ Some other virtues of MCFGs for NLP
- └ Probability distributions

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

Evidence against context-freeness of natural languages
Joshi's notion of mildly context sensitivity
Multiple context free languages

The convergence of mildly context sensitive formalisms

Hyperedge replacement grammars
Second order ACGs
The convergence

Some other virtues of MCFGs for NLP

Derivation trees, Algebra and Compositional semantics
Recognizability algorithmic and linguistic descriptions
Probability distributions

Course outline

- └ Some other virtues of MCFGs for NLP
- └ Probability distributions

Probabilistic automata



► Ellis (1971)

It is quite natural to assign probabilities (or frequencies) to the sentences of a language to try to get some quantitative measure of “efficiency” of translators and grammars for the language. The model obtained by doing this is called a probabilistic language. Probabilistic languages, along with related concepts of probabilistic grammars and probabilistic automata are defined, and a specific application of this theory is presented in the development of Probabilistic Tree Automata which characterize Probabilistic Context Free Languages.

- └ Some other virtues of MCFGs for NLP
 - └ Probability distributions

MCFG and probabilities

- Probabilistic techniques that are use for PCFG can be straightforwardly extended to MCFG.

Outline

Mildly context sensitive formalism and Multiple Context Free Languages

- Evidence against context-freeness of natural languages
- Joshi's notion of mildly context sensitivity
- Multiple context free languages

The convergence of mildly context sensitive formalisms

- Hyperedge replacement grammars
- Second order ACGs
- The convergence

Some other virtues of MCFGs for NLP

- Derivation trees, Algebra and Compositional semantics
- Recognizability algorithmic and linguistic descriptions
- Probability distributions

Course outline

Course outline

- ▶ Day 2: expressive power and parsing
- ▶ Day 3: well-nestedness
- ▶ Day 4: pumping properties
- ▶ Day 5: MIX is a 2-MCFL
- ▶ For learning see: Clark, Yoshinaka (2009-10)

Course outline

- ▶ Day 2: expressive power and parsing
- ▶ Day 3: well-nestedness
- ▶ Day 4: pumping properties
- ▶ Day 5: MIX is a 2-MCFL
- ▶ For learning see: Clark, Yoshinaka (2009-10)

Course outline

- ▶ Day 2: expressive power and parsing
- ▶ Day 3: well-nestedness
- ▶ Day 4: pumping properties
- ▶ Day 5: MIX is a 2-MCFL
- ▶ For learning see: Clark, Yoshinaka (2009-10)

Course outline

- ▶ Day 2: expressive power and parsing
- ▶ Day 3: well-nestedness
- ▶ Day 4: pumping properties
- ▶ Day 5: MIX is a 2-MCFL
- ▶ For learning see: Clark, Yoshinaka (2009-10)

Course outline

- ▶ Day 2: expressive power and parsing
- ▶ Day 3: well-nestedness
- ▶ Day 4: pumping properties
- ▶ Day 5: MIX is a 2-MCFL
- ▶ For learning see: Clark, Yoshinaka (2009-10)