On the Parameterized Complexity of Linear Context-Free Rewriting Systems

Martin Berglund & Henrik Björklund & Frank Drewes

Umeå University

MOL 13, Sofia, August 9, 2013

Mildly Context Sensitive Languages

Motivation: Context-free grammars cannot express all features of natural languages.

Examples:

- Copy structures: $\{w \cdot w \mid w \in \Sigma^*\}$
- Permutation structures: $w \cdot \pi(w)$
- 3-fold counting: $a^n\ b^n\ c^n$
- Etc.

Mildly Context-Sensitive Languages

Aravind Joshi suggested that a formalism for modelling natural languages should have the following properties:

- Capture, e.g, the features from the last slide.
- Extend the context-free languages.
- Have linear growth.
- Have polynomial time parsing.

The Chomsky Hierarchy

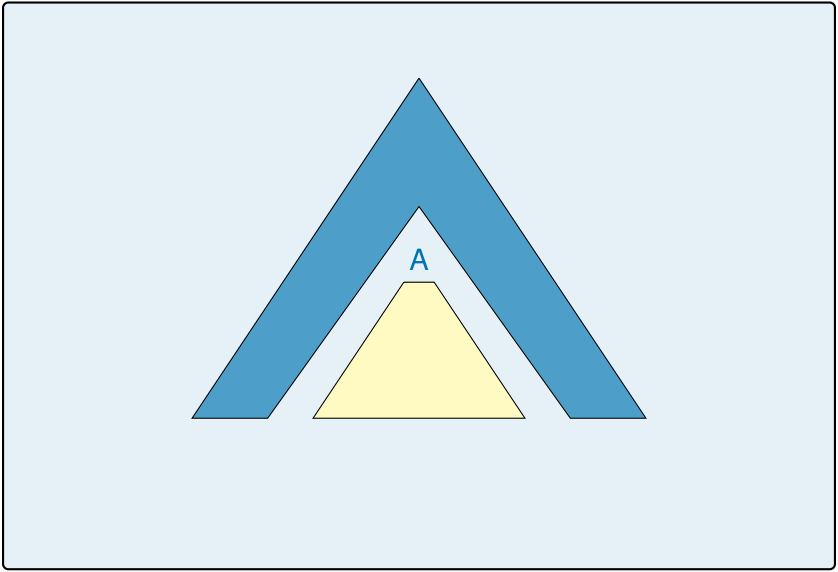
Class	**Automaton**	**Grammar**
Regular	FA	Regular
Context-free	PDA	Context-free
Mildly context-sensitive	???	???
Context-sensitive	Linspace TMs	Context-sensitive
RE	TM acceptors	Type 0

Linear Context-Free Rewriting Systems

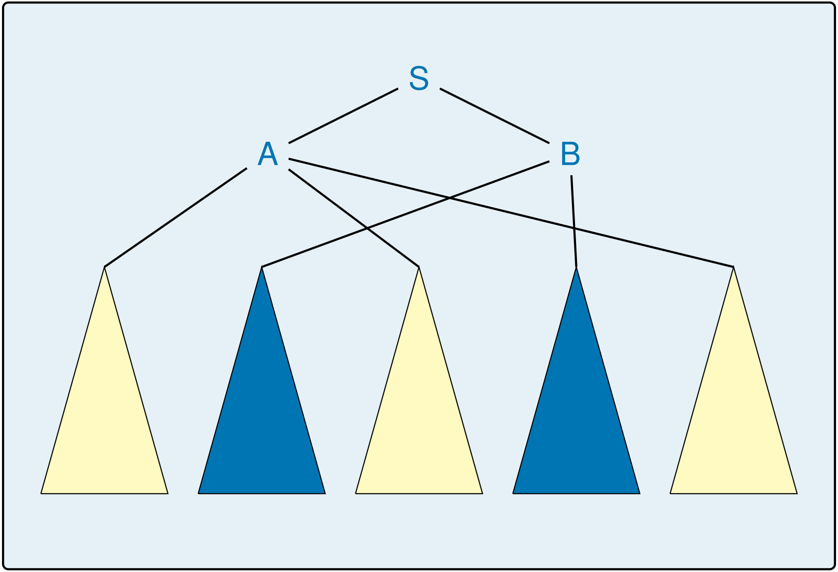
We will focus on Linear Context-Free Rewriting Systems (LCFRS).

- Vijay-Shanker, Weir, Joshi, 1987
- Seki, Matsumura, Fujii, Kasami independently defined the very similar Multiple Context-Free Grammars in 1991

Linear Context-Free Rewriting Systems



Linear Context-Free Rewriting Systems



Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$

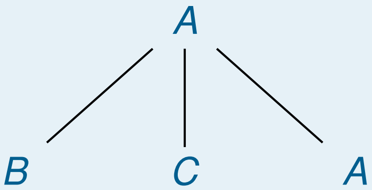
Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$

A

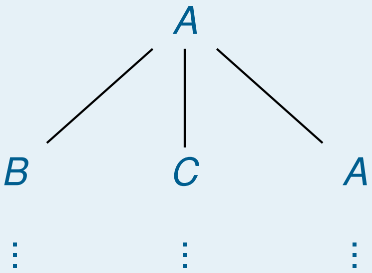
Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$



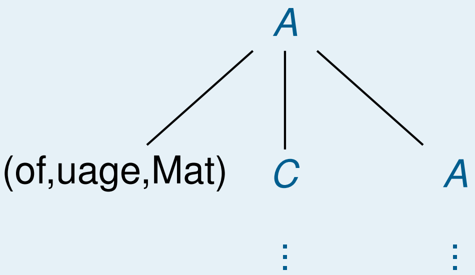
Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$



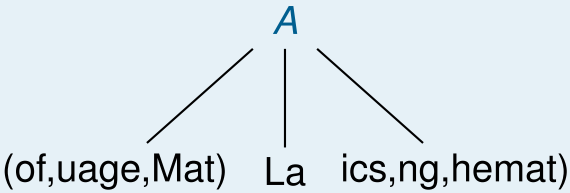
Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$



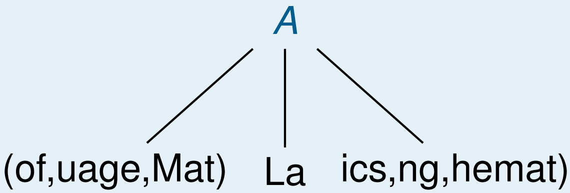
Linear Context-Free Rewriting Systems

$A \rightarrow f_A(B, C, A)$



Linear Context-Free Rewriting Systems

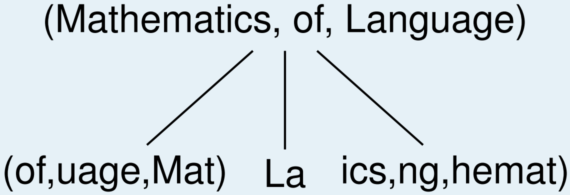
$A \rightarrow f_A(B, C, A)$



$$f_A((x_1, x_2, x_3), y, (z_1, z_2, z_3)) = (x_3 z_3 z_1, x_1, y z_2 x_2)$$

Linear Context-Free Rewriting Systems

$A \to f_A(B, C, A)$



$$f_A((x_1, x_2, x_3), y, (z_1, z_2, z_3)) = (x_3 z_3 z_1, x_1, y z_2 x_2)$$

Linear Context-Free Rewriting System

Consider a grammar with the following rules:

- $S \rightarrow f(A)$
- $A \rightarrow g(A)$
- $A \rightarrow (\varepsilon, \varepsilon, \varepsilon, \varepsilon)$

where

$$g(x, y, z, w) = (ax, by, cz, dw)$$

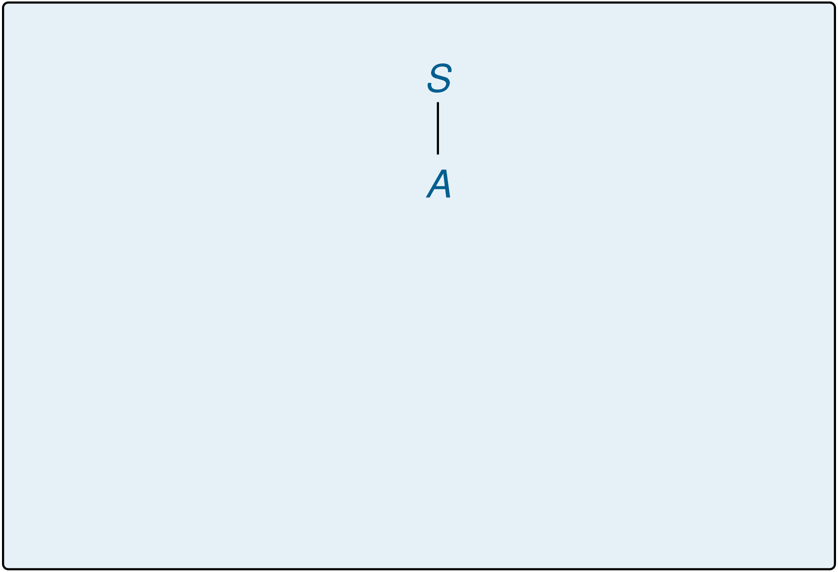
and

$$f(x, y, z, w) = xyzw.$$

Linear Context-Free Rewriting System

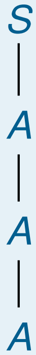
S

Linear Context-Free Rewriting System

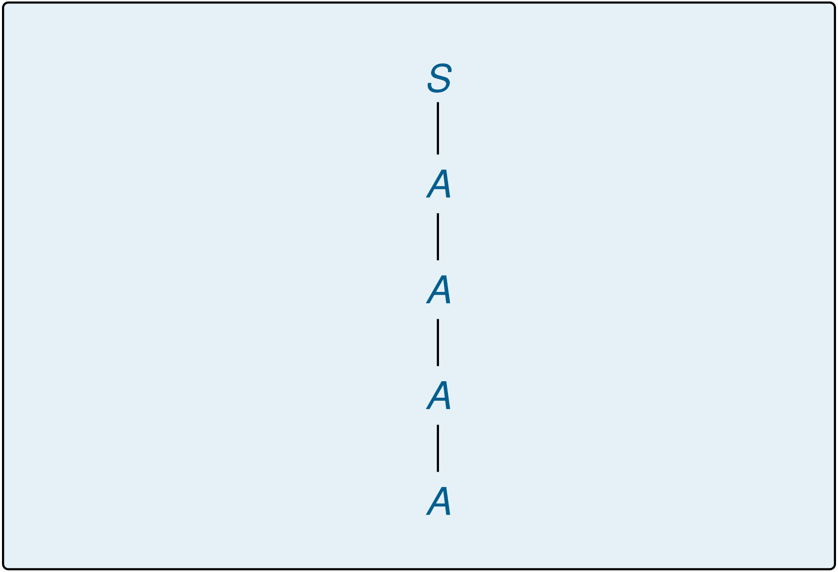


Linear Context-Free Rewriting System

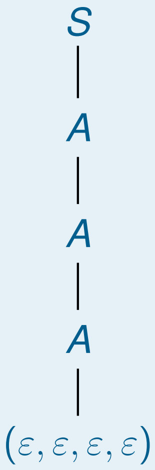
Linear Context-Free Rewriting System



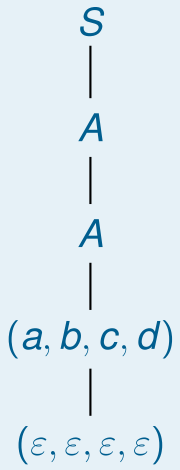
Linear Context-Free Rewriting System



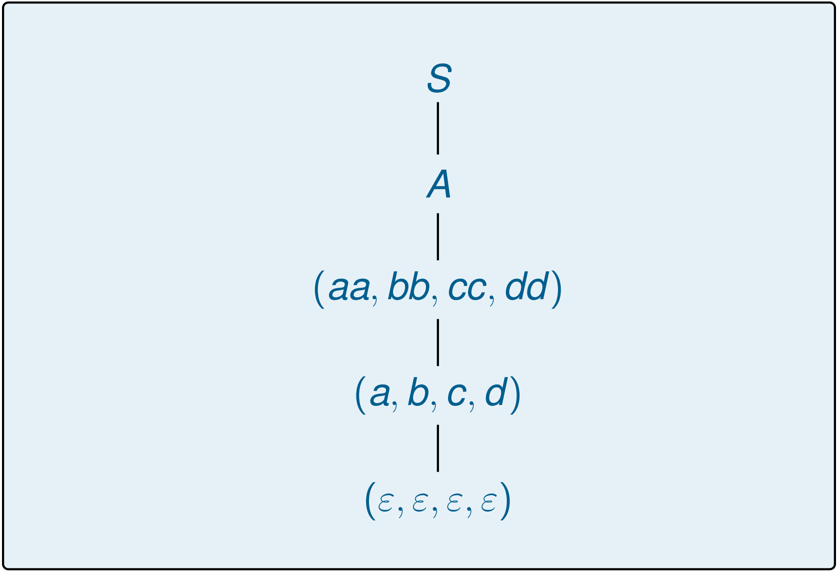
Linear Context-Free Rewriting System



Linear Context-Free Rewriting System



Linear Context-Free Rewriting System



Linear Context-Free Rewriting System

$$S$$

$$(aaa, bbb, ccc, ddd)$$

$$(aa, bb, cc, dd)$$

$$(a, b, c, d)$$

$$(\varepsilon, \varepsilon, \varepsilon, \varepsilon)$$

Linear Context-Free Rewriting System

$$aaabbbcccddd$$

$$(aaa, bbb, ccc, ddd)$$

$$(aa, bb, cc, dd)$$

$$(a, b, c, d)$$

$$(\varepsilon, \varepsilon, \varepsilon, \varepsilon)$$

PTIME Parsing?

The non-uniform membership problem for LCFRS is polynomial.

There are algorithms that run in time $\mathcal{O}(n^d)$, where d is the maximal combined arity of the functions.

The combined arity of the function

$$f_A((x_1, x_2, x_3), y, (z_1, z_2, z_3)) = (x_3 z_3 z_1, x_1, x_2 y)$$

is 10.

The uniform membership problem is PSPACE-complete.

Parameterized Complexity

Developed by Downey and Fellows in the 1990s.

Basic idea: Focus on a particular aspect of the problem and its effect on the complexity.

A parameterized problem instance has two parts:

- the problem description
- a parameter k that depends on the description.

For the LCFRS membership problem, the problem description will be

- a description of the grammar and
- the string to test.

We consider three different parameterizations.

Fixed-Parameter Tractability

A parameterized problem is fixed-parameter tractable if there is an algorithm that can solves every instance (x, k) in time

$$f(k) \cdot |x|^c$$

for some computable function f and constant c.

Is this the same as saying that ”for fixed k, the problem is polynomial”?

What's the Difference?

Consider LTL Model Checking.

This is the problem of deciding whether, e.g, a circuit board N satisfies an LTL formula ϕ.

The problem can be solved in time $2^{|\phi|} \cdot |N|$.

This is a monomial in $|N|$, where the coefficients are determined by ϕ. Parameterized by $|\phi|$, it is FPT.

Imagine, instead, that the best algorithms ran in time $|N|^{|\phi|}$.

This would be a polynomial in $|N|$, where the degree would be determined by ϕ.

The W Hierarchy

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[t] \subseteq W[SAT] \subseteq W[P] \subseteq XP$$

Problems:

- FPT: p-VERTEX COVER
- W[1]: p-CLIQUE
- W[2]: p-DOMINATING SET
- W[SAT]: WEIGHTED SATISFIABILITY
- W[P]: p-MINIMUM AXIOM SET
- XP: p-PEG GAME

Parameterized Reductions

A parameterized reduction is basically just like a polynomial time reduction.

The difference is that

- the running time does not have to be polynomial, but fixed-parameter tractable
- the new parameter can only depend on the old parameter.

This means that if problem A is FPT and we can reduce from B to A, then B is also FPT.

First parameterization: Fan-out

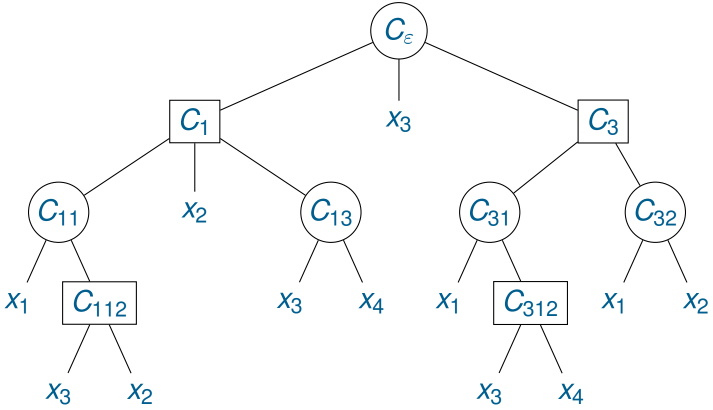
Theorem

The membership problem for LCFRS, parameterized by the fan-out, is W[SAT]-hard, even if the rank is fixed to 1.

The proof is by reduction from WEIGHTED MONOTONE SATISFIABILITY.

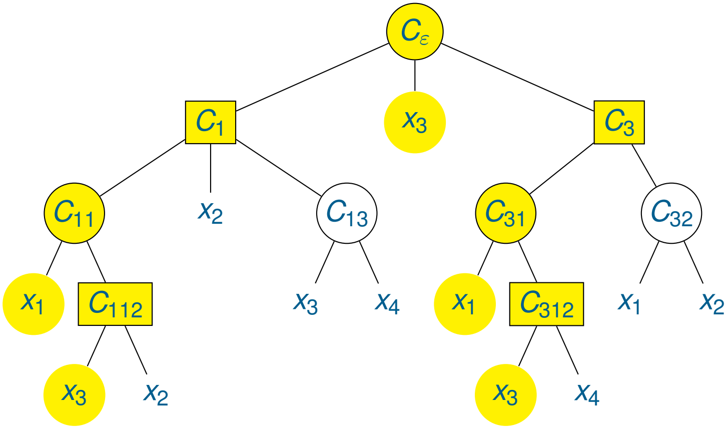
Weighted Monotone Satisfiability

$$\phi = (((x_1 \wedge (x_3 \vee x_2)) \vee x_2 \vee (x_2 \wedge x_4)) \wedge$$
$$\wedge x_3 \wedge ((x_1 \wedge (x_3 \vee x_4)) \vee (x_1 \wedge x_2)))$$



Weighted Monotone Satisfiability

$$\phi = (((x_1 \wedge (x_3 \vee x_2)) \vee x_2 \vee (x_2 \wedge x_4)) \wedge$$
$$\wedge x_3 \wedge ((x_1 \wedge (x_3 \vee x_4)) \vee (x_1 \wedge x_2)))$$



The word

$$[x_1 \cdots x_1][x_2 \cdots x_2][x_3 \cdots x_3][x_4 \cdots x_4]$$

The grammar

The grammar

$$S \rightarrow f_S(F)$$

The grammar

$$S \rightarrow f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

The grammar

$$S \to f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

$$\underbrace{\quad}_{t_0}\underbrace{[x_1 \cdots x_1]}_{s_1}\underbrace{[x_2 \cdots x_2]}_{t_1}\underbrace{[x_3 \cdots x_3]}_{s_2}\underbrace{[x_4 \cdots x_4]}_{t_2}$$

The grammar

$$S \to f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

$$F \to f_{F,i,j}(F)$$

$$\underbrace{\quad}_{t_0}[\underbrace{x_1 \cdots x_1}_{s_1}]\,\underbrace{[x_2 \cdots x_2]}_{t_1}[\underbrace{x_3 \cdots x_3}_{s_2}]\,\underbrace{[x_4 \cdots x_4]}_{t_2}$$

The grammar

$$S \rightarrow f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

$$F \rightarrow f_{F,i,j}(F)$$

$$F \rightarrow f_F(C_\varepsilon)$$

$$\underbrace{}_{t_0}\underbrace{[x_1 \cdots x_1]}_{s_1}\underbrace{[x_2 \cdots x_2]}_{t_1}\underbrace{[x_3 \cdots x_3]}_{s_2}\underbrace{[x_4 \cdots x_4]}_{t_2}$$

The grammar

$$S \to f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

$$F \to f_{F,i,j}(F)$$

$$F \to f_F(C_\varepsilon)$$

$$f_F(s_1, s_2) = (\varepsilon, \varepsilon, \varepsilon, s_1, s_2)$$

$$\underbrace{}_{t_0}\underbrace{[x_1 \cdots x_1]}_{s_1}\underbrace{[x_2 \cdots x_2]}_{t_1}\underbrace{[x_3 \cdots x_3]}_{s_2}\underbrace{[x_4 \cdots x_4]}_{t_2}$$

The grammar

$$S \to f_S(F)$$

$$f_S(t_0, t_1, t_2, s_1, s_2) = (t_0[s_1]t_1[s_2]t_3)$$

$$F \to f_{F,i,j}(F)$$

$$F \to f_F(C_\varepsilon)$$

$$f_F(s_1, s_2) = (\varepsilon, \varepsilon, \varepsilon, s_1, s_2)$$

$$C_s \to f(C_{s1})$$

$$C_s \to \bigvee f(C_{si})$$

$$C_s \to f(Next(C_s))$$

$$\underbrace{\quad}_{t_0}\underbrace{[x_1 \cdots x_1]}_{s_1}\ \underbrace{[x_2 \cdots x_2]}_{t_1}\underbrace{[x_3 \cdots x_3]}_{s_2}\ \underbrace{[x_4 \cdots x_4]}_{t_2}$$

Second parameterization: Rank

Theorem

The membership problem for LCFRS, parameterized with the rank, is W[1]-hard, even if the fan-out is fixed to 2.

The proof is by reduction from p-CLIQUE.

Third parameterization: Fan-out, rank, and derivation length

Theorem

The membership problem for LCFRS, parameterized by fan-out, rank, and derivation lenght, is W[1]-complete.

The membership proof is by reduction to SHORT CONTEXT-SENSITIVE DERIVATION.

Open problems

Other parameterizations.

Other formalisms.