Pumping lemma and Ogden lemma for displacement context-free grammars

Alexey Sorokin[1,2]

Moscow State University,
Faculty of Mathematics and Mechanics

Moscow Institute of Physics and Technology,
Faculty of Innovations and High Technologies

Developments in Language Theory'14
Ekaterinburg, August, 26th 2014

Motivation

- What class of languages is the best for natural language syntax?

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like ww.

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like ww.
- Context-sensitive — too vast. Something between?

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
    - Some syntactic patterns have the form like *ww*.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985).

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like *ww*.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985). But what are these languages?
- Some requirements:

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like ww.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985). But what are these languages?
- Some requirements:
  - Polynomial complexity.

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like ww.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985). But what are these languages?
- Some requirements:
  - Polynomial complexity.
  - Context independence (whatever “ context ” means).

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
  - Some syntactic patterns have the form like *ww*.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985). But what are these languages?
- Some requirements:
  - Polynomial complexity.
  - Context independence (whatever " context " means).
  - Constant growth property.

Motivation

- What class of languages is the best for natural language syntax?
- context-free — not enough (Shieber, 1985;)
    - Some syntactic patterns have the form like *ww*.
- Context-sensitive — too vast. Something between?
- The answer is mildly context-sensitive (Joshi, 1985). But what are these languages?
- Some requirements:
    - Polynomial complexity.
    - Context independence (whatever " context " means).
    - Constant growth property.
    - Limited number of cross-serial dependencies.

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:
- Multiple context-free? First three requirements are satisfied.

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:
- Multiple context-free? First three requirements are satisfied.
- But MIX is a 2-MCFL (Salvati, 2012)!

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:
- Multiple context-free? First three requirements are satisfied.
- But MIX is a 2-MCFL (Salvati, 2012)!
- Possible variant: well-nested multiple context-free (wMCFL) (Kanazawa, 2008).

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:
- Multiple context-free? First three requirements are satisfied.
- But MIX is a 2-MCFL (Salvati, 2012)!
- Possible variant: well-nested multiple context-free (wMCFL) (Kanazawa, 2008).
- MIX is not a 2-wMCFL (equivalently, not a tree-adjoining language) (Kanazawa, Salvati, 2012).

Mildly context-sensitive languages

- $\mathrm{MIX} = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ is not mildly context-sensitive (Bach, 1984; Kanazawa, Salvati, 2012).
- Candidates for mildly context-sensitive class:
- Multiple context-free? First three requirements are satisfied.
- But MIX is a 2-MCFL (Salvati, 2012)!
- Possible variant: well-nested multiple context-free (wMCFL) (Kanazawa, 2008).
- MIX is not a 2-wMCFL (equivalently, not a tree-adjoining language) (Kanazawa, Salvati, 2012).
- Probably, it is not a k-wMCFL for any k (Kanazawa-Salvati conjecture, 2012). But how to prove it?

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).
- Our goal

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).
- Our goal
    1. Strong pumping lemma for wMCFGs.

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).
- Our goal
  1. Strong pumping lemma for wMCFGs.
  2. Ogden's lemma for wMCFGs.

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).
- Our goal
  1. Strong pumping lemma for wMCFGs.
  2. Ogden's lemma for wMCFGs.
  3. Some examples of non-wMCFGs.

Proving non-wellnestedness

- The proof of Kanazawa and Salvati uses combinatorial and geometrical arguments. Difficult to generalize.
- Something more simple? Pumping lemma?
- Slightly weak version (Kanazawa, 2008).
- Usual version — only for TALs (Palis, Shende, 1995).
- For TALs — also Ogden's lemma (Palis, Shende, 1995).
- Our goal
  1. Strong pumping lemma for wMCFGs. Done
  2. Ogden's lemma for wMCFGs. Not yet
  3. Some examples of non-wMCFGs. Done for TALs

Basic notions

- Σ — alphabet.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0;k]$ — rank function, $k \in N$.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0; k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s \colon N \to [0; k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.
- $Op_k = \{\cdot, \odot_1, \ldots, \odot_k\}$ — operations (on tuples).

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0;k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.
- $Op_k = \{\cdot, \odot_1, \ldots, \odot_k\}$ — operations (on tuples).
- $\cdot$ — concatenation:
  $(u_0, \ldots, u_s) \cdot (v_0, \ldots, v_t) = (u_0, \ldots, u_{s-1}, u_s v_0, v_2, \ldots, v_t)$.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0; k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.
- $Op_k = \{\cdot, \odot_1, \ldots, \odot_k\}$ — operations (on tuples).
- $\cdot$ — concatenation:
  $(u_0, \ldots, u_s) \cdot (v_0, \ldots, v_t) = (u_0, \ldots, u_{s-1}, u_s v_0, v_2, \ldots, v_t)$.
- $\odot_j$ — j-intercalation: $(u_0, \ldots, u_s) \odot_j (v_0, \ldots, v_t) =$
  $(u_0, \ldots, u_{j-2}, u_{j-1} v_0, v_1, \ldots, v_{t-1}, v_t u_j, \ldots, u_s)$.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0; k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.
- $Op_k = \{\cdot, \odot_1, \ldots, \odot_k\}$ — operations (on tuples).
- $\cdot$ — concatenation:
  $(u_0, \ldots, u_s) \cdot (v_0, \ldots, v_t) = (u_0, \ldots, u_{s-1}, u_s v_0, v_2, \ldots, v_t)$.
- $\odot_j$ — j-intercalation: $(u_0, \ldots, u_s) \odot_j (v_0, \ldots, v_t) =$
  $(u_0, \ldots, u_{j-2}, u_{j-1} v_0, v_1, \ldots, v_{t-1}, v_t u_j, \ldots, u_s)$.
- This operations generate the algebra of well-nested operations on tuples.

Basic notions

- Σ — alphabet.
- N — nonterminals, $s\colon N \to [0;k]$ — rank function, $k \in N$.
- Nonterminals of rank k derive $(k+1)$-tuples of words.
- $Op_k = \{\cdot, \odot_1, \ldots, \odot_k\}$ — operations (on tuples).
- $\cdot$ — concatenation:
  $(u_0, \ldots, u_s) \cdot (v_0, \ldots, v_t) = (u_0, \ldots, u_{s-1}, u_s v_0, v_2, \ldots, v_t)$.
- $\odot_j$ — j-intercalation: $(u_0, \ldots, u_s) \odot_j (v_0, \ldots, v_t) =$
  $(u_0, \ldots, u_{j-2}, u_{j-1} v_0, v_1, \ldots, v_{t-1}, v_t u_j, \ldots, u_s)$.
- This operations generate the algebra of well-nested operations on tuples.
- Instead of tuples, gapped strings may be used.

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
  - for any $s \leqslant k$ and $u_0, \ldots, u_s \in \Sigma$ it holds that $u = (u_0, \ldots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
  - for any $s \leqslant k$ and $u_0, \ldots, u_s \in \Sigma$ it holds that $u = (u_0, \ldots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.

Terms

- $\mathrm{Tm}_k(N,\Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N,\Sigma)$,
  - for any $s \leqslant k$ and $u_0,\ldots,u_s \in \Sigma$ it holds that $u = (u_0,\ldots,u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
  - If $\alpha,\beta \in Tm_k$ and $rk(\alpha) + rk(\beta) \leqslant k$, then $(\alpha \cdot \beta) \in Tm_k, rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta)$.

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
  - for any $s \leqslant k$ and $u_0, \dots, u_s \in \Sigma$ it holds that $u = (u_0, \dots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
  - If $\alpha, \beta \in Tm_k$ and $rk(\alpha) + rk(\beta) \leqslant k$, then $(\alpha \cdot \beta) \in Tm_k, rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta)$.
  - If $j \leqslant k$, $\alpha, \beta \in Tm_k$, $rk(\alpha) + rk(\beta) \leqslant k + 1$, $rk(\alpha) \geqslant j$, then $(\alpha \odot_j \beta) \in Tm_k$, $rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta) - 1$.

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
    - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
    - for any $s \leqslant k$ and $u_0, \ldots, u_s \in \Sigma$ it holds that $u = (u_0, \ldots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
    - If $\alpha, \beta \in Tm_k$ and $rk(\alpha) + rk(\beta) \leqslant k$, then $(\alpha \cdot \beta) \in Tm_k, rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta)$.
    - If $j \leqslant k$, $\alpha, \beta \in Tm_k$, $rk(\alpha) + rk(\beta) \leqslant k + 1$, $rk(\alpha) \geqslant j$, then $(\alpha \odot_j \beta) \in Tm_k$, $rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta) - 1$.
- GrTm_k — ground terms (no nonterminals).
- Any ground term α has a value $\nu(\alpha)$.

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
  - for any $s \leqslant k$ and $u_0, \ldots, u_s \in \Sigma$ it holds that $u = (u_0, \ldots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
  - If $\alpha, \beta \in Tm_k$ and $rk(\alpha) + rk(\beta) \leqslant k$, then $(\alpha \cdot \beta) \in Tm_k, rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta)$.
  - If $j \leqslant k$, $\alpha, \beta \in Tm_k$, $rk(\alpha) + rk(\beta) \leqslant k + 1$, $rk(\alpha) \geqslant j$, then $(\alpha \odot_j \beta) \in Tm_k$, $rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta) - 1$.
- GrTm_k — ground terms (no nonterminals).
- Any ground term α has a value $\nu(\alpha)$.
- $\mathrm{Var} = \{x_1, x_2, \ldots\}$ — ranked set of free variables,

Terms

- $\mathrm{Tm}_k(N,\Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N,\Sigma)$,
  - for any $s \leqslant k$ and $u_0,\ldots,u_s \in \Sigma$ it holds that $u = (u_0,\ldots,u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
  - If $\alpha,\beta \in Tm_k$ and $rk(\alpha)+rk(\beta) \leqslant k$, then $(\alpha\cdot\beta) \in Tm_k, rk(\alpha\cdot\beta) = rk(\alpha)+rk(\beta)$.
  - If $j \leqslant k$, $\alpha,\beta \in Tm_k$, $rk(\alpha)+rk(\beta) \leqslant k+1$, $rk(\alpha) \geqslant j$, then $(\alpha \odot_j \beta) \in Tm_k$, $rk(\alpha\cdot\beta) = rk(\alpha)+rk(\beta)-1$.
- GrTm_k — ground terms (no nonterminals).
- Any ground term α has a value $\nu(\alpha)$.
- $\mathrm{Var} = \{x_1, x_2, \ldots\}$ — ranked set of free variables,
- multicontext — a term with variables in some leaves (possibly zero)

Terms

- $\mathrm{Tm}_k(N, \Sigma)$ — terms:
  - $N \subset \mathrm{Tm}_k(N, \Sigma)$,
  - for any $s \leqslant k$ and $u_0, \ldots, u_s \in \Sigma$ it holds that $u = (u_0, \ldots, u_s) \in \mathrm{Tm}_k$, $\mathrm{rk}(u) = s$.
  - If $\alpha, \beta \in Tm_k$ and $rk(\alpha) + rk(\beta) \leqslant k$, then $(\alpha \cdot \beta) \in Tm_k, rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta)$.
  - If $j \leqslant k$, $\alpha, \beta \in Tm_k$, $rk(\alpha) + rk(\beta) \leqslant k + 1$, $rk(\alpha) \geqslant j$, then $(\alpha \odot_j \beta) \in Tm_k$, $rk(\alpha \cdot \beta) = rk(\alpha) + rk(\beta) - 1$.
- GrTm_k — ground terms (no nonterminals).
- Any ground term α has a value $\nu(\alpha)$.
- $\mathrm{Var} = \{x_1, x_2, \ldots\}$ — ranked set of free variables,
- multicontext — a term with variables in some leaves (possibly zero)
- $C[x_1, \ldots, x_t]$ — a multicontext, $\alpha_1, \ldots, \alpha_t$ — terms, s.t $\forall j \; (rk(\alpha_j) = \mathrm{rk}(x_j))$, then $C[\alpha_1, \ldots, \alpha_t]$ — substitution of all x_j-s by α_j-s.

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- Σ — alphabet,

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- - Σ — alphabet,
  N - ranked set of nonterminals,

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- Σ — alphabet,
  N - ranked set of nonterminals,
  $S \in N$ — initial nonterminal, $s(N) = 0$,

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- - Σ — alphabet,
    N - ranked set of nonterminals,
    $S \in N$ — initial nonterminal, $s(N) = 0$,
    P — the set of rules of the form
    $A \to \alpha$, $A \in N, \alpha \in \mathrm{Tm}_k, s(A) = s(\alpha)$.

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- Σ — alphabet,
  N - ranked set of nonterminals,
  $S \in N$ — initial nonterminal, $s(N) = 0$,
  P — the set of rules of the form
  $A \to \alpha,\ A \in N, \alpha \in \mathrm{Tm}_k, s(A) = s(\alpha)$.

- $\vdash_G \in N \times Tm_k$ — the smallest reflexive transitive relation, s.t $(B \to \beta) \in P$ and $A \vdash C[B]$ imply $A \vdash C[\beta]$ for any context C.

Displacement context-free grammars

Definition

A k-displacement context-free grammar is $G = \langle N, \Sigma, P, S \rangle$, where

- Σ — alphabet,
  N - ranked set of nonterminals,
  $S \in N$ — initial nonterminal, $s(N) = 0$,
  P — the set of rules of the form
  $A \to \alpha,\ A \in N, \alpha \in \mathrm{Tm}_k, s(A) = s(\alpha)$.

- $\vdash_G \in N \times Tm_k$ — the smallest reflexive transitive relation, s.t $(B \to \beta) \in P$ and $A \vdash C[B]$ imply $A \vdash C[\beta]$ for any context C.
- $L_G(A) = \{\nu(\alpha) \mid A \vdash_G \alpha,\ \alpha \in GrTm_k\},\quad L(G) = L_G(S)$.

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} .$$

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} \; .$$

$$L(G_i) = \{ w^{i+1} \mid w \in \Sigma^+ \}$$

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$
Here is the derivation of $(aba)^3$ in G_2:

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S\rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a\rangle) + \ldots) \odot_i \langle \varepsilon, a\rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b\rangle + \ldots) \odot_i \langle \varepsilon, b\rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon\rangle}_{(i+1) \text{ times}} \; .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$
Here is the derivation of $(aba)^3$ in G_2:
S.

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$
Here is the derivation of $(aba)^3$ in G_2:
$S \to (aT \odot_1 a) \odot_1 a$.

Example

$G_i = \langle\{S, T\}, \{a, b\}, P_i, S\rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle\varepsilon, a\rangle) + \ldots) \odot_i \langle\varepsilon, a\rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle\varepsilon, b\rangle + \ldots) \odot_i \langle\varepsilon, b\rangle \mid \underbrace{\langle\varepsilon, \ldots, \varepsilon\rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$
Here is the derivation of $(aba)^3$ in G_2:
$S \to (aT \odot_1 a) \odot_1 a \to (a((bT \odot_1 \langle\varepsilon, b\rangle) \odot_2 \langle\varepsilon, b\rangle) \odot_1 a) \odot_1 a.$

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \quad aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \quad bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$
Here is the derivation of $(aba)^3$ in G_2:
$S \to (aT \odot_1 a) \odot_1 a \to (a((bT \odot_1 \langle \varepsilon, b \rangle) \odot_2 \langle \varepsilon, b \rangle) \odot_1 a) \odot_1 a \to$
$(a((b((aT \odot_1 \langle \varepsilon, a \rangle) \odot_2 \langle \varepsilon, a \rangle) \odot_1 \langle \varepsilon, b \rangle) \odot_2 \langle \varepsilon, b \rangle) \odot_1 a) \odot_1 a \to .$

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S\rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a\rangle) + \ldots) \odot_i \langle \varepsilon, a\rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b\rangle + \ldots) \odot_i \langle \varepsilon, b\rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon\rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$

Here is the derivation of $(aba)^3$ in G_2:

$S \to (aT \odot_1 a) \odot_1 a \to (a((bT \odot_1 \langle \varepsilon, b\rangle) \odot_2 \langle \varepsilon, b\rangle) \odot_1 a) \odot_1 a \to$
$(a((b((aT \odot_1 \langle \varepsilon, a\rangle) \odot_2 \langle \varepsilon, a\rangle) \odot_1 \langle \varepsilon, b\rangle) \odot_2 \langle \varepsilon, b\rangle) \odot_1 a) \odot_1 a \to$
$(a((b((\langle a, \varepsilon, \varepsilon\rangle \odot_1 \langle \varepsilon, a\rangle) \odot_2 \langle \varepsilon, a\rangle) \odot_1 \langle \varepsilon, b\rangle) \odot_2 \langle \varepsilon, b\rangle) \odot_1 a)$.

Example

$G_i = \langle \{S, T\}, \{a, b\}, P_i, S \rangle$ with $\mathrm{rk}(T) = i$ and the following P_i:

$$S \to \underbrace{(\ldots(}_{i-1 \text{ times}} \; aT \odot_1 a) + \ldots) \odot_1 a \mid \underbrace{(\ldots(}_{i-1 \text{ times}} \; bT \odot_1 b) + \ldots) \odot_1 b,$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} aT \odot_1 \langle \varepsilon, a \rangle) + \ldots) \odot_i \langle \varepsilon, a \rangle$$

$$T \to \underbrace{(\ldots(}_{i-1 \text{ times}} bT \odot_1 \langle \varepsilon, b \rangle + \ldots) \odot_i \langle \varepsilon, b \rangle \mid \underbrace{\langle \varepsilon, \ldots, \varepsilon \rangle}_{(i+1) \text{ times}} .$$

$L(G_i) = \{w^{i+1} \mid w \in \Sigma^+\}$

Here is the derivation of $(aba)^3$ in G_2:

$S \to (aT \odot_1 a) \odot_1 a \to (a((bT \odot_1 \langle \varepsilon, b \rangle) \odot_2 \langle \varepsilon, b \rangle) \odot_1 a) \odot_1 a \to$
$(a((b((aT \odot_1 \langle \varepsilon, a \rangle) \odot_2 \langle \varepsilon, a \rangle) \odot_1 \langle \varepsilon, b \rangle) \odot_2 \langle \varepsilon, b \rangle) \odot_1 a) \odot_1 a \to$
$(a((b((\langle a, \varepsilon, \varepsilon \rangle \odot_1 \langle \varepsilon, a \rangle) \odot_2 \langle \varepsilon, a \rangle) \odot_1 \langle \varepsilon, b \rangle) \odot_2 \langle \varepsilon, b \rangle) \odot_1 a) =$
$(a(b(\langle a, a, a \rangle \odot_1 (\langle \varepsilon, b \rangle)) +_2 (\langle \varepsilon, b \rangle)) +_1 a) \odot_1 a =$
$((\langle aba, ba, ba \rangle) \odot_1 a) \odot_1 a = abaabaaba.$

Chomsky normal form I

Two grammars are equivalent if they generate the same language.

Theorem (Chomsky normal form)

Every k-DCFG is equivalent to a k-DCFG $G = \langle N, \Sigma, P, S \rangle$ with the rules only of the form:

1. $A \to B \cdot C$, *where* $A \in N$, $B, C \in N - \{S\}$,
2. $A \to B \odot_j C$, *where* $j \leqslant k$, $A \in N$, $B, C \in N - \{S\}$,
3. $A \to a$ *or* $A \to \langle \varepsilon, \varepsilon \rangle$, *where* $A \in N$, $a \in \Sigma$,
4. $S \to \varepsilon$.

Chomsky normal form II

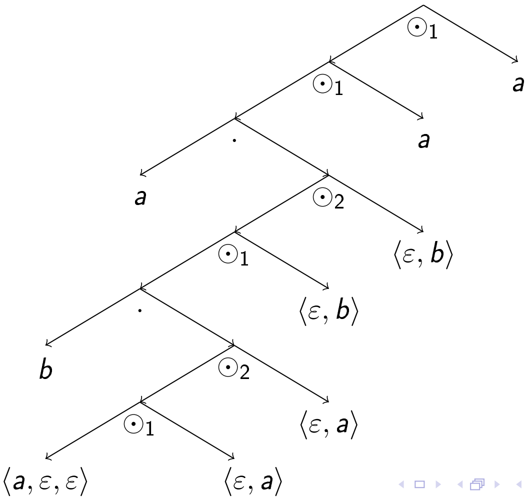
Example

The grammar
$G'_2 = \langle \{S, T_2, U_1, V_1, U_2, V_2, W_2, X_2, A_1, B_1, A, B\}, \{a, b\}, P, S\rangle$
with the following set of rules generates the language
$\{www \mid w \in \Sigma^+\}$.

$$
\begin{array}{rcl@{\qquad}rcl}
S & \to & U_1 \odot_1 A & S & \to & V_1 \odot_1 B \\
U_1 & \to & U_2 \odot_1 A & V_1 & \to & V_2 \odot_1 B \\
U_2 & \to & A \cdot T & V_2 & \to & B \cdot T \\
T_2 & \to & W_2 \odot_2 A_1 & T_2 & \to & X_2 \odot_2 B_1 \\
W_2 & \to & U_2 \odot_1 A_1 & X_2 & \to & V_2 \odot B_1 \\
U_2 & \to & \langle a, \varepsilon, \varepsilon\rangle & V_2 & \to & \langle b, \varepsilon, \varepsilon\rangle \\
A_1 & \to & \langle a, \varepsilon\rangle & B_1 & \to & \langle b, \varepsilon\rangle \\
A & \to & a & B & \to & b
\end{array}
$$

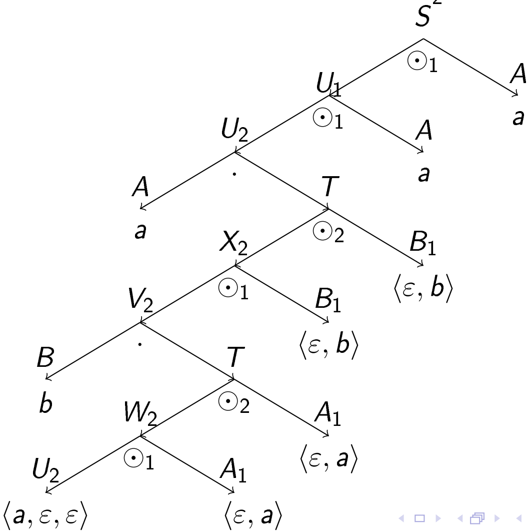
Derivation tree

The derivation tree of the word *abaabaaba* in G_2':



Derivation tree

The derivation tree of the word *abaabaaba* in G_2':



Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.
- If B is a label of a subtree, corresponding to a β, then $B \vdash \beta$.

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.
- If B is a label of a subtree, corresponding to a β, then $B \vdash \beta$.
- If $A \vdash C[\beta_1, \ldots, \beta_t]$ then there are nonterminals $B_1, \ldots, B_t$ such $A \vdash C[B_1, \ldots, B_t]$ and $\forall i \; (B_i \vdash \beta_i)$.

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.
- If B is a label of a subtree, corresponding to a β, then $B \vdash \beta$.
- If $A \vdash C[\beta_1, \dots, \beta_t]$ then there are nonterminals $B_1, \dots, B_t$ such $A \vdash C[B_1, \dots, B_t]$ and $\forall i \; (B_i \vdash \beta_i)$.
- If $\alpha = C[\beta]$ for some ground context C and $rk(\beta) = l$, then there are tuples $s_1, s_2, u_1, \dots, u_l$ such that $\alpha = s_1(\beta \otimes (u_1, \dots, u_l))s_t$. ($\otimes$ denotes simultaneous replacement of all the separators)

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.
- If B is a label of a subtree, corresponding to a β, then $B \vdash \beta$.
- If $A \vdash C[\beta_1, \ldots, \beta_t]$ then there are nonterminals $B_1, \ldots, B_t$ such $A \vdash C[B_1, \ldots, B_t]$ and $\forall i\ (B_i \vdash \beta_i)$.
- If $\alpha = C[\beta]$ for some ground context C and $rk(\beta) = l$, then there are tuples $s_1, s_2, u_1, \ldots, u_l$ such that $\alpha = s_1(\beta \otimes (u_1, \ldots, u_l))s_t$. ($\otimes$ denotes simultaneous replacement of all the separators)

Definition

A descendent v of a node u is direct, if all the nodes on the path from u to v have the same rank (including u and v). β is a direct subterm of α, if its root is a direct descendant of the root of α.

Properties of trees and terms

- Downward-closed subtrees correspond to subterms of the derived term, and vice versa (so we may speak about nodes' ranks).
- Well-formed subtrees correspond to terms.
- If B is a label of a subtree, corresponding to a β, then $B \vdash \beta$.
- If $A \vdash C[\beta_1, \ldots, \beta_t]$ then there are nonterminals $B_1, \ldots, B_t$ such $A \vdash C[B_1, \ldots, B_t]$ and $\forall i\ (B_i \vdash \beta_i)$.
- If $\alpha = C[\beta]$ for some ground context C and $rk(\beta) = l$, then there are tuples $s_1, s_2, u_1, \ldots, u_l$ such that $\alpha = s_1(\beta \otimes (u_1, \ldots, u_l))s_t$. ($\otimes$ denotes simultaneous replacement of all the separators)

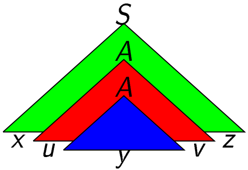
Definition

A descendent v of a node u is direct, if all the nodes on the path from u to v have the same rank (including u and v). β is a direct subterm of α, if its root is a direct descendant of the root of α.

Lemma

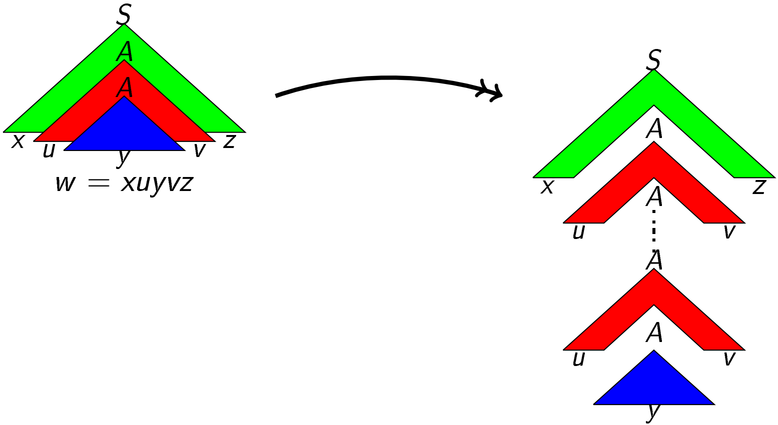
If $\alpha = C[\beta]$ and β is a direct subterm of α, then there are words $s_1, s_2, u_1, v_1, \ldots, u_l, v_l$, s.t. $\alpha = s_1(\beta \otimes (\langle u_1, v_1 \rangle, \ldots, \langle u_l, v_l \rangle))s_2$.

Pumping lemma: reminder

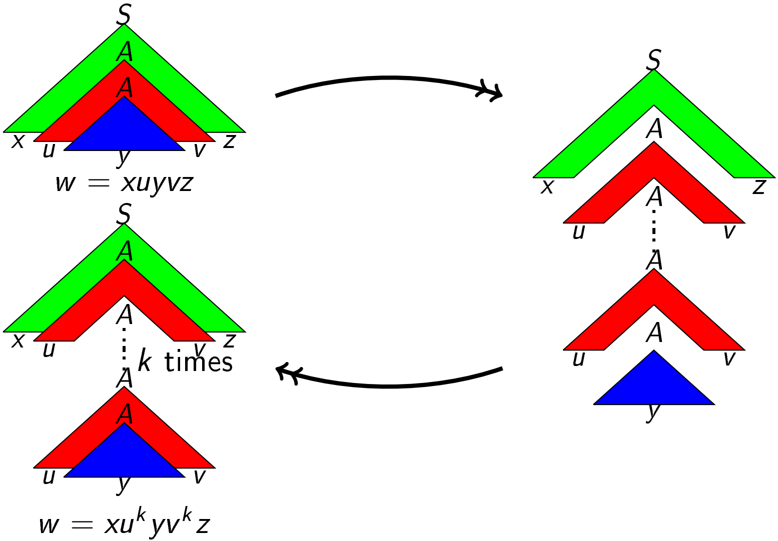


$w = xuyvz$

Pumping lemma: reminder



Pumping lemma: reminder



Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.

Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.
We expect: for any recursive terminal A with rank s of a k-DCFG G there are words $u_0, \ldots, u_{rk(A)}, v_0, \ldots, v_{rk(A)}$ s.t. $u_0 \ldots u_s v_0 \ldots v_s \neq \varepsilon$ and $A \vdash_G \langle y_0, \ldots, y_s \rangle$ implies $A \vdash_G \langle u_0 y_0 v_0, \ldots, u_s y_s v_s \rangle$.

Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.

We expect: for any recursive terminal A with rank s of a k-DCFG G there are words $u_0, \ldots, u_{rk(A)}, v_0, \ldots, v_{rk(A)}$ s.t. $u_0 \ldots u_s v_0 \ldots v_s \neq \varepsilon$ and $A \vdash_G \langle y_0, \ldots, y_s \rangle$ implies $A \vdash_G \langle u_0 y_0 v_0, \ldots, u_s y_s v_s \rangle$.

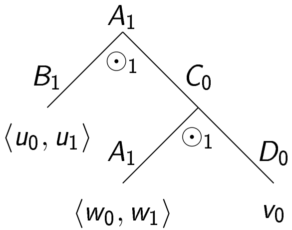
That's not the case!

Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.

We expect: for any recursive terminal A with rank s of a k-DCFG G there are words $u_0, \ldots, u_{rk(A)}, v_0, \ldots, v_{rk(A)}$ s.t. $u_0 \ldots u_s v_0 \ldots v_s \neq \varepsilon$ and $A \vdash_G \langle y_0, \ldots, y_s \rangle$ implies $A \vdash_G \langle u_0 y_0 v_0, \ldots, u_s y_s v_s \rangle$.

That's not the case!

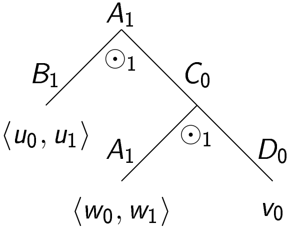


Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.

We expect: for any recursive terminal A with rank s of a k-DCFG G there are words $u_0, \ldots, u_{rk(A)}, v_0, \ldots, v_{rk(A)}$ s.t. $u_0 \ldots u_s v_0 \ldots v_s \neq \varepsilon$ and $A \vdash_G \langle y_0, \ldots, y_s \rangle$ implies $A \vdash_G \langle u_0 y_0 v_0, \ldots, u_s y_s v_s \rangle$.

That's not the case!



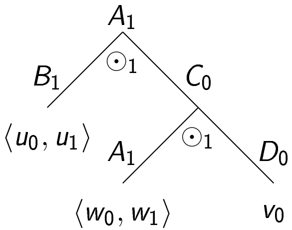
In this case $A \vdash \langle w_0, w_1 \rangle$ implies $A \vdash_G \langle u_0, u_1 w_0 v_0 w_1 \rangle$.

Problems for DCFGs

For any recursive terminal A of a CFG G there are words u, v s.t. $uv \neq \varepsilon$ and $A \vdash_G y$ implies $A \vdash_G uyv$.

We expect: for any recursive terminal A with rank s of a k-DCFG G there are words $u_0, \dots, u_{rk(A)}, v_0, \dots, v_{rk(A)}$ s.t. $u_0 \dots u_s v_0 \dots v_s \neq \varepsilon$ and $A \vdash_G \langle y_0, \dots, y_s \rangle$ implies $A \vdash_G \langle u_0 y_0 v_0, \dots, u_s y_s v_s \rangle$.

That's not the case!



In this case $A \vdash \langle w_0, w_1 \rangle$ implies $A \vdash_G \langle u_0, u_1 w_0 v_0 w_1 \rangle$.

That's because the foot node of the pumped subtree is not a direct descendant of its root.

Equivalence of terms

- $\nu : \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.

Equivalence of terms

- $\nu \colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.

Equivalence of terms

- $\nu\colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.
- Extended to ground multicontexts in a natural way.

Equivalence of terms

- $\nu\colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.
- Extended to ground multicontexts in a natural way.
- Ground multicontexts C_1 and C_2 are equivalent ($C_1 \sim C_2$) iff $\forall \nu \, (\nu(C_1) = \nu(C_2))$. Terms α and β are equivalent if $\alpha = C_1[A_1, \ldots, A_t]$, $\beta = C_2[A_1, \ldots, A_t]$ and $C_1 \sim C_2$. Two syntactic trees are equivalent if they correspond to equivalent terms.

Equivalence of terms

- $\nu\colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.
- Extended to ground multicontexts in a natural way.
- Ground multicontexts C_1 and C_2 are equivalent ($C_1 \sim C_2$) iff $\forall \nu\, (\nu(C_1) = \nu(C_2))$. Terms α and β are equivalent if $\alpha = C_1[A_1, \ldots, A_t]$, $\beta = C_2[A_1, \ldots, A_t]$ and $C_1 \sim C_2$. Two syntactic trees are equivalent if they correspond to equivalent terms.
- Note: replacing a well-formed subtree by an equivalent one doesn't change the word corresponding to the tree.

Equivalence of terms

- $\nu\colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.
- Extended to ground multicontexts in a natural way.
- Ground multicontexts C_1 and C_2 are equivalent ($C_1 \sim C_2$) iff $\forall \nu\ (\nu(C_1) = \nu(C_2))$. Terms α and β are equivalent if $\alpha = C_1[A_1, \ldots, A_t]$, $\beta = C_2[A_1, \ldots, A_t]$ and $C_1 \sim C_2$. Two syntactic trees are equivalent if they correspond to equivalent terms.
- Note: replacing a well-formed subtree by an equivalent one doesn't change the word corresponding to the tree.
- A term is l-essential, if its root and its leaves are of rank l or less.

Equivalence of terms

- $\nu\colon \mathrm{Var} \to \mathcal{B}((\Sigma^*)^*)$ — valuation, if $\nu(A) \subseteq (\Sigma^*)^{rk(A)+1}$ for any $A \in N \cup \mathrm{Var}$.
- For any $u \in (\Sigma^*)^+$ $\nu(u) = u$.
- Extended to ground multicontexts in a natural way.
- Ground multicontexts C_1 and C_2 are equivalent ($C_1 \sim C_2$) iff $\forall \nu\ (\nu(C_1) = \nu(C_2)$. Terms α and β are equivalent if $\alpha = C_1[A_1, \ldots, A_t]$, $\beta = C_2[A_1, \ldots, A_t]$ and $C_1 \sim C_2$. Two syntactic trees are equivalent if they correspond to equivalent terms.
- Note: replacing a well-formed subtree by an equivalent one doesn't change the word corresponding to the tree.
- A term is l-essential, if its root and its leaves are of rank l or less.

Lemma

Every l-essential term α is equivalent to some l-correct β.

Removing bad subtrees

- A term is I-redundant if it is I-essential, but not I-correct.

Removing bad subtrees

- A term is l-redundant if it is l-essential, but not l-correct.
- Let N_l denote the number of nonterminals of rank l.

Removing bad subtrees

- A term is l-redundant if it is l-essential, but not l-correct.
- Let N_l denote the number of nonterminals of rank l.
- A grammar G is l-duplicated if for any derivable rule $A \to \alpha$ with l-redundant α with $depth(\alpha) \leqslant N_l + 1$ there is a derivable rule $A \to \alpha'$, where $\alpha \sim \alpha'$ and $\alpha \in \mathrm{Tm}_l$. <4->

Lemma

For any k-DCFG G and any $l \leqslant k$ there is an equivalent l-duplicated k-DCFG G' with the same set of nonterminals of rank l and greater.

Compactness

Definition

A subbranch in a syntactic tree is an l-matreshka, iff all the nodes of it have rank l and its length is at least $N_l + 1$.

Compactness

Definition

A subbranch in a syntactic tree is an l-matreshka, iff all the nodes of it have rank l and its length is at least $N_l + 1$.

Definition

A vicinity of a node v if the largest well-formed subtree containing T such that all its internal nodes have the same rank as v.

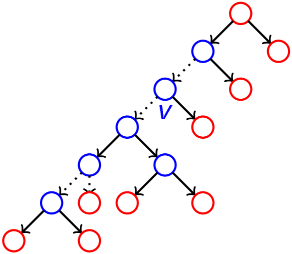
Compactness

Definition

A subbranch in a syntactic tree is an l-matreshka, iff all the nodes of it have rank l and its length is at least $N_l + 1$.

Definition

A vicinity of a node v if the largest well-formed subtree containing T such that all its internal nodes have the same rank as v.



blue — the same rank as v
red — rank, diverse from v

Compactness

Definition

A subbranch in a syntactic tree is an l-matreshka, iff all the nodes of it have rank l and its length is at least $N_l + 1$.

Definition

A vicinity of a node v if the largest well-formed subtree containing T such that all its internal nodes have the same rank as v.

Definition

A derivation tree is m-compact, if for any node v of any rank l there is a path from l to an element of an l' - matreshka with $l' \geqslant l$ such that its length is not greater than l and all the nodes on it are of rank l or greater.

Compactness

Definition

A subbranch in a syntactic tree is an l-matreshka, iff all the nodes of it have rank l and its length is at least $N_l + 1$.

Definition

A vicinity of a node v if the largest well-formed subtree containing T such that all its internal nodes have the same rank as v.

Definition

A derivation tree is m-compact, if for any node v of any rank l there is a path from l to an element of an l' - matreshka with $l' \geqslant l$ such that its length is not greater than l and all the nodes on it are of rank l or greater.

Lemma

For any k-DCFG G for some m there is an equivalent m-compact k-DCFG.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.
- Consider $l := l - 1$, $m := m + 2N_l$ and duplicate all $(l-1)$-redundant rules.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.
- Consider $l := l - 1,\ m := m + 2N_l$ and duplicate all $(l - 1)$-redundant rules.
- Take any node v of rank l and consider its vicinity.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.
- Consider $l := l - 1,\ m := m + 2N_l$ and duplicate all $(l - 1)$-redundant rules.
- Take any node v of rank l and consider its vicinity.
- It is deeper than $N_l + 1$ — it contains an l-matreshka.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.
- Consider $l := l - 1$, $m := m + 2N_l$ and duplicate all $(l - 1)$-redundant rules.
- Take any node v of rank l and consider its vicinity.
- It is deeper than $N_l + 1$ — it contains an l-matreshka.
- If it contains some node of greater rank: use induction hypothesis.

Sketch of the proof

- Proceed by downwards induction on l, start with $l = k + 1$ and $m = 0$.
- Consider $l := l - 1$, $m := m + 2N_l$ and duplicate all $(l - 1)$-redundant rules.
- Take any node v of rank l and consider its vicinity.
- It is deeper than $N_l + 1$ — it contains an l-matreshka.
- If it contains some node of greater rank: use induction hypothesis.
- Otherwise it is l-redundant — replace the subtree by an equivalent one, which is l-correct.

Pumping lemma

Theorem (Pumping lemma for k-DCFGs)

For any k-DCFG G there is a number n such that for any word $w \in L(G)$, such that $|w| \geqslant n$, there is a decomposition $w = s_0 u_1 x_1 v_1 s_1 \ldots u_k x_k v_k s_k$, satisfying:

1. $\sum_{i=1}^{k} |u_i x_i v_i| \leqslant n, \quad u_1 v_1 \ldots u_k v_k \neq \varepsilon,$
2. *For any* $m \in \mathbb{N}$ $s_0 u_1^m x_1 v_1^m s_1 \ldots u_k^m x_k v_k^m s_k \in L(G)$.

Pumping lemma

Theorem (Pumping lemma for k-DCFGs)

For any k-DCFG G there is a number n such that for any word $w \in L(G)$, such that $|w| \geqslant n$, there is a decomposition $w = s_0 u_1 x_1 v_1 s_1 \dots u_k x_k v_k s_k$, satisfying:

1. $\sum_{i=1}^{k} |u_i x_i v_i| \leqslant n, \quad u_1 v_1 \dots u_k v_k \neq \varepsilon,$
2. *For any* $m \in \mathbb{N}$ $s_0 u_1^m x_1 v_1^m s_1 \dots u_k^m x_k v_k^m s_k \in L(G)$.

Theorem (Ogden lemma for 1-DCFGs (Palis, Shende, 1995))

For any TAG G there is a number n such that for any word winL(G) with at least n marked positions, such that $|w| \geqslant n$, there is a decomposition $w = s_0 u_1 x_1 v_1 s_1 u_2 x_2 v_2 s_2$, satisfying:

1. $\sum_{i=1}^{k} |u_i x_i v_i| \leqslant n, \quad u_1 v_1 u_2 v_2 \neq \varepsilon,$
2. *There is at least one marked position in one of* u_1, v_1, u_2, v_2.
3. *There is at least one marked position in one of* x_1, x_2.

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i;j] \cap [i';j'] = \varnothing$ or $[i;j] \subseteq [i';j']$.

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i;j] \cap [i';j'] = \varnothing$ or $[i;j] \subseteq [i';j']$. Geometrically:

Possible: $\underset{i'}{[}\ \underset{i}{[}\quad \underset{j}{]}\ \underset{j'}{]}$

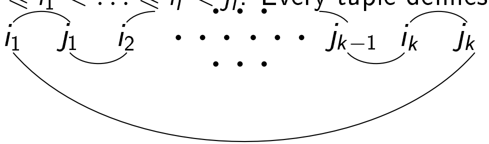
Impossible: $\underset{i}{[}\ \underset{i'}{[}\quad \underset{j}{]}\ \underset{j'}{]}$

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i; j] \cap [i'; j'] = \varnothing$ or $[i; j] \subseteq [i'; j']$.
- DCFG constituents of rank l are characterized by $2(l+1)$ numbers $i_0 < j_0 \leqslant i_1 < \ldots \leqslant i_l < j_l$.

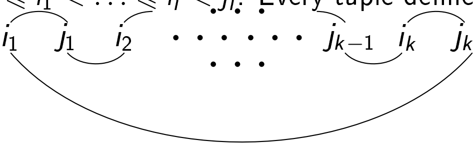
Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i;j] \cap [i';j'] = \varnothing$ or $[i;j] \subseteq [i';j']$.
- DCFG constituents of rank l are characterized by $2(l+1)$ numbers $i_0 < j_0 \leqslant i_1 < \ldots \leqslant i_l < j_l$. Every tuple defines a curve:



Geometry of constituents

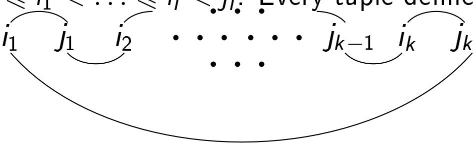
- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i;j] \cap [i';j'] = \varnothing$ or $[i;j] \subseteq [i';j']$.
- DCFG constituents of rank l are characterized by $2(l+1)$ numbers $i_0 < j_0 \leqslant i_1 < \ldots \leqslant i_l < j_l$. Every tuple defines a curve:



- Regions inside different curves are either embedded or disjoint.

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i; j] \cap [i'; j'] = \varnothing$ or $[i; j] \subseteq [i'; j']$.
- DCFG constituents of rank l are characterized by $2(l+1)$ numbers $i_0 < j_0 \leqslant i_1 < \ldots \leqslant i_l < j_l$. Every tuple defines a curve:



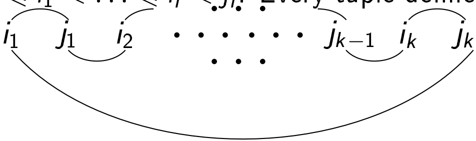
- Regions inside different curves are either embedded or disjoint.

Possible: $[_1\ [_1 \quad]_1\ ]_1 \quad [_2 \quad]_2 \quad [_3\ [_2 \quad]_2\ ]_3$

$i_1\ i'_1 \quad j'_1\ j_1 \quad i_2 \quad j_2 \quad i_3\ i'_2 \quad j'_2\ j_3$

Geometry of constituents

- Any constituent of a context-free derivation tree of a word w is specified by two positions $0 \leqslant i < j \leqslant |w|$.
- Constituens should be properly embedded: either $[i;j] \cap [i';j'] = \varnothing$ or $[i;j] \subseteq [i';j']$.
- DCFG constituents of rank l are characterized by $2(l+1)$ numbers $i_0 < j_0 \leqslant i_1 < \ldots \leqslant i_l < j_l$. Every tuple defines a curve:



- Regions inside different curves are either embedded or disjoint.

Possible: $[_1 \; [_1 \quad]_1 \;]_1 \quad [_2 \quad]_2 \quad [_3 \; [_2 \quad]_2 \;]_3$

$i_1 \; i_1' \quad j_1' \; j_1 \quad i_2 \quad j_2 \quad i_3 \; i_2' \quad j_2' \; j_3$

Impossible: $[_1 \quad]_1 \quad [_1 \quad]_1 \quad [_2 \quad]_2 \quad [_2 \quad]_2 \quad [_3 \quad]_3$

$i_1 \quad j_1 \quad i_1' \quad j_1' \quad i_2 \quad j_2 \quad i_2' \quad j_2' \quad i_3 \quad j_3$

Geometry of pumps

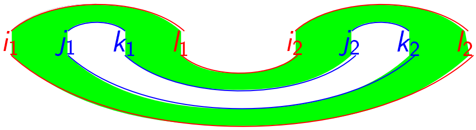
- Every pump is a difference of two constituents.

Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:



Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

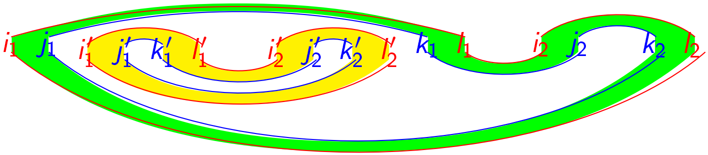
Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$



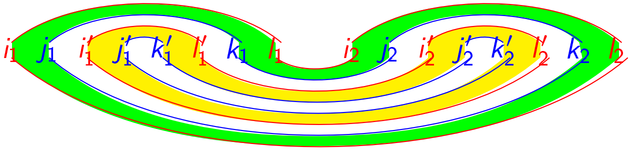
Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$
2. $j_1 \leqslant i'_1 \leqslant l'_1 \leqslant k_1 \leqslant j_2 \leqslant i'_2 \leqslant l'_2 \leqslant k_2$



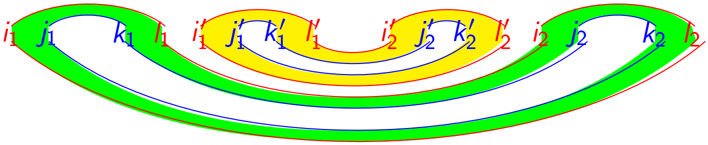
Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$
2. $j_1 \leqslant i'_1 \leqslant l'_1 \leqslant k_1 \leqslant j_2 \leqslant i'_2 \leqslant l'_2 \leqslant k_2$
3. $l_1 \leqslant i'_1 \leqslant l'_2 \leqslant i_2$



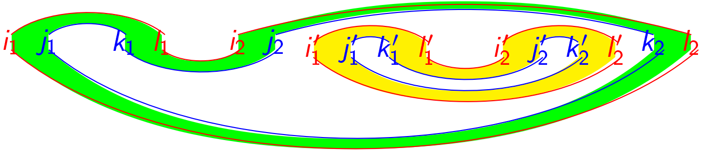
Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$
2. $j_1 \leqslant i'_1 \leqslant l'_1 \leqslant k_1 \leqslant j_2 \leqslant i'_2 \leqslant l'_2 \leqslant k_2$
3. $l_1 \leqslant i'_1 \leqslant l'_2 \leqslant i_2$
4. $j_2 \leqslant i'_1 \leqslant l'_2 \leqslant k_2$



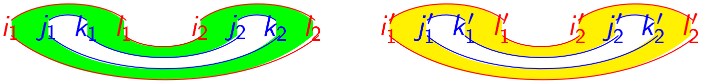
Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$
2. $j_1 \leqslant i'_1 \leqslant l'_1 \leqslant k_1 \leqslant j_2 \leqslant i'_2 \leqslant l'_2 \leqslant k_2$
3. $l_1 \leqslant i'_1 \leqslant l'_2 \leqslant i_2$
4. $j_2 \leqslant i'_1 \leqslant l'_2 \leqslant k_2$
5. $l_2 \leqslant i'_1$



Geometry of pumps

- Every pump is a difference of two constituents.
- So a 2-DCFG pump is characterized by 8 numbers $i_1 \leqslant j_1 \leqslant k_1 \leqslant l_1 \leqslant i_2 \leqslant j_2 \leqslant k_2 \leqslant l_2$:

Lemma

If the pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ do not have common inner points, then one of the following conditions hold:

1. $j_1 \leqslant i'_1 \leqslant l'_2 \leqslant k_1$
2. $j_1 \leqslant i'_1 \leqslant l'_1 \leqslant k_1 \leqslant j_2 \leqslant i'_2 \leqslant l'_2 \leqslant k_2$
3. $l_1 \leqslant i'_1 \leqslant l'_2 \leqslant i_2$
4. $j_2 \leqslant i'_1 \leqslant l'_2 \leqslant k_2$
5. $l_2 \leqslant i'_1$

Note: For any pumps $\pi = \langle i_1, \ldots, l_2 \rangle$ and $\pi' = \langle i'_1, \ldots, l'_2 \rangle$ either $l_2 \leqslant i'_1$ or $i_1 \leqslant i'_1 \leqslant l'_2 \leqslant l_2$.

4MIX language

- We prove that $4\mathrm{MIX} = \{w \in \{a, b, c, d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL.

4MIX language

- We prove that $4\mathrm{MIX} = \{w \in \{a, b, c, d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL. Sufficient for $4\mathrm{MIX} \cap (a^+ b^+ c^+ d^+)^2$.

4MIX language

- We prove that $\mathrm{4MIX} = \{w \in \{a, b, c, d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL. Sufficient for $\mathrm{4MIX} \cap (a^+ b^+ c^+ d^+)^2$.
- Every word in this language consists of 8 maximal homogeneous fragments, each pump intersects with exactly 4 such fragments and has equal number of a-s, b-s, c-s and d-s.

4MIX language

- We prove that $4\mathrm{MIX} = \{w \in \{a, b, c, d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL. Sufficient for $4\mathrm{MIX} \cap (a^+b^+c^+d^+)^2$.
- Every word in this language consists of 8 maximal homogeneous fragments, each pump intersects with exactly 4 such fragments and has equal number of a-s, b-s, c-s and d-s. We call a pump intersecting with fragments $i_1, \ldots, i_r$ a $[i_1, \ldots, i_r]$-pump.

4MIX language

- We prove that $4\mathrm{MIX} = \{w \in \{a,b,c,d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL. Sufficient for $4\mathrm{MIX} \cap (a^+b^+c^+d^+)^2$.
- Every word in this language consists of 8 maximal homogeneous fragments, each pump intersects with exactly 4 such fragments and has equal number of a-s, b-s, c-s and d-s. We call a pump intersecting with fragments $i_1, \ldots, i_r$ a $[i_1, \ldots, i_r]$-pump.
- Consider the word $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$ with
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

4MIX language

- We prove that $4\mathrm{MIX} = \{w \in \{a, b, c, d\}^* \mid |w|_a = |w|_b = |w|_c = |w|_d\}$ is not a 1-DCFL. Sufficient for $4\mathrm{MIX} \cap (a^+b^+c^+d^+)^2$.
- Every word in this language consists of 8 maximal homogeneous fragments, each pump intersects with exactly 4 such fragments and has equal number of a-s, b-s, c-s and d-s. We call a pump intersecting with fragments $i_1, \ldots, i_r$ a $[i_1, \ldots, i_r]$-pump.
- Consider the word $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$ with
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.
- We want to prove the existence of a $[1, 3, 6, 8]$-pump.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a\ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a\ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \ldots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a\ldots a}_{t}$$

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a \ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \ldots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \ldots a}_{t}$$

- If two groups belong to the same pump, then there are at least $(M+2)$ a-s in the pump.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a \ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \ldots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \ldots a}_{t}$$

- If two groups belong to the same pump, then there are at least $(M+2)$ a-s in the pump. It cannot be a $[1,2], [1,4]$ or $[1,7]$-pump (m_2, n_3, m_4 are too small), then it is a $[1,3,6,8]$-pump.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
    1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
    2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
    3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a\ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a\ldots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \ldots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a\ldots a}_{t}$$

- If two groups belong to the same pump, then there are at least $(M+2)$ a-s in the pump. It cannot be a $[1,2], [1,4]$ or $[1,7]$-pump (m_2, n_3, m_4 are too small), then it is a $[1,3,6,8]$-pump.
- If all the groups belong to different pumps, then there are at least $M+2$ non-intersecting $[1,2]$ pumps, but m_2 is too small. Again, there exists a $[1,3,6,8]$-pump.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
    1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
    2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
    3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a \dots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \dots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \dots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \dots a}_{t}$$

- If two groups belong to the same pump, then there are at least $(M+2)$ a-s in the pump. It cannot be a $[1,2]$, $[1,4]$ or $[1,7]$-pump (m_2, n_3, m_4 are too small), then it is a $[1,3,6,8]$-pump.
- If all the groups belong to different pumps, then there are at least $M+2$ non-intersecting $[1,2]$ pumps, but m_2 is too small. Again, there exists a $[1,3,6,8]$-pump.
- By analogous arguments, there is a $[1,4]$-pump.

4MIX language II

- $w = a^{m_1} b^{m_2} c^{m_3} d^{m_4} a^{n_1} b^{n_2} c^{n_3} d^{n_4}$,
  1. $\min(m_j, n_j) \geqslant t$, where t is the number from Ogden's lemma,
  2. $m_1 \geqslant (3M+1)(M+t)$, where $M = \max(m_2, m_4, n_3)$,
  3. $m_4 \geqslant (n_1+1)(n_1+t)$.

$$\underbrace{a \dots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \dots a}_{t} \quad \underset{M}{\longleftrightarrow} \quad \dots \quad \underset{M}{\longleftrightarrow} \quad \underbrace{a \dots a}_{t}$$

- If two groups belong to the same pump, then there are at least $(M+2)$ a-s in the pump. It cannot be a $[1,2], [1,4]$ or $[1,7]$-pump (m_2, n_3, m_4 are too small), then it is a $[1,3,6,8]$-pump.
- If all the groups belong to different pumps, then there are at least $M+2$ non-intersecting $[1,2]$ pumps, but m_2 is too small. Again, there exists a $[1,3,6,8]$-pump.
- By analogous arguments, there is a $[1,4]$-pump.
- It should be a $[1,4,8]$-pump, which is impossible. Then $\mathrm{4MIX}$ is not a 2-DCFL.

MIX language

Theorem

MIX *is not a* 1-*DCFL.*

MIX language

Theorem

MIX *is not a* 1*-DCFL.*

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.

MIX language

Theorem

MIX *is not a* 1*-DCFL.*

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.
- 1. $\min(m_j, n_j) \geqslant t$,
  2. $m_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  3. $n_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  4. $m_3 \geqslant (2n_2+1)(n_2+t)$.

MIX language

Theorem

MIX *is not a 1-DCFL.*

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.
- 1. $\min(m_j, n_j) \geqslant t$,
  2. $m_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  3. $n_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  4. $m_3 \geqslant (2n_2+1)(n_2+t)$.
- There are at least $2M+1$ [1]-groups and [4]-groups on the distance at least M.

MIX language

Theorem

MIX *is not a* 1-*DCFL*.

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.
- 1. $\min(m_j, n_j) \geqslant t$,
  2. $m_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  3. $n_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  4. $m_3 \geqslant (2n_2+1)(n_2+t)$.
- There are at least $2M+1$ [1]-groups and [4]-groups on the distance at least M.
- Consequently, there is a $[1, 2, 5, 6]$-group.

MIX language

Theorem

MIX *is not a 1-DCFL.*

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.
- 1. $\min(m_j, n_j) \geqslant t$,
  2. $m_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  3. $n_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  4. $m_3 \geqslant (2n_2+1)(n_2+t)$.
- There are at least $2M+1$ [1]-groups and [4]-groups on the distance at least M.
- Consequently, there is a $[1,2,5,6]$-group.
- There is a $[2,3]$-group, which should be a $[1,2,3,6]$-group

MIX language

Theorem

MIX *is not a* 1*-DCFL.*

- We prove for $\mathrm{MIX} \cap a^+b^+c^+b^+c^+a^+$, $w = a^{m_1}b^{m_2}c^{m_3}b^{n_2}c^{n_3}a^{n_1}$.
-
  1. $\min(m_j, n_j) \geqslant t$,
  2. $m_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  3. $n_1 \geqslant (4M+1)(M+t)$, where $M = \max(m_3, n_2)$,
  4. $m_3 \geqslant (2n_2+1)(n_2+t)$.
- There are at least $2M+1$ [1]-groups and [4]-groups on the distance at least M.
- Consequently, there is a $[1,2,5,6]$-group.
- There is a $[2,3]$-group, which should be a $[1,2,3,6]$-group
- There is a [4]-group, but it could be only a $[1,2,4,6]$-group, which has no c-s. Contradiction.

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ *is not a 1-DCFL.*

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ *is not a 1-DCFL.*

- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ is not a 1-DCFL.

- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.
- Every pump satisfies the prefix condition.

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ is not a 1-DCFL.

- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.
- Every pump satisfies the prefix condition.
- Therefore [4]-pump is automatically a [1, 2, 4]-pump and [3]-pump a [3, 5, 6]-pump.

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ *is not a 1-DCFL.*

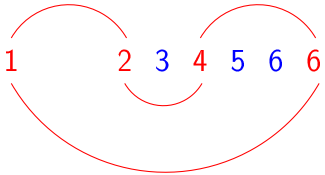
- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.
- Every pump satisfies the prefix condition.
- Therefore [4]-pump is automatically a [1, 2, 4]-pump and [3]-pump a [3, 5, 6]-pump.
- If the first is embracing the second, it is a [1, 2, 4, 6]-pump — impossible to combine them.

MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ is not a 1-DCFL.

- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.
- Every pump satisfies the prefix condition.
- Therefore [4]-pump is automatically a [1, 2, 4]-pump and [3]-pump a [3, 5, 6]-pump.
- If the first is embracing the second, it is a [1, 2, 4, 6]-pump — impossible to combine them.



MIX_p language

Theorem

$\mathrm{MIX}_p = \{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c, \forall u \subseteq w |u|_a \geqslant |u|_b \geqslant |u|_c\}$ *is not a 1-DCFL.*

- Consider the word $a^{m_1} b^{m_2} a^{n_1} c^{m_3} b^{n_2} c^{n_3}$ with $\min(m_j, n_j) \geqslant t$.
- Every pump satisfies the prefix condition.
- Therefore [4]-pump is automatically a [1, 2, 4]-pump and [3]-pump a [3, 5, 6]-pump.
- If the first is embracing the second, it is a [1, 2, 4, 6]-pump — impossible to combine them.
- If the second — symmetrically. Contradiction.

Future work

- Ogden's lemma for k-DCFGs (or any other technique to localize the pumps).

Future work

- Ogden's lemma for k-DCFGs (or any other technique to localize the pumps).
- Apply this lemma to give counterexamples on higher levels of DCFG hierarchy.

Future work

- Ogden's lemma for k-DCFGs (or any other technique to localize the pumps).
- Apply this lemma to give counterexamples on higher levels of DCFG hierarchy.
- Kanazawa-Salvati conjecture (MIX is not a DCFL).

Thank you!

Спасибо за внимание!