On *n*-Path-Controlled Grammars

Jiří Koutný

Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 00 Brno, CZ
www.fit.vutbr.cz/∼ikoutny

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Outline

- **Introduction**
- **Definitions**
- **Results**
- **Examples**
- **Conclusion**
- **References**

Acknowledgement

This work was partially supported by the FRVŠ MŠMT grant FR2581/2010/G1, the BUT FIT grant FIT-10-S-2, and the research plan MSM0021630528.

What's going on

- Regulated formal model.
- Model based on the restrictions on the derivation trees.
- Actual trend in today's FLT (see (1), (2), (3), (4), (5), (6)).
- Simple extension of context-free grammars.
- One of the ways to increase the generative power of context-free grammar.
- Potentially applicable model.

Introduction

What's going on

- Regulated formal model.
- Model based on the restrictions on the derivation trees.
- Actual trend in today's FLT (see (1), (2), (3), (4), (5), (6)).
- Simple extension of context-free grammars.
- One of the ways to increase the generative power of context-free grammar.
- Potentially applicable model.

Motivation

Generation of *not context-free* languages of the form

- $a^n b^n c^n, a^n b^n c^n d^n, a^n b^n c^n d^n e^n, \ldots$
- $a^k b^l a^k b^l, a^k b^l c^m a^k b^l c^m, a^k b^l c^m d^n a^k b^l c^m d^n, \ldots$

Linear grammar

$G = (V, T, P, S)$, where

- V is an alphabet,
- $T \subseteq V$ is a terminal alphabet,
- P is a finite set of production rules of the form $A \rightarrow x$, where $A \in V - T$, $x \in T^*NT^*$, $N = V - T$,
- $S \in V - T$ is the starting symbol.

Linear grammar

$G = (V, T, P, S)$, where

- V is an alphabet,
- $T \subseteq V$ is a terminal alphabet,
- P is a finite set of production rules of the form $A \rightarrow x$, where $A \in V - T$, $x \in T^* N T^*$, $N = V - T$,
- $S \in V - T$ is the starting symbol.

Context-free grammar

$G = (V, T, P, S)$, where

- V is an alphabet,
- $T \subseteq V$ is a terminal alphabet,
- P is a finite set of production rules of the form $A \rightarrow x$, where $A \in V - T$, $x \in V^*$,
- $S \in V - T$ is the starting symbol.

Set of the derivation trees

- Let $G = (V, T, P, S)$ be a grammar.
- Let ${}_G\triangle(x)$ denote a set of the derivation trees with frontier x with respect to the grammar G starting from S.

Set of the derivation trees

- Let $G = (V, T, P, S)$ be a grammar.
- Let ${}_G\triangle(x)$ denote a set of the derivation trees with frontier x with respect to the grammar G starting from S.

A path

- A path s of $t \in_G \triangle(x)$ is sequence $a_1 \dots a_n$, $n \geq 1$, of nodes of t with:
  - a_1 is the root of t,
  - a_1 is labeled by starting symbol of G,
  - a_n is a leaf of t,
  - a_n is labeled by terminal symbol of G,
  - for each $i = 1, \dots, n-1$, there is an edge from a_i to a_{i+1} in t.
- Let $path(s)$ denote the word obtained by concatenating all symbols of the path s (in order from the top).

Path-controlled grammars

- *PC* grammars, for short.

Path-controlled grammars

- *PC* grammars, for short.
- Based on a new type of the restriction in a derivation (see Introduction in (4)).

Path-controlled grammars

- *PC* grammars, for short.
- Based on a new type of the restriction in a derivation (see Introduction in (4)).

Informal idea of *PC* grammars

A derivation tree in a context-free grammar is accepted only if it contains a path described by a string generated by another context-free grammar.

Path-controlled grammars

- *PC* grammars, for short.
- Based on a new type of the restriction in a derivation (see Introduction in (4)).

Informal idea of *PC* grammars

A derivation tree in a context-free grammar is accepted only if it contains a path described by a string generated by another context-free grammar.

- Two grammars G and G':
  - G generates a language over its alphabet of terminals T.
  - G' generates a language over the total alphabet of G.

Path-controlled grammars

- *PC* grammars, for short.
- Based on a new type of the restriction in a derivation (see Introduction in (4)).

Informal idea of *PC* grammars

A derivation tree in a context-free grammar is accepted only if it contains a path described by a string generated by another context-free grammar.

- Two grammars G and G':
  - G generates a language over its alphabet of terminals T.
  - G' generates a language over the total alphabet of G.

More formal idea of *PC* grammars

A string w generated by G is accepted only if there is a derivation tree t of w with respect to G such that there exists a path in t which is described by a string from $L(G')$.

n-path-controlled grammars

- ${}_nPC$ grammars, for short.

n-path-controlled grammars

- ${}_nPC$ grammars, for short.
- A generalization of PC grammars.

- ${}_nPC$ grammars, for short.
- A generalization of *PC* grammars.

Idea of *n*-path-controlled grammars

The string w generated by G is accepted only if there is a derivation tree t of w with respect to G such that there exists $n \geq 0$ paths in t that are described by the strings from linear language $L(G')$.

- ${}_{n}PC$ grammars, for short.
- A generalization of PC grammars.

Idea of n-path-controlled grammars

The string w generated by G is accepted only if there is a derivation tree t of w with respect to G such that there exists $n \geq 0$ paths in t that are described by the strings from linear language $L(G')$.

Several types of ${}_{n}PC$ grammars in relation to

- Path-controlled grammars,
- The pumping lemma for linear languages.

Definition of ${}_nPC$ grammar

An ${}_nPC$ grammar is a pair (G, G'), where

- $G = (V, T, P, S)$ is a context-free grammar,
- $G' = (V', V, P', S')$ is a linear grammar.

Definition of $_nPC$ grammar

An $_nPC$ grammar is a pair (G, G'), where

- $G = (V, T, P, S)$ is a context-free grammar,
- $G' = (V', V, P', S')$ is a linear grammar.

Why G' is a linear grammar?

Definition of $_nPC$ grammar

An $_nPC$ grammar is a pair (G, G'), where

- $G = (V, T, P, S)$ is a context-free grammar,
- $G' = (V', V, P', S')$ is a linear grammar.

Why G' is a linear grammar?

- Regular paths do not increase the generative power (see (3) and (4), Prop. 2).
- Linear paths can increase the generative power (see (4)).

Definition of ${}_nPC$ grammar

An ${}_nPC$ grammar is a pair (G, G'), where

- $G = (V, T, P, S)$ is a context-free grammar,
- $G' = (V', V, P', S')$ is a linear grammar.

Why G' is a linear grammar?

- Regular paths do not increase the generative power (see (3) and (4), Prop. 2).
- Linear paths can increase the generative power (see (4)).

Generated language

$L(G, G') = \{w \in L(G) |$ there is a set C of n different paths in $t \in_G \triangle(w)$ such that for all $p \in C$ it holds $path(p) \in L(G')$ and all $p \in C$ are divided in the common node of $t\}$.

Obvious facts about the paths

Clearly

- Each two paths of a derivation tree contain at least one common node.

Clearly

- Each two paths of a derivation tree contain at least one common node.
- For a ${}_nPC$ grammar (G, G'), there is some $m_C \in \mathbb{N}$ that denotes a number of common nodes for all $p \in C$.

Obvious facts about the paths

Clearly

- Each two paths of a derivation tree contain at least one common node.
- For a ${}_nPC$ grammar (G, G'), there is some $m_C \in \mathbb{N}$ that denotes a number of common nodes for all $p \in C$.
- For each two $p_1, p_2 \in C$ it holds that $path(p_1) = rDs_1$, $path(p_2) = rDs_2$, where $r \in N^*$, $D \in N$, $s_1, s_2 \in N^*T$ and $|rD| = m_C$.

Obvious facts about the paths

Clearly

- Each two paths of a derivation tree contain at least one common node.
- For a ${}_nPC$ grammar (G, G'), there is some $m_C \in \mathbb{N}$ that denotes a number of common nodes for all $p \in C$.
- For each two $p_1, p_2 \in C$ it holds that $path(p_1) = rDs_1$, $path(p_2) = rDs_2$, where $r \in N^*$, $D \in N$, $s_1, s_2 \in N^*T$ and $|rD| = m_C$.
- All the paths $s \in C$ are described by the strings of $L(G')$ which is linear.

Obvious facts about the paths

Clearly

- Each two paths of a derivation tree contain at least one common node.
- For a ${}_nPC$ grammar (G, G'), there is some $m_C \in \mathbb{N}$ that denotes a number of common nodes for all $p \in C$.
- For each two $p_1, p_2 \in C$ it holds that $path(p_1) = rDs_1$, $path(p_2) = rDs_2$, where $r \in N^*$, $D \in N$, $s_1, s_2 \in N^*T$ and $|rD| = m_C$.
- All the paths $s \in C$ are described by the strings of $L(G')$ which is linear.

Pumping lemma for linear languages

If L is a linear language, then there are $p, q \in \mathbb{N}$ such that each string $z \in L$ with $|z| \geq p$ can be written in the form $z = uvwxy$ with $0 < |vx| \leq |uvxy| \leq q$, such that $uv^iwx^iy \in L$ for all $i \geq 1$.

Types of n-path-controlled grammars

- Five types of ${}_nPC$ grammars depending on the value of m_C in relation to the pumping lemma for $L(G')$.

- Five types of ${}_nPC$ grammars depending on the value of m_C in relation to the pumping lemma for $L(G')$.

Types of ${}_nPC$ grammars

- ${}_n^{I}PC$ if C satisfies $0 \leq m_C \leq |u|$,
- ${}_n^{II}PC$ if C satisfies $|u| < m_C \leq |uv|$,
- ${}_n^{III}PC$ if C satisfies $|uv| < m_C \leq |uvw|$,
- ${}_n^{IV}PC$ if C satisfies $|uvw| < m_C \leq |uvwx|$,
- ${}_n^{V}PC$ if C satisfies $|uvwx| < m_C \leq |uvwxy|$,

where $uvwxy$ is the shortest path from C.

- Five types of ${}_nPC$ grammars depending on the value of m_C in relation to the pumping lemma for $L(G')$.

Types of ${}_nPC$ grammars

- ${}^{I}_{n}PC$ if C satisfies $0 \leq m_C \leq |u|$,
- ${}^{II}_{n}PC$ if C satisfies $|u| < m_C \leq |uv|$,
- ${}^{III}_{n}PC$ if C satisfies $|uv| < m_C \leq |uvw|$,
- ${}^{IV}_{n}PC$ if C satisfies $|uvw| < m_C \leq |uvwx|$,
- ${}^{V}_{n}PC$ if C satisfies $|uvwx| < m_C \leq |uvwxy|$,

where $uvwxy$ is the shortest path from C.

Language families

The family of the languages generated by LIN, CF, PC, ${}_nPC$, ${}^{I}_{n}PC$, ${}^{II}_{n}PC$, ${}^{III}_{n}PC$, ${}^{IV}_{n}PC$, ${}^{V}_{n}PC$ grammars is denoted by **LIN**, **CF**, **PC**, ***n*-PC**, **I-*n*-PC**, **II-*n*-PC**, **III-*n*-PC**, **IV-*n*-PC**, **V-*n*-PC**, respectively.

Theorem 1

$\mathbf{PC} = \mathbf{1\text{-}PC} = \mathbf{I\text{-}1\text{-}PC} = \mathbf{II\text{-}1\text{-}PC} = \mathbf{III\text{-}1\text{-}PC} = \mathbf{IV\text{-}1\text{-}PC} = \mathbf{V\text{-}1\text{-}PC}$.

Theorem 1

PC $=$ **1-PC** $=$ **I-1-PC** $=$ **II-1-PC** $=$ **III-1-PC** $=$ **IV-1-PC** $=$ **V-1-PC**.

Proof: The equality clearly follows from the definitions of PC, ${}_nPC$, and ${}_n^iPC$, for $i = I, II, III, IV, V$, grammars.
Informally: One path to control means no division of the controlled paths.

Theorem 1

PC $=$ **1-PC** $=$ **I-1-PC** $=$ **II-1-PC** $=$ **III-1-PC** $=$ **IV-1-PC** $=$ **V-1-PC**.

Proof: The equality clearly follows from the definitions of PC, ${}_nPC$, and ${}^i_nPC$, for $i = I, II, III, IV, V$, grammars.
Informally: One path to control means no division of the controlled paths.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Theorem 1

PC $=$ **1-PC** $=$ **I-1-PC** $=$ **II-1-PC** $=$ **III-1-PC** $=$ **IV-1-PC** $=$ **V-1-PC**.

Proof: The equality clearly follows from the definitions of PC, ${}_nPC$, and ${}^i_nPC$, for $i = I, II, III, IV, V$, grammars.
Informally: One path to control means no division of the controlled paths.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Notice that for $n = 0$, the Theorem 2 holds for context-free languages.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Let (G, G') be a ${}^{III}_{n}PC$-grammar, where
  - $G = (V, T, P, S)$,
  - $G' = (V', V, P', S')$.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Let (G, G') be a ${}^{III}_{n}PC$-grammar, where
  - $G = (V, T, P, S)$,
  - $G' = (V', V, P', S')$.
- Consider $t \in_{(G,G')} \triangle(z)$. For each $path(s) = SA_1 \dots A_k a$ of t, where $s \in C$, consider
  - the rules $A_i \rightarrow x_i A_{i+1} y_i$ used when passing from A_i to A_{i+1} on this path,
  - the rule $A_k \rightarrow x_k a y_k$ used in the last step of the derivation in G corresponding to the path s.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Consider that any $x_i y_i$, $i = 1, \ldots, k$, contains a nonterminal B that do not belong on any path $s \in C$. Clearly, there is substring z' of z derived from B.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Consider that any $x_i y_i$, $i = 1, \dots, k$, contains a nonterminal B that do not belong on any path $s \in C$. Clearly, there is substring z' of z derived from B.
- Since G is context-free, it follows that if $|z'| \geq k_1$, for some $k_1 \geq 0$, then there are two substrings z'_1, z'_2 of z' that can be pumped.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Consider that any $x_i y_i$, $i = 1, \dots, k$, contains a nonterminal B that do not belong on any path $s \in C$. Clearly, there is substring z' of z derived from B.
- Since G is context-free, it follows that if $|z'| \geq k_1$, for some $k_1 \geq 0$, then there are two substrings z_1', z_2' of z' that can be pumped.
- By the pumping lemma for context-free languages, z_1', z_2' are bounded in length.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- If $L(G)$ is infinite, the string $path(s) \in L(G')$ is potentially arbitrarily long. Thus, if $path(s) = u_s v_s x_s y_s z_s$ with $|u_s v_s x_s y_s z_s| \geq k_2$, for some $k_2 \geq 0$, then $u_s v_s x_s y_s z_s$ satisfies $u_s v_s^i x_s y_s^i z_s \in L(G')$, for $i \geq 1$.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- If $L(G)$ is infinite, the string $path(s) \in L(G')$ is potentially arbitrarily long. Thus, if $path(s) = u_s v_s x_s y_s z_s$ with $|u_s v_s x_s y_s z_s| \geq k_2$, for some $k_2 \geq 0$, then $u_s v_s x_s y_s z_s$ satisfies $u_s v_s^i x_s y_s^i z_s \in L(G')$, for $i \geq 1$.
- The derivations starting from the symbols of v and y can be repeated in G.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- If $L(G)$ is infinite, the string $path(s) \in L(G')$ is potentially arbitrarily long. Thus, if $path(s) = u_s v_s x_s y_s z_s$ with $|u_s v_s x_s y_s z_s| \geq k_2$, for some $k_2 \geq 0$, then $u_s v_s x_s y_s z_s$ satisfies $u_s v_s^i x_s y_s^i z_s \in L(G')$, for $i \geq 1$.
- The derivations starting from the symbols of v and y can be repeated in G.
- Since (G, G') is ${}^{III}_{n}PC$ grammar, it follows that:
  - the derivations starting from the symbols of v in G are common for all $s \in C$,
  - the derivations starting from the symbols of y in G are potentially unique for each $s \in C$.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Consider the derivations starting from v in G. This leads to the pumping of two substrings v_1, v_{2n+2} of z—one in the left-hand side, one in the right-hand side controlled by the common part of all $s \in C$.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- Consider the derivations starting from v in G. This leads to the pumping of two substrings v_1, v_{2n+2} of z—one in the left-hand side, one in the right-hand side controlled by the common part of all $s \in C$.
- Consider the derivations starting from y in G. This leads to the pumping of two substrings of z—one in the left-hand side, one in the right-hand side corresponding to each $s \in C$. For each $s_{i+1} \in C$, denote this two substrings v_{2i+2}, v_{2i+3}, $i = 0, 1, \dots, n-1$. Since (G, G') is ${}^{III}_{n}PC$ grammar, we obtain $2n$ pumped substrings of z.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \dots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \dots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \dots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- By the pumping lemma for context-free languages, the substrings $v_1, v_2, \dots, v_{2n+2}$ are bounded in length.

Theorem 2

If $L \in$ **III-n-PC**, for $n = card(C) \geq 0$, then there are $p, q \in \mathbb{N}$ such that each $z \in L$ with $|z| > p$ can be written in the form $z = u_1 v_1 u_2 v_2 \ldots u_{2n+2} v_{2n+2} u_{2n+3}$, such that $0 < |v_1 v_2 \ldots v_{2n+2}| \leq q$ and $u_1 v_1^i u_2 v_2^i \ldots u_{2n+2} v_{2n+2}^i u_{2n+3} \in L$ for all $i \geq 1$.

Proof Idea:

- By the pumping lemma for context-free languages, the substrings $v_1, v_2, \ldots, v_{2n+2}$ are bounded in length.
- Thus, the total length of the $2n + 2$ pumped substrings of z is bounded by a constant q.

Corollary 3

III-n-PC cannot count to $2n+3$, but can count to $2n+2$.

Proof: $L = \{a^i b^i c^i d^i e^i f^i g^i \mid i \geq 1\} \notin$ **III-2-PC**, but $L \in$ **III-3-PC**.

Corollary 3

III-n-PC cannot count to $2n+3$, but can count to $2n+2$.

Proof: $L = \{a^i b^i c^i d^i e^i f^i g^i \mid i \geq 1\} \notin$ **III-2-PC**, but $L \in$ **III-3-PC**.

Corollary 4

There is an infinite hierarchy of $\bigcup_{i=0}^{n}$ **III-i-PC** languages.

Proof: $\bigcup_{i=0}^{n}$ **III-i-PC** $\subset \bigcup_{i=0}^{n+1}$ **III-i-PC**, for $n \geq 0$, is proper.

Corollary 3

III-n-PC cannot count to $2n+3$, but can count to $2n+2$.

Proof: $L = \{a^i b^i c^i d^i e^i f^i g^i \mid i \geq 1\} \notin$ **III-2-PC**, but $L \in$ **III-3-PC**.

Corollary 4

There is an infinite hierarchy of $\bigcup_{i=0}^{n}$ **III-i-PC** languages.

Proof: $\bigcup_{i=0}^{n}$ **III-i-PC** $\subset \bigcup_{i=0}^{n+1}$ **III-i-PC**, for $n \geq 0$, is proper.

Corollary 5

III-n-PC is not closed under concatenation.

Proof: $L = \{a^i a^i a^i a^i a^i a^i \mid i \geq 1\} \in$ **III-2-PC**, but $LL \notin$ **III-2-PC**.

Example 1

Let us have ${}^{lll}_{n}PC$ grammar (G, G'), $n \geq 0$, where

$$
\begin{aligned}
G_1 = (&\{S\} \cup \{A_i, B_i |\ i = 1, \dots, n\} \cup \{a_i |\ i = 1, \dots, 2n+2\},\\
&\{a_i |\ i = 1, \dots, 2n+2\}, P, S)\\
P =&\{S \rightarrow a_1 S a_{2n+2}, \quad S \rightarrow a_1 A_1 A_2 \dots A_n a_{2n+2}\} \cup\\
&\{A_{i+1} \rightarrow a_{2i+2} A_{i+1} a_{2i+3}, \quad A_{i+1} \rightarrow B_{i+1},\\
&\ B_{i+1} \rightarrow a_{2i+2} a_{2i+3} |\ i = 0, \dots, n-1\}\\
L(G') =& \bigcup_{i=1}^{n} \{S^k A_i^k B_i a_{2i} |\ k \geq 1\}
\end{aligned}
$$

Example 1

Let us have ${}^{III}_{n}PC$ grammar (G, G'), $n \geq 0$, where

$$\begin{aligned}
G_1 = (&\{S\} \cup \{A_i, B_i \mid i = 1, \ldots, n\} \cup \{a_i \mid i = 1, \ldots, 2n+2\}, \\
&\{a_i \mid i = 1, \ldots, 2n+2\}, P, S) \\
P = &\{S \to a_1 S a_{2n+2}, \quad S \to a_1 A_1 A_2 \ldots A_n a_{2n+2}\} \cup \\
&\{A_{i+1} \to a_{2i+2} A_{i+1} a_{2i+3}, \quad A_{i+1} \to B_{i+1}, \\
&\quad B_{i+1} \to a_{2i+2} a_{2i+3} \mid i = 0, \ldots, n-1\} \\
L(G') = &\bigcup_{i=1}^{n} \{S^k A_i^k B_i a_{2i} \mid k \geq 1\}
\end{aligned}$$

Consider a derivation in (G, G'):

$$\begin{aligned}
S &\Rightarrow^k a_1^k S a_{2n+2}^k \\
&\Rightarrow a_1^k a_1 A_1 \ldots A_n a_{2n+2} a_{2n+2}^k \\
&\Rightarrow^{n \times k} a_1^{k+1} a_2^k B_1 a_3^k \ldots a_{2n}^k B_n a_{2n+1}^k a_{2n+2}^{k+1} \\
&\Rightarrow^n a_1^{k+1} a_2^{k+1} a_3^{k+1} \ldots a_{2n}^{k+1} a_{2n+1}^{k+1} a_{2n+2}^{k+1}
\end{aligned}$$

Example 1

Let us have ${}^{III}_{n}PC$ grammar (G, G'), $n \geq 0$, where

$$
\begin{aligned}
G_1 = (&\{S\} \cup \{A_i, B_i \mid i = 1, \dots, n\} \cup \{a_i \mid i = 1, \dots, 2n+2\}, \\
&\{a_i \mid i = 1, \dots, 2n+2\}, P, S) \\
P = &\{S \rightarrow a_1 S a_{2n+2}, \quad S \rightarrow a_1 A_1 A_2 \dots A_n a_{2n+2}\} \cup \\
&\{A_{i+1} \rightarrow a_{2i+2} A_{i+1} a_{2i+3}, \quad A_{i+1} \rightarrow B_{i+1}, \\
&B_{i+1} \rightarrow a_{2i+2} a_{2i+3} \mid i = 0, \dots, n-1\} \\
L(G') = &\bigcup_{i=1}^{n} \{S^k A_i^k B_i a_{2i} \mid k \geq 1\}
\end{aligned}
$$

Consider a derivation in (G, G'):

$$
\begin{aligned}
S &\Rightarrow^k a_1^k S a_{2n+2}^k \\
&\Rightarrow a_1^k a_1 A_1 \dots A_n a_{2n+2} a_{2n+2}^k \\
&\Rightarrow^{n \times k} a_1^{k+1} a_2^k B_1 a_3^k \dots a_{2n}^k B_n a_{2n+1}^k a_{2n+2}^{k+1} \\
&\Rightarrow^n a_1^{k+1} a_2^{k+1} a_3^{k+1} \dots a_{2n}^{k+1} a_{2n+1}^{k+1} a_{2n+2}^{k+1}
\end{aligned}
$$

$$L(G_1, G') = \{a_1^k, \dots, a_{2n+2}^k \mid k \geq 1\}.$$

Example 2

Consider $^{lll}_{2}PC$ grammar (G, G'), where

$$G = (\{S, X, Y, U, V, a, b, c, d, e, f\}, \{a, b, c, d, e, f\}, P, S)$$
$$\begin{aligned} P = \{ & S \rightarrow aSf, \quad S \rightarrow aXYf, \quad X \rightarrow bXc, \quad Y \rightarrow dYe, \\ & X \rightarrow U, \quad U \rightarrow bc, \quad Y \rightarrow V, \quad V \rightarrow de\} \end{aligned}$$
$$L(G') = \{S^n X^n Ub \cup S^n Y^n Vd \mid n \geq 1\}$$

$$L(G, G') = \{a^i b^i c^i d^i e^i f^i \mid i \geq 1\}$$

Example 2

Consider ${}^{lll}_{2}PC$ grammar (G, G'), where

$$G = (\{S, X, Y, U, V, a, b, c, d, e, f\}, \{a, b, c, d, e, f\}, P, S)$$
$$P = \{S \to aSf, \quad S \to aXYf, \quad X \to bXc, \quad Y \to dYe,$$
$$X \to U, \quad U \to bc, \quad Y \to V, \quad V \to de\}$$
$$L(G') = \{S^n X^n Ub \cup S^n Y^n Vd \mid n \geq 1\}$$

$$L(G, G') = \{a^i b^i c^i d^i e^i f^i \mid i \geq 1\}$$

Example of the derivation:

$S \Rightarrow aSf \Rightarrow aaSff \Rightarrow aaaSfff \Rightarrow aaaaXYffff \Rightarrow aaaabXcYffff \Rightarrow$
$aaaabbXccYffff \Rightarrow aaaabbbXcccYfffff \Rightarrow$
$aaaabbbUcccYffff \Rightarrow aaaabbbbccccYffff \Rightarrow$
$aaaabbbbccccdYeffff \Rightarrow aaaabbbbccccddYeeffff \Rightarrow$
$aaaabbbbccccdddYeeeffff \Rightarrow aaaabbbbccccdddVeeeffff \Rightarrow$
$aaaabbbbccccddddeeeeffff = a^4 b^4 c^4 d^4 e^4 f^4$

Example 3

Let $m \geq 0$ with $m \bmod 2 = 0$. Let us have ${}^{lll}_{n}PC$ grammar (G, G'), $n \geq 0$, where

$$\begin{aligned}
G = (&\{A_j, B_j, a_j \mid j = 1, \dots, m\} \cup \{C\}, \{a_j \mid j = 1, \dots, m\}, P, A_1) \\
P = &\{A_1 \to a_1 A_1, \quad A_1 \to a_1 A_2, \quad B_1 \to B_1 a_1, \quad B_1 \to C, \quad C \to a_1\} \cup \\
&\{A_m \to A_m a_m, \quad A_m \to \{B_m\}^n\} \cup \\
&\{A_i \to A_i a_i, \quad A_i \to A_{i+1} \mid i = 2, \dots, m-1 \text{ with } i \bmod 2 = 0\} \cup \\
&\{A_i \to a_i A_i, \quad A_i \to A_{i+1} \mid i = 3, \dots, m-1 \text{ with } i \bmod 2 = 1\} \cup \\
&\{B_i \to a_i B_i, \quad B_i \to B_{i-1} \mid i = 2, \dots, m \text{ with } i \bmod 2 = 0\} \cup \\
&\{B_i \to B_i a_i, \quad B_i \to B_{i-1} \mid i = 3, \dots, m \text{ with } i \bmod 2 = 1\}
\end{aligned}$$

$$L(G') = \{A_1^{k_1} A_2^{k_2} \dots A_m^{k_m} B_m^{k_m} B_{m-1}^{k_{m-1}} \dots B_2^{k_2} B_1^{k_1} C a_1 \mid k_i \geq 0, i = 1, \dots, m\}$$

Examples

Consider a derivation in (G, G')*:*

$$
\begin{aligned}
A_1 &\Rightarrow^{k_1} a_1^{k_1} A_1 \Rightarrow a_1^{k_1+1} A_2 \Rightarrow^{k_2} a_1^{k_1+1} A_2 a_2^{k_2} \Rightarrow a_1^{k_1+1} A_3 a_2^{k_2} \\
&\Rightarrow^{*} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} A_m a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{B_m\}^n a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n \times k_m} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_m\}^n a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_{m-1}\}^n a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n \times k_{m-1}} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_{m-1} a_{m-1}^{k_{m-1}}\}^n a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{*} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \dots a_2^{k_2} B_1 a_1^{k_1} \dots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\quad a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \dots a_2^{k_2} C a_1^{k_1} \dots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\quad a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \dots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \dots a_2^{k_2} a_1^{k+1} \dots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\quad a_m^{k_m} \dots a_6^{k_6} a_4^{k_4} a_2^{k_2}
\end{aligned}
$$

Consider a derivation in (G, G'):

$$
\begin{aligned}
A_1 &\Rightarrow^{k_1} a_1^{k_1} A_1 \Rightarrow a_1^{k_1+1} A_2 \Rightarrow^{k_2} a_1^{k_1+1} A_2 a_2^{k_2} \Rightarrow a_1^{k_1+1} A_3 a_2^{k_2} \\
&\Rightarrow^* a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} A_m a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{B_m\}^n a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n \times k_m} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_m\}^n a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_{m-1}\}^n a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n \times k_{m-1}} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} B_{m-1} a_{m-1}^{k_{m-1}}\}^n a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^* a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \ldots a_2^{k_2} B_1 a_1^{k_1} \ldots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\qquad a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \ldots a_2^{k_2} C a_1^{k_1} \ldots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\qquad a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2} \\
&\Rightarrow^{n} a_1^{k_1+1} a_3^{k_3} a_5^{k_5} \ldots a_{m-1}^{k_{m-1}} \{a_m^{k_m} a_{m-2}^{k_{m-2}} \ldots a_2^{k_2} a_1^{k+1} \ldots a_{m-3}^{k_{m-3}} a_{m-1}^{k_{m-1}}\}^n \\
&\qquad a_m^{k_m} \ldots a_6^{k_6} a_4^{k_4} a_2^{k_2}
\end{aligned}
$$

$$
\begin{aligned}
L(G, G') = \{(&a_1^{k_1+1} a_3^{k_3} \ldots a_{m-1}^{k_{m-1}} a_m^{m} a_{m-2}^{k_{m-2}} a_{m-4}^{k_{m-4}} \ldots a_2^{k_2})^{n+1} \\
&\mid k_i \geq 0, i = 1, \ldots, m\}
\end{aligned}
$$

Examples

Example 4

Consider $m = 4$ and ${}_3^{lll}PC$ grammar (G, G'), where

$$G = (\{A, B, C, D, E, F, G, H, I, a, b, c, d\}, \{a, b, c, d\}, P, A)$$

$$\begin{aligned} P = \{ & A \rightarrow aA, & A \rightarrow aB, && B \rightarrow Bb, && B \rightarrow C, \\ & C \rightarrow cC, & C \rightarrow D, && D \rightarrow Dd, && D \rightarrow HHH, \\ & E \rightarrow Ea, & E \rightarrow I, && F \rightarrow bF, && F \rightarrow E, \\ & G \rightarrow Gc, & G \rightarrow F, && H \rightarrow dH, && H \rightarrow G, && I \rightarrow a\} \end{aligned}$$

$$L(G') = \{A^r B^s C^t D^u H^u G^t F^s E^r Ia \mid r, s, t, u \geq 0\}$$

$$L(G, G') = \{a^v c^w d^x b^y a^v c^w d^x b^y a^v c^w d^x b^y a^v c^w d^x b^y \mid v > 0, w, x, y \geq$$

Examples

Example of the derivation:

$$
\begin{aligned}
& A \Rightarrow aA \Rightarrow aaB \Rightarrow aaBb \Rightarrow aaCb \Rightarrow aacCb \Rightarrow aacDb \Rightarrow \\
& aacDdb \Rightarrow aacHHHdb \Rightarrow aacdHHHdb \Rightarrow aacdGHHdb \Rightarrow \\
& aacdGcHHdb \Rightarrow aacdFcHHdb \Rightarrow aacdbFcHHdb \Rightarrow \\
& aacdbEcHHdb \Rightarrow aacdbEacHHdb \Rightarrow aacdbIacHHdb \Rightarrow \\
& aacdbaacHHdb \Rightarrow aacdbaacdHHdb \Rightarrow aacdbaacdGHdb \Rightarrow \\
& aacdbaacdGcHdb \Rightarrow aacdbaacdFcHdb \Rightarrow \\
& aacdbaacdbFcHdb \Rightarrow aacdbaacdbEcHdb \Rightarrow \\
& aacdbaacdbEacHdb \Rightarrow aacdbaacdbIacHdb \Rightarrow \\
& aacdbaacdbaacHdb \Rightarrow aacdbaacdbaacdHdb \Rightarrow \\
& aacdbaacdbaacdGdb \Rightarrow aacdbaacdbaacdGcdb \Rightarrow \\
& aacdbaacdbaacdFcdb \Rightarrow aacdbaacdbaacdbFcdb \Rightarrow \\
& aacdbaacdbaacdbEcdb \Rightarrow aacdbaacdbaacdbEacdb \Rightarrow \\
& aacdbaacdbaacdbIacdb \Rightarrow aacdbaacdbaacdbaacdb
\end{aligned}
$$

Future research ideas

Investigation of **III-n-PC**

${}_{n}^{III}PC$ grammars are potentially usable.

- Generative power?
- Closure properties?
- Decidability properties?
- Parsing properties?
- Descriptional complexity?

Future research ideas

Investigation of **III-n-PC**

$^{III}_{n}PC$ grammars are potentially usable.

- Generative power?
- Closure properties?
- Decidability properties?
- Parsing properties?
- Descriptional complexity?

Investigation of **I-n-PC** and **V-n-PC**

- $^{n}_{I}PC$ grammars are equal to concatenation of n independent PC grammars?
- $^{n}_{V}PC$ grammars are equal to CF grammars?

Future research ideas

Investigation of **III-n-PC**

$_{n}^{III}PC$ grammars are potentially usable.

- Generative power?
- Closure properties?
- Decidability properties?
- Parsing properties?
- Descriptional complexity?

Investigation of **I-n-PC** and **V-n-PC**

- $_{I}^{n}PC$ grammars are equal to concatenation of n independent PC grammars?
- $_{V}^{n}PC$ grammars are equal to CF grammars?

Investigation of **II-n-PC** and **IV-n-PC**

$_{II}^{n}PC$ grammars and $_{IV}^{n}PC$ grammars are unusable?

References

K. Čulik and H. A. Maurer.
Tree controlled grammars.
Computing, 19:129–139, 1977.

J. Dassow and B. Truthe.
Subregularly tree controlled grammars and languages.
In *Automata and Fromal Languages - 12th International Conference AFL 2008, Balatonfured*, pages 158–169. Hungarian Academy of Sciences, 2008.

J. Koutný.
Regular paths in derivation trees of context-free grammars.
In *Proceedings of the 15th Conference and Competition STUDENT EEICT 2009 Volume 4*, pages 410–414. Faculty of Information Technology BUT, 2009.

S. Marcus, C. Martín-Vide, V. Mitrana, and Gh. Păun.
A new-old class of linguistically motivated regulated grammars.
In *CLIN*, pages 111–125, 2000.

C. Martín-Vide and V. Mitrana.
Further properties of path-controlled grammars.
In *Formal Grammar / Mathematics of Language 2005*, pages 219–230. Edimburgh, 2005.

Gh. Păun.
On the generative capacity of tree controlled grammars.
Computing, 21(3):213–220, 1979.

Thank you for your attention!