Parallel Communicating Grammar Systems

Jiří Techet Tomáš Masopust Alexander Meduna

Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, Brno 61266, Czech Republic

Modern Formal Language Theory, 2007

PC Grammar Systems

PC Grammar System

A parallel communicating (PC) grammar system of degree n, $n \geq 1$, is a construct

$$\Gamma = (N, K, T, (S_1, P_1), \ldots, (S_n, P_n))$$

where

K is a finite set of query symbols, $K = \{Q_1, \ldots, Q_n\}$

P_i is a finite set of productions of the form

$$A \rightarrow x$$

with $A \in N$ and $x \in (N \cup T \cup K)^*$, for all $i = 1, \ldots, n$

S_i is the start symbol of the ith component, $S_i \in N$ for all $i = 1, \ldots, n$

N, T are defined as usual, N, K, T are pairwise disjoint

PC Grammar Systems – Derivation I

Two Kinds of Derivation Steps

- generating
- communicating

g-Step

If

- either $x_i \Rightarrow y_i$ in $G_i = (N \cup K, T, P_i, S_i)$,
- or $x_i = y_i \in T^*$

for all $1 \leq i \leq n$, then

$$(x_1, \ldots, x_n) \; {}_g\!\Rightarrow (y_1, \ldots, y_n)$$

PC Grammar Systems – Derivation II

c-Step

- set $z_i = x_i$ for all $i = 1, \ldots, n$

For each $i = 1, \ldots, n$, if

$$\mathrm{alph}(x_i) \cap K \neq \emptyset$$

and for each Q_j in x_i,

$$\mathrm{alph}(x_j) \cap K = \emptyset,$$

then for each Q_j in x_i

1. set $z_j = S_j$,
2. replace Q_j with x_j in x_i,
3. set z_i to the string resulting from (2)

Perform

$$(x_1, \ldots, x_n) \;{}_c\!\Rightarrow (y_1, \ldots, y_n)$$

with $y_i = z_i$, for all $i = 1, \ldots, n$

PC Grammar Systems – Generated Language

Direct Derivation

If either

$$(x_1, \ldots, x_n) \;{}_g\!\Rightarrow (y_1, \ldots, y_n)$$

or

$$(x_1, \ldots, x_n) \;{}_c\!\Rightarrow (y_1, \ldots, y_n)$$

then

$$(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$$

Generated Language

$$L(\Gamma) = \{x \in T^* \,:\, (S_1, S_2, \ldots, S_n) \Rightarrow^* (x, \alpha_2, \ldots, \alpha_n), \\ \alpha_i \in (N \cup T \cup K)^*, \text{ for all } i = 2, \ldots, n\}$$

Centralized PC Grammar Systems

- only P_1 can produce query symbols

Centralized PC Grammar System

Let

$$\Gamma = (N, K, T, (S_1, P_1), \ldots, (S_n, P_n))$$

be a PC grammar system. Γ is centralized if for all $A \to x \in P_i$, where $i = 2, \ldots, n$,

$$\mathrm{alph}(x) \cap K = \emptyset$$

Returning and Non-Returning PC Grammar Systems

Returning PC Grammar System

After communicating, each component that has sent its string to another component returns to its axiom.

- generated language denoted by $L_r(\Gamma)$

Non-Returning PC Grammar System

After communicating, each component that has sent its string to another component continues to process the current string. That is, remove (1) in the basic definition.

- generated language denoted by $L_{nr}(\Gamma)$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned} P_1 = \{ & S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2, \\ & S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\ P_2 = \{ & S_2 \rightarrow bS_2\} \\ P_3 = \{ & S_3 \rightarrow cS_3\} \end{aligned}$$

(S_1, S_2, S_3)

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{ & S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2, \\
& S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\
P_2 = \{ & S_2 \rightarrow bS_2\} \\
P_3 = \{ & S_3 \rightarrow cS_3\}
\end{aligned}$$

$$(S_1, S_2, S_3) \Rightarrow (aS_1', bS_2, cS_3)$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{&S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2, \\
&S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\
P_2 = \{&S_2 \rightarrow bS_2\} \\
P_3 = \{&S_3 \rightarrow cS_3\}
\end{aligned}$$

$$(S_1, S_2, S_3) \Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3)$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S'_1, S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned} P_1 = \{ & S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS'_1, S_1 \rightarrow a^3Q_2, \\ & S'_1 \rightarrow aS'_1, S'_1 \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\ P_2 = \{ & S_2 \rightarrow bS_2\} \\ P_3 = \{ & S_3 \rightarrow cS_3\} \end{aligned}$$

$$\begin{aligned} (S_1, S_2, S_3) & \Rightarrow (aS'_1, bS_2, cS_3) \Rightarrow^* (a^nS'_1, b^nS_2, c^nS_3) \\ & \Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \end{aligned}$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{ & S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2, \\
& S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\
P_2 = \{ & S_2 \rightarrow bS_2\} \\
P_3 = \{ & S_3 \rightarrow cS_3\}
\end{aligned}$$

$$\begin{aligned}
(S_1, S_2, S_3) & \Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3) \\
& \Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \Rightarrow (a^{n+3}b^{n+1}S_2, S_2, c^{n+1}S_3)
\end{aligned}$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned} P_1 = \{ & S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2, \\ & S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\} \\ P_2 = \{ & S_2 \rightarrow bS_2\} \\ P_3 = \{ & S_3 \rightarrow cS_3\} \end{aligned}$$

$$\begin{aligned} (S_1, S_2, S_3) & \Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3) \\ & \Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \Rightarrow (a^{n+3}b^{n+1}S_2, S_2, c^{n+1}S_3) \\ & \Rightarrow (a^{n+3}b^{n+3}Q_3, bS_2, c^{n+2}S_3) \end{aligned}$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{&S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2,\\
&S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\}\\
P_2 = \{&S_2 \rightarrow bS_2\}\\
P_3 = \{&S_3 \rightarrow cS_3\}
\end{aligned}$$

$$\begin{aligned}
(S_1, S_2, S_3) &\Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3)\\
&\Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \Rightarrow (a^{n+3}b^{n+1}S_2, S_2, c^{n+1}S_3)\\
&\Rightarrow (a^{n+3}b^{n+3}Q_3, bS_2, c^{n+2}S_3) \Rightarrow (a^{n+3}b^{n+3}c^{n+2}S_3, bS_2, S_3)
\end{aligned}$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{&S_1 \rightarrow abc, S_1 \rightarrow a^2b^2c^2, S_1 \rightarrow aS_1', S_1 \rightarrow a^3Q_2,\\
&S_1' \rightarrow aS_1', S_1' \rightarrow a^3Q_2, S_2 \rightarrow b^2Q_3, S_3 \rightarrow c\}\\
P_2 = \{&S_2 \rightarrow bS_2\}\\
P_3 = \{&S_3 \rightarrow cS_3\}
\end{aligned}$$

$$\begin{aligned}
(S_1, S_2, S_3) &\Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3)\\
&\Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \Rightarrow (a^{n+3}b^{n+1}S_2, S_2, c^{n+1}S_3)\\
&\Rightarrow (a^{n+3}b^{n+3}Q_3, bS_2, c^{n+2}S_3) \Rightarrow (a^{n+3}b^{n+3}c^{n+2}S_3, bS_2, S_3)\\
&\Rightarrow (a^{n+3}b^{n+3}c^{n+3}, bbS_2, cS_3)
\end{aligned}$$

PC Grammar Systems – Example

Example

$$\Gamma = (\{S_1, S_1', S_2, S_3\}, K, \{a, b\}, (S_1, P_1), (S_2, P_2), (S_3, P_3))$$

where

$$\begin{aligned}
P_1 = \{&S_1 \to abc, S_1 \to a^2b^2c^2, S_1 \to aS_1', S_1 \to a^3Q_2,\\
&S_1' \to aS_1', S_1' \to a^3Q_2, S_2 \to b^2Q_3, S_3 \to c\}\\
P_2 = \{&S_2 \to bS_2\}\\
P_3 = \{&S_3 \to cS_3\}
\end{aligned}$$

$$\begin{aligned}
(S_1, S_2, S_3) &\Rightarrow (aS_1', bS_2, cS_3) \Rightarrow^* (a^nS_1', b^nS_2, c^nS_3)\\
&\Rightarrow (a^{n+3}Q_2, b^{n+1}S_2, c^{n+1}S_3) \Rightarrow (a^{n+3}b^{n+1}S_2, S_2, c^{n+1}S_3)\\
&\Rightarrow (a^{n+3}b^{n+3}Q_3, bS_2, c^{n+2}S_3) \Rightarrow (a^{n+3}b^{n+3}c^{n+2}S_3, bS_2, S_3)\\
&\Rightarrow (a^{n+3}b^{n+3}c^{n+3}, bbS_2, cS_3)
\end{aligned}$$

$$L_r(\Gamma) = L_{nr}(\Gamma) = \{a^nb^nc^n \,:\, n \geq 1\}$$

Denotation of PC Language Families

Denotation of PC Language Families

$$XPC_nY$$

X

- N – non-returning mode
- C – centralized PC grammar systems

n number of components (by analogy with CD grammar systems)

Y specification of the type of productions (REG, LIN, CF)

Example

CPC_2REG, NPC_2LIN, $NCPC_\infty$

PC Grammar Systems – Generative Power I

Theorem

- $PC_nREG - \mathscr{L}(LIN) \neq \emptyset$, *for* $n \geq 2$
- $PC_nREG - \mathscr{L}(CF) \neq \emptyset$, *for* $n \geq 3$
- $PC_nLIN - \mathscr{L}(CF) \neq \emptyset$, *for* $n \geq 2$

Theorem

$\mathscr{L}(LIN) - CPC_\infty REG \neq \emptyset$

PC Grammar Systems – Generative Power II

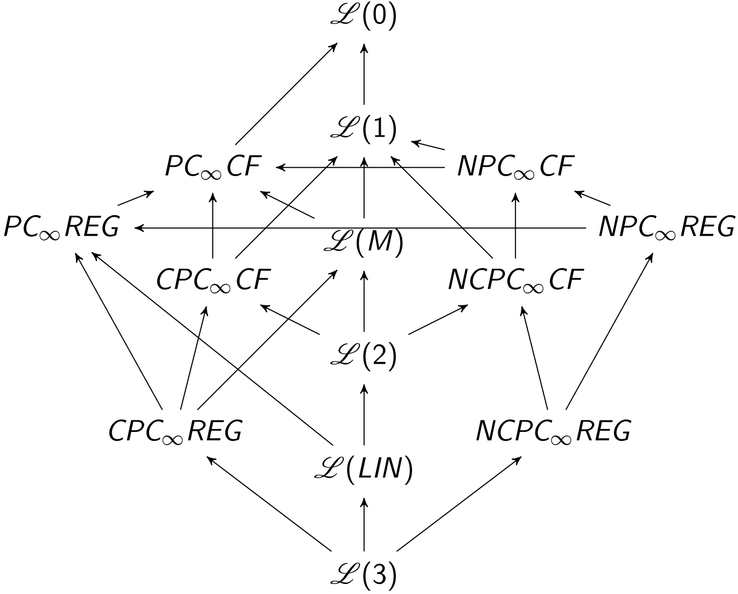
Theorem
$PC_nREG \subset PC_{n+1}REG$, *for* $n \geq 1$

Theorem
$CPC_nREG \subset CPC_nLIN \subset CPC_nCF$, *for* $n \geq 1$

Theorem

- $NPC_\infty CF \subseteq PC_\infty CF$
- $\mathscr{L}(M) \subset PC_\infty CF$
- $\mathscr{L}(LIN) \subset PC_\infty REG$

PC Grammar Systems – The Hierarchy



Bibliography

Grammar systems.
`http://www.sztaki.hu/mms/bib.html`.

Gh. Păun and L. Sântean.
Parallel communicating grammar systems: the regular case.
Annals Universitatis Bucharest, Mathematics-Informatics Series, 38:55–63, 1989.