On the Generative Power of CD Grammar Systems With Scattered Context Components

Jakub Martiško, Alexander Meduna
Faculty of Information Technology, Brno University of Technology
imartisko@fit.vutbr.cz, meduna@fit.vutbr.cz

46 Excel@FIT 2016

Introduction

Context sensitive grammars (CSG):

CSGG $= (N, T, P, S)$, where rules are of one of the following forms (Kuroda normal form):

- $AB \rightarrow CD$
- $A \rightarrow CD$
- $A \rightarrow C$
- $A \rightarrow a$

(Propagating) scattered context grammars ((P)SCG):

SCGG $= (N, T, P, S)$, multiple symbols are rewritten at the same time. Order of these symbols in the sentential form is important. For example rule $(A, B, C) \rightarrow (a, b, c)$:

$$aaAbbBccC \Rightarrow aaabbbccc$$

We use only *propagating* SCGs (PSCG) — no ε rules.

Cooperating distributed grammar systems (CDGS):

CDGS $\Gamma = (N, T, S, P_0, P_1, \ldots, P_n)$, where Ps are sets of rules of some form. CDGS can work in several modes of derivation. We are using only the t mode — one P_i set is chosen, as long as it is possible, only rules from this set have to be used to do the rewriting of the sentential form. When this is no longer possible new P_i is chosen.

We combine SCGs and CDGSs and call this model scattered context grammar system (SCGS). This model works as standard CDGS where sets P_i contain propagating scattered context rules.

Problem Definition

What is known:

- $\mathcal{L}(PSCG) \subseteq \mathcal{L}(CSG)$.
- CDGS with CSG type rules is as powerful as CSG.

What is not known (long time open problem):

- $\mathcal{L}(CSG) \subseteq \mathcal{L}(PSCG)$.

What we do:

- $\mathcal{L}(SCGS) \subseteq \mathcal{L}(CSG)$ — "what is known".
- $\mathcal{L}(CSG) \subseteq \mathcal{L}(SCGS)$ — our main result.

Main Result

Basic ideas:

- We are simulating CSG using the SCGS.
- There are no ε rules.
- SCGS has two components (sets P_0 and P_1).
- Terminals are replaced by placeholder nonterminals that are rewritten in the end.
- The first component – applies a rule of the simulated CSG.
- The second component – makes sure that the rule was used correctly.

The first component:

- Simulates a rule of the CSG.
- Uses special marks $({}_{|}, {}_{<}, {}_{>})$ that denote context free and context sensitive symbols, respectively.
- Works in two phases (leftmost symbol is marked with + or - to indicate current phase).

Phase one (apply rule):

$${}^{+}AAABBB \Rightarrow {}^{-}A_{||}C_{<}AB_{>}D_{|}B$$

Phase two (mark remaining symbols with ${}_{|}$):

$${}^{-}A_{||}C_{<}AB_{>}D_{|}B \Rightarrow^{*} {}^{-}A_{|}{}_{|}C_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|}$$

The second component:

- Checks that the first component applied the rule correctly.
- Special mark $(\wedge)$ is passed along the sentential form.
- Pairs of symbols are checked — the one with $(\wedge)$ and a symbol right of it — $\wedge$ can never move left.
- Skipped symbols are replaced by blocking symbols (!) in the end. This is ensured by the t mode.

$${}^{-}A_{|}{}_{|}C_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|} \Rightarrow {}^{+}A^{\wedge}_{|}{}_{|}C_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|}$$

$${}^{+}A^{\wedge}_{|}{}_{|}C_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|} \Rightarrow {}^{+}AC^{\wedge}_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|}$$

$${}^{+}AC^{\wedge}_{<}{}_{|}A_{|}{}_{|}B_{|}{}_{>}D_{|}{}_{|}B_{|} \Rightarrow {}^{+}AC{}_{|}A_{|}{}_{|}B_{|}D^{\wedge}_{|}{}_{|}B_{|}$$

$${}^{+}AC{}_{|}A_{|}{}_{|}B_{|}D^{\wedge}_{|}{}_{|}B_{|} \Rightarrow^{*} {}^{+}AC{}_{|}A_{|}{}_{|}B_{|}DB$$

$${}^{+}AC{}_{|}A_{|}{}_{|}B_{|}DB \Rightarrow^{*} {}^{+}AC!!DB$$

Main result:

$\mathcal{L}(CSG) = \mathcal{L}(SCGS)$.