PARSING BASED ON SCG

Petr BLATNÝ, Master Degree Programme (5)
Dept. of Information Systems, FIT, BUT
E-mail: xblatn00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

The family of languages generated by scattered context grammar is contained in the family of context sensitive languages and contains all languages accepted by linear time nondeterministic Turing machines.

1 ÚVOD

Hlavním rysem gramatik s rozptýleným kontextem (SCG) je možnost aplikovat více pravidel v jediném derivačním kroku. Síla těchto gramatik převyšuje sílu bezkontextových gramatik a popisuje třídu jazyků typu 0.

2 GRAMATIKA S ROZPTÝLENÝM KONTEXTEM (SCG)

SCG je čtveřice $G = (N,T,P,Z)$ kde:

N - konečná množina nonterminálních symbolů	P - konečná množina pravidel tvaru:
T - konečná množina terminálních symbolů	$(A_1,\ldots,A_n) \rightarrow (\omega_1,\ldots,\omega_n) \in P$
Z - počáteční symbol gramatiky	$\omega_i \in ((N \setminus \{Z\} \cup T)^+, A_i \in T, n \geq 1$

3 POPIS ALGORITMU I

3.1 POMOCNÉ PROSTŘEDKY

Set Q:
For every $(A_1,\ldots,A_n) \rightarrow (\omega_1,\ldots,\omega_n) \in P$
add productions $A_1 \rightarrow \omega_1,\ldots,A_n \rightarrow \omega_n$ into Q where $n \geq 1$
Function $first(index)$ return first component of index
Function $second(index)$ return second component of index

3.2 ANALÝZA ZDOLA NAHORU

input string $a_1 \dots a_m$ for some $m \geq 1$

1. $S[i,i] = a_i \qquad 1 \leq i \leq m$
2. Apply the following rules until no set $S[i,j]$ can be changed:

   if $A \rightarrow x_1 \dots x_z \in Q$ **and** $x_\alpha \in S[index_\alpha]$ **then**
   add $x_1 \dots x_z$ into $S[i, second(index_z)]$ **and add** $[index_\alpha]$ into $x_\alpha[i, second(index_z)]$
   for every i,j such that $1 \leq i \leq j \leq m$, $x_\alpha \in N \cup T$ and
   for every $index_\alpha$ such that $first(index_\alpha) \leq second(index_\alpha)$ and
   $second(index_\alpha) = first(index_{\alpha+1}) + 1$, z is positive integer, $1 \leq \alpha \leq z$

   if $A \rightarrow \omega \in P$ **and** $\omega \in S[i,j]$ **then**
   add A into $S[i,j]$ **and add** $[i,j]$ into $A[i,j]$
   for every i,j such that $1 \leq i \leq j \leq m$, $\omega \in (N \cup T)^+$

   if $(A_1, \dots, A_z) \rightarrow (\omega_1, \dots, \omega_z) \in P$ **and** $\omega_\alpha \in S[index_\alpha]$ **then**
   add A_α into $S[index_\alpha]$ **and add** $[index_{\alpha-1}, index_{\alpha+1}]$ into $A_\alpha[index_\alpha]$
   for every i,j such that $1 \leq i \leq j \leq m$ and for every $index_\alpha$ such that
   $first(index_\alpha) \leq second(index_\alpha)$ and $second(index_\alpha) \leq first(index_{\alpha+1}) + 1$

3. **if** $Z \in S[1,m]$ then apply following rules, **else** end analysis with negative decision

3.3 ANALÝZA SHORA DOLŮ

1. **for** every $A, \omega \in S[i,j]$ **and** $[i,j] \in A[i,j]$ **find** every ω such that
   $A \rightarrow \omega \in Q$, $\omega = x_1 \dots x_z$ **and go to** each $S[index]$ such that $[index] \in x_\alpha[i,j]$
   where $z = |\omega|$, $\omega \in (N \cup T)^+$, $1 \leq \alpha \leq z$, $1 \leq i \leq j \leq m$
   and make new parallel tree by a copy of the tree under construction.
2. **for** $(A_1, \dots, A_z) \rightarrow (\omega_1, \dots, \omega_z) \in P$ **and** $A_\alpha, \omega_\alpha \in S[index_\alpha]$
   if $[index_{\alpha-1}, index_{\alpha+1}] \in A_\alpha[index_\alpha]$ **then remove** all indices from $A_\alpha[index_\alpha]$
   put $[index_\alpha]$ into $A_\alpha[index_\alpha]$ in tree under construction and return to previous rule

3.4 POČÁTEČNÍ HODNOTY A PODMÍNKA AKCEPTOVÁNÍ VĚTY

Initialize set $A = Z$, $i = 1$, $j = m$, Consider every possible analysis made by the algorithm.
If at least one of them ends in each set $S[u,u]$ with $1 \leq u \leq m$, the input string is accepted.

4 POPIS ALGORITMU II

4.1 POMOCNÉ PROSTŘEDKY

Set Q, **function** $first(index)$, **function** $second(index)$
Structure of symbol: symbol has two sets which contains numbers of terminals
cover: contains terminals covered by the symbol.

shadow cover: contains lists of terminals covered by symbols adherent to the symbol.
Table of indices: table contains indices of each terminal, initial value of each index is 1. Terminals are numbered by our position in input string.
Function *last_index*(number of input terminal): return index from *table of indices*.
Function *inc_cover*(terminal, *symbol*): increment index from *table of indices* and add it to *cover* of *symbol*.
Function *cover*(*num* =number of input terminal): add *num* with $last\_index(num)$ to set *cover* and $inc\_index(num)$.
Function *shadow_cover*(num = number of input terminal): add *num* with $last\_index(num)$ to set *shadow cover* and $inc\_index(num)$.
Function $copy\_sets(sym_1, sym_2 = symbol)$: copy *cover* and *shadow cover* from sym_1 to sym_2.

4.2 ANALÝZA POUZE ZDOLA NAHORU

input string $a_1 \dots a_m$ for some $m \geq 1$

1. $S[i,i] = a_i$ $\quad cover(i) \quad$ $1 \leq i \leq m$
2. Apply the following rules until no set $S[i,j]$ can be changed

   if $A \rightarrow x_1 \dots x_z \in Q$ **and** $x_\alpha \in S[index_\alpha]$ **then**
   add $x_1 \dots x_z$ with sets *cover* and *shadow cover* into $S[i, second(index_z)]$
   for every i, j such that $1 \leq i \leq j \leq m$, $x_\alpha \in N \cup T$
   and for every $index_\alpha$ such that $first(index_\alpha) \leq second(index_\alpha)$ and
   $second(index_\alpha) = first(index_{\alpha+1}) + 1$, where z is positive integer, $1 \leq \alpha \leq z$.

   if $A \rightarrow x_1 \dots x_z \in P$ **and** $x_1 \dots x_z \in S[i,j]$ **then**
   add A into S[i,j] **if** $x_\alpha \in N$ **then** $copy\_sets(x_\alpha, A)$ **else** $inc\_cover(x_\alpha, A)$
   if *cover* from x_α= some list L from *shadow cover* x_α **then**
   remove this list L from *shadow cover* x_α
   if *cover* from x_α contains some terminal(s), but not all or with others indices, as same list from *shadow cover* **then rollback** this rule and **block** it.
   for every i, j such that $1 \leq i \leq j \leq m$

   if $(A_1, \dots, A_z) \rightarrow (\omega_1, \dots, \omega_z) \in P$ **and** $\omega_\alpha \in S[index_\alpha]$ **then**
   if $x_\beta \in N$ **then** $copy\_sets(x_\beta, A_\alpha)$ **else** $inc\_cover(x_\beta, A_\alpha)$ **and**
   add all *cover* from remaining A to *shadow cover* as new list or add to current list.
   if in one of A no terminal added to *shadow cover* **then** return rule and **block** it.
   for every i, j such that $1 \leq i \leq j \leq m$ and for every $index_\alpha$ such that
   $first(index_\alpha) \leq second(index_\alpha)$ and $second(index_\alpha) \leq first(index_{\alpha+1}) + 1$,
   $z = |\omega|$, $\omega = x_1 \dots x_v$, $1 \leq \alpha \leq z$, $x_\beta \in N \cup T$, $1 \leq \beta \leq v$
3. **If** $Z \in S[i,m]$ then input string is accepted, **else** end analysis with negative decision.

REFERENCE

[1] Meduna, A.: Automata and Languages, Springer, London 2000