Compiler Construction WS09/10

Exercise Sheet 3

Please hand in the solutions to the theoretical exercises until the beginning of the lecture next Wednesday 2009-11-11, 10:00. Please write the number of your tutorial group or the name of your tutor on the first sheet of your solution. Solutions submitted later will not be accepted.

Exercise 3.1: Item-PDAs Revisited (Points: 4+2)

Let the pushdown automaton \( P = (\{a, b\}, \{q_0, q_1, q_2, q_3\}, \Delta, q_0, \{q_3\}) \), where
\[
\Delta = \{(q_0, a, q_0q_1), (q_0, b, q_0q_2), (q_0, \#, q_3), (q_1, a, q_1q_1), (q_1, b, \epsilon), (q_2, a, \epsilon), (q_2, b, q_2q_2)\}
\]
and \( \# \notin \Sigma \) symbolizes the end of the input word, be given.

Provide a context-free grammar that generates the language \( L \) accepted by \( P \). If possible, provide also a regular expression for \( L \). Otherwise provide sufficient arguments why this is not possible.

Exercise 3.2: Grammar Flow Analysis (Points: 3+6)

Let \( G = (\{S', S, A, B, C, D, E, F, G, H, K, L\}, \{a, b, c, d, e\}, P, S') \) be a given grammar with the set of productions \( P \) defined as:

\[
egin{align*}
S' & \rightarrow S \\
S & \rightarrow BH | HA \\
A & \rightarrow SaBC | bcA \\
B & \rightarrow Ba | b \\
C & \rightarrow dS | Bd \\
D & \rightarrow deL \\
E & \rightarrow FG \\
G & \rightarrow b \\
H & \rightarrow cA | A | b \\
K & \rightarrow b \\
L & \rightarrow dD
\end{align*}
\]

1. Remove all unreachable and all non-productive rules.

2. Compute the sets \( FIRST_1(T) \) and \( FOLLOW_1(T) \) for each nonterminal \( T \) of the reduced grammar.

You are to use the algorithms from the lecture and to provide for each subtask the corresponding system of equations.

Exercise 3.3: LL(k) (Points: 2+2+2+2)

A grammar is an LL(k)-grammar for some \( k \in \mathbb{N} \) if whenever there exist \( u, x, y \in V_T^* \) with \( k : x = k : y, Y \in V_N \) and \( \alpha, \beta, \gamma \in (V_T \cup V_N)^* \) such that

\[
\begin{align*}
S & \xrightleftharpoons[+]{lm} uY\alpha \xrightarrow[+]{lm} u\beta\alpha \xrightarrow[+]{lm} u\gamma\alpha \xrightarrow[+]{lm} u\gamma y \\
S & \xrightleftharpoons[+]{lm} u\beta\alpha \xrightarrow[+]{lm} u\gamma\alpha \xrightarrow[+]{lm} u\gamma y
\end{align*}
\]
$\beta = \gamma$

A language $L$ is an $LL(k)$-language if there exists an $LL(k)$-grammar that generates $L$.

1. Prove that for each $k \in \mathbb{N}$ there exists a grammar which is $LL(k + 1)$ but not $LL(k)$.

2. Prove that for each $k \in \mathbb{N}$ an $LL(k)$-grammar is an $LL(k + 1)$-grammar.

3. Investigate the relationship between $LL(0)$-languages and regular languages.

4. A grammar is left-recursive if it has a production of the form $A \rightarrow A\mu$. Show that a left-recursive grammar is not $LL(k)$ for any $k$. 