02/10/2010

Recitation #4 – Worksheet

Grammars

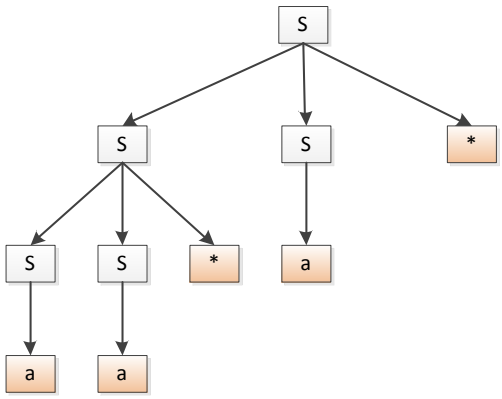
1. Consider the context-free grammar $G = (\{S\}, \{a\}, S, P)$:

   $S \rightarrow SS+ \mid SS* \mid a$

   a. Show how the string $aa+a*$ can be generated by this grammar.

   $S \rightarrow SS* \rightarrow Sa* \rightarrow SS+a* \rightarrow Sa+a* \rightarrow aa+a*$

   b. Construct a parse tree for this string.



   c. What language does this grammar generate? Justify your answer!

   All strings that start with 2 aa's and that consist of substrings of $a*$ and $a+$

2. Write an unambiguous grammar that leads to correct parse trees for the language consisting of expressions involving the operand **id** and the operators described below.

Operator	Associativity	Precedence	Binary / Unary
!	Left to right	Highest (4)	Binary
@	Right to left	High (3)	Unary
^	Right to left	Medium (2)	Binary
#	Left to right	Lowest (1)	Binary

Parentheses are also allowed, with their usual interpretation.

Solution:

$$A_1 \to A_1 \# A_2 \mid A_2$$
$$A_2 \to A_3 \wedge A_2 \mid A_3$$
$$A_3 \to @ A_3 \mid A_4$$
$$A_4 \to A_4 ! A_5 \mid A_4$$
$$A_5 \to \mathrm{id} \mid (A_1)$$

3. The following grammar contains occurrences of left recursion. Rewrite it so that there is no left recursion. Also do left factoring to remove rules for a single non-terminal that start with the same sequence of symbols.

$$G = \left(\{A_1, A_2, A_3\}, \{a, b\}, A_1, P\right)$$

where the production rules P are:

$$A_1 \to A_1 aa \mid A_2 b$$
$$A_2 \to A_1 aa \mid A_2 A_3 b$$
$$A_3 \to aA_3 b \mid ab$$

Solution:

First, remove the immediate left recursion in A_1 and A_2 by transforming it into right recursion**:**

$$A_1 \rightarrow A_2 b R_1$$
$$R_1 \rightarrow aaR_1 \mid \varepsilon$$
$$A_2 \rightarrow A_1 aaR_2$$
$$R_2 \rightarrow A_3 bR_2 \mid \varepsilon$$
$$A_3 \rightarrow aA_3 b \mid ab$$

There is still a left recursion involving a two-step derivation: $A_2 \Rightarrow A_2 bR_1 aaR_2$.
An algorithm for handling these kinds of non-immediate left recursions is described in the "Systems Software Knights" textbook on pages 285 – 286.

In essence, we keep rewriting the grammar until we only have immediate left recursions left:

$$A_2 \rightarrow A_2 bR_1 aaR_2$$
$$A_1 \rightarrow A_2 bR_1$$

Now simply transform the remaining recursion:

$$A_2 \rightarrow R_3$$
$$R_3 \rightarrow bR_1 aaR_2 R_3 \mid \varepsilon$$

Then, do the left factoring. The only non-terminal with multiple productions with a common prefix is A_3 so this rule will have to change to:

$$A_3 \rightarrow aA_3'$$
$$A_3' \rightarrow A_3 b \mid b$$

The final transformed grammar looks like this:

$$A_1 \rightarrow A_2 bR_1$$
$$R_1 \rightarrow aaR_1 \mid \varepsilon$$
$$A_2 \rightarrow R_3$$
$$R_3 \rightarrow bR_1 aaR_2 R_3 \mid \varepsilon$$
$$R_2 \rightarrow A_3 bR_2 \mid \varepsilon$$
$$A_3 \rightarrow aA_3'$$
$$A_3' \rightarrow A_3 b \mid b$$