Writing Assignment II

Released: Sunday, 20/07/99

Due: Sunday, 04/08/99 at 11:59pm

This assignment covers lecture notes 4 and 5 about top-down parsing. You can discuss and work out this assignment with other students. However, your write-up must be your own individual work. A PDF format of the solution (with your full name and student number) should be uploaded in the course page in Quera before 11:59 PM, Sunday 04/08/99 (https://quera.ir/course/6364/). Submissions with more than 50 hours delay will not be graded. Submissions with less than 50 hours delay will be penalized by the following rule:

Penalized mark = M * (100 - 2 * D) / 100

Where M = the mark achieved from your solution and D is number of hours passed the deadline.

1. Write recursive descent subroutines for the following grammar. Then modify these subroutines and the Match subroutine in a way that the parser can recover from the syntax errors using the Panic Mode method. The error recovery behavior of the parser should be almost identical to the behavior of an LL(1) parser using the algorithm mentioned on page 24 of Lecture Note 5 (Note that we don't have an explicit parsing stack in Recursive Descent Parsing).

   1   E → T X
   2-3 T → ( E )  |  \textbf{int} Y
   4-5 X → + E  |  \varepsilon
   6-7 Y → * T  |  \varepsilon

2. Left factor the following grammar:

   S → ( S )
   S → ( ) S
   S → ( S ) S
   S → ( )

3. Eliminate left recursion from the following grammar:

   E → E + T  |  T
   T → T F  |  F
   F → F *  |  ( E )  |  b

4. Explain why the following grammar is not LL(1):

   1-2 S → d A S  |  b
   3-4 A → B C  |  d
   5-6 B → a  |  \varepsilon
   7-8 C → b  |  \varepsilon
5. Suppose there is a grammar $G$ with at least two non-terminals $A$ and $B$. Also, assume non-terminal $A$ has exactly two productions rules ($A \to \alpha \mid \beta$) with the following properties:

   a. $\alpha \neq \varepsilon$
   b. $\alpha \Rightarrow^* \varepsilon$
   c. $\beta \not\Rightarrow^* \varepsilon$
   d. $\text{First}(\alpha) \cap \text{First}(\beta) = \emptyset$
   e. $\text{First}(\beta) \cap \text{Follow}(A) = \emptyset$
   f. $\text{First}(\alpha) \cap \text{Follow}(A) \neq \emptyset$

   Find out which one of the following propositions about this grammar is correct. Justify your answer. Note that we do not know if there exist other non-terminals besides $A$ and $B$. We also do know the exact number of rules that $B$ may have.

   - $G$ is LL(1).
   - $G$ is not LL(1).
   - The given information is not enough to determine whether or not $G$ is LL(1).

6. Compute the FIRST and FOLLOW functions for the non-terminals of the following grammar. Then construct the parse table of the grammar. Modify the parse table so that syntax errors can be recovered by the Panic Mode method (i.e., add 'synch' to the appropriate cells). Finally, show the sequence of stack, input, and action configurations that occur during an LL(1) parse of the string 'for id and = id id ; ; ) ; $'. Recover from syntax errors using the Panic Mode method. Issue an appropriate error message in situations where an error is detected. Note that you must not simplify or change the given grammar!

   

Ghassem Sani, Gholamreza,
20/08/99