Arizona State University
Computer Science and Engineering
CSE 340 - Spring 2019

Homework 2

Due Wednesday January 30th 2018 by 11:59 PM
This is different from the tentative date in the syllabus

Your solution should be typed and the questions should be answered in order. When you submit the homework, you should submit only one file not multiple files.

Problem I. Parsing. Consider the grammar

$$
\begin{array}{lcll}
S & \rightarrow & A\,g\,BC & (1)\\
A & \rightarrow & a\,A & (2)\\
A & \rightarrow & C\,B & (3)\\
B & \rightarrow & d\,B\,c & (4)\\
B & \rightarrow & \epsilon & (5)\\
C & \rightarrow & f\,C & (6)\\
C & \rightarrow & \epsilon & (7)
\end{array}
$$

where S, A, B, and C are the non-terminals, S is the start symbol, and a, c, d, e, f, and g are the terninals.

1. Calculate the FIRST and FOLLOW sets for this grammar.

   - For FIRST sets
     (a) Do an initialization pass by applying FIRST sets rules I and II.
     (b) Do successive passes, on the grammar rules in the order they are listed and apply to each grammar rule FIRST set rules III, then FIRST set rule IV then FIRST set rule V

     In your answer, you should show the FIRST sets after each pass. You do not have to use the notation I used in class to indicate the order in which the elements are added to the sets.
   - For FOLLOW sets, you should do:
     (a) Do an initialization pass by applying FOLLOW sets rule I
     (b) Do **one pass** on all grammar rules, in the order they are listed, and apply to each grammar rule FOLLOW set rules IV and V.
     (c) Do successive passes on all grammar rules in the order they are listed and apply to each grammar rule FOLLOW set rules II and III until there is no change.

     In your answer, you should show the FOLLOW sets after each pass. You do not have to use the notation I used in class to indicate the order in which the elements are added to the sets.

2. Show that the grammar has a predictive recursive descent parser

3. Write the parser for the grammar. Your parser should follow the general model of predictive parser that we saw in class. In particular, for non-terminals that can generate ϵ, the parser should check the FOLLOW set before choosing to parse the righthand side that generates ϵ.

4. Give a full execution trace for your parser from part (a) above on input $c\,d\,e$.