Assignment #2

Date Due: February 16, 2018

Total: 100 marks

Instructions

INSTRUCTIONS

Combine all your files into a single compressed file. Please use a filename that includes your UPEI username and is of the form: username_a2.zip; submit the zip file via moodle.

Step 1

Assume your grammars have $V_N = \{A, B, \ldots, Z\}$ and $V_T = \{a, b, c, \ldots, z\}$. Productions $A \to \alpha$, of a given grammar are stored in a file as

$A\ \alpha$

one per line and the first nonterminal on the first line gives the axiom of the grammar.

1. (10 marks) Write programs/methods/functions to compute $FIRST(\alpha)$, for a given string $\alpha \in (V_n \cup V_T)^*$.

2. (10 marks) Write programs/methods/functions to compute $FOLLOW(A)$, for any nonterminal of the grammar.

3. (10 marks) Write programs/methods/functions to compute $LEFT(A)$, for any nonterminal of the grammar.

4. (10 marks) Write programs/methods/functions to compute $RIGHT(A)$, for any nonterminal of the grammar.

   Step 2

5. (20 marks) Write a program that will build the precedence table and decide if your grammar is

   (a) Simple precedence grammar;

   (b) Weak precedence grammar;

   (c) Not a precedence grammar.

6. (10 marks) Given a grammar and a word, use your program to find a rightmost derivation. In case the word is not in the language, produce appropriate error messages.

   Step 3

7. (10 marks) Write a program that will test if a given grammar is $LL(1)$
8. (10 marks) Compute the $LL(1)$ parsing table for a given grammar
9. (10 marks) Given a grammar and a word, use your program to find a leftmost derivation. In case the word is not in the language, produce appropriate error messages.

   Step 4

10. (10 marks) Design a number of test cases to provide reasonable confidence that your programs are correct. Run your test cases and produce a file for the output of each test (record the execution in a file). Provide an overall summary of your test runs and the confidence you have that it is working correctly. Submit all program code and any relevant output from program testing.