COSC 4785/5785
Compiler Construction I, Fall 2022
Homework 2

Question 1: [6 points]

Consider the string ***aa + a**** and the following context-free grammar.

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

(a) [2 points] Give a leftmost derivation for the string.

(b) [2 points] Give a rightmost derivation for the string.

(c) [2 points] Give a parse tree for the string.

NOTE: Make sure your notation is correct especially for the derivations. You must give **all** the steps of the derivation.

Question 2: [10 points]

A grammar, G, is LL(1) if and only if whenever $A \rightarrow \alpha|\beta$ are two distinct productions of G, the following conditions hold:

1. For no **terminal a** do both α and β derive strings beginning with **a**.
2. At most one of α and β can derive the empty string.
3. If $\beta \stackrel{*}{\Rightarrow} \epsilon$, then α does derive any string beginning with a terminal in FOLLOW(A) Likewise if $\alpha \stackrel{*}{\Rightarrow} \epsilon$, then β does derive any string beginning with a terminal in FOLLOW(A)

Design LL(1) grammars for the following languages (and they CAN be designed):

(a) [5 points] The set of all strings of 0s and 1s such that every 0 is immediately followed by at least one 1.

(b) [5 points] The set of all even length (0, 2, 4, etc) strings of 0s and 1s that are *palindromes*; that is, the string reads the same backward and forward.

See next page for more questions.

Question 3: [20 points]

The following is a simple grammar for regular expressions over symbols a and b only, using $+$ in place of | for union (that way we avoid confusing it with the use of the vertical bar as a grammar metasymbol):

$$\begin{array}{lcl} rexpr & \rightarrow & rexpr \ + \ rterm \mid rterm \\ rterm & \rightarrow & rterm \ rfactor \mid rfactor \\ rfactor & \rightarrow & rfactor \ * \mid rprimary \\ rprimary & \rightarrow & \mathbf{a} \mid \mathbf{b} \end{array}$$

(a) [5 points] Left factor this grammar.

(b) [5 points] Does left factoring make the grammar suitable for top-down parsing? Briefly explain your answer.

(c) [5 points] In addition to left factoring, eliminate left-recursion from the original grammar.

(d) [5 points] Now is the resulting grammar suitable for top-down parsing? Briefly explain your answer.

Question 4: [20 points]

Compute FIRST() and FOLLOW() for the following grammar and then construct a parsing table for the grammar. See sections 4.4.2 and 4.4.3 in the text. Note that if the grammar is left-recursive you must first remove the left-recursion.

$$\begin{array}{lcl} S & \rightarrow & S + S \\ S & \rightarrow & SS \\ S & \rightarrow & (S) \\ S & \rightarrow & S* \\ S & \rightarrow & \mathbf{a} \end{array}$$