Ben Smith
Email: bxs566@case.edu
Course: CSDS 337 - Compiler Design
Instructor: Dr. Vipin Chaudhary

Problem Set - 3
ID: 3559750
Term: Spring 2024
Due Date: 28^{th} Feb, 2024

Number of hours delay for this Problem Set: 0
Cumulative number of hours delay so far: 26

I discussed this homework with: Jackson Schuetzle.

Problem 1 - 10 points
Design grammar for the following language:

- The set of all strings of 0s and 1s such that every 0 is immediately followed by at least one 1.

Solution:
$S \rightarrow 0x \quad | \quad 1S \quad | \quad \epsilon$
$x \rightarrow 1S$

Problem 2 - 20 points
The following is a grammar for regular expressions over symbols a and b only, using + in place of | for union, to avoid conflict with the use of vertical bar as a metasymbol in grammars:

$rexpr \rightarrow rexpr + rterm \quad | \quad rterm$
$rterm \rightarrow rterm \quad rfactor \quad | \quad rfactor$
$rfactor \rightarrow rfactor * \quad | \quad rprimary$
$rprimary \rightarrow a|b$

a Left factor this grammar.

b Does left factoring make the grammar suitable for top-down parsing?

c In addition to left factoring, eliminate left recursion from the original grammar.

d Is the resulting grammar suitable for top-down parsing?

Solution: Your solutions go here

a The grammar is already left factored. No nonterminal has two or more productions beginning with the same symbol.

b No because the grammar is left recursive.

c $rexpr \rightarrow A_1$
$A_1 \rightarrow +rterm \quad A_1 \quad | \quad \epsilon$
$rterm \rightarrow rfactor \quad A_2$
$A_2 \rightarrow rfactor \quad A_2 \quad | \quad \epsilon$
$rfactor \rightarrow rprimary \quad A_3$
$A_3 \rightarrow *A_3 \quad | \quad \epsilon$
$rprimary \rightarrow a \quad | \quad b$

d Yes because it is left-factored and right-recursive.

Problem 3 - 20 points
Consider the grammar for $S \longrightarrow S+S|SS|(S)|S*|a$ and the string $(a+a)*a$.

a Devise a predictive parser and show the parsing tables. You may use left-factor and/or eliminate left-recursion from your grammar first.

b Compute FIRST and FOLLOW for your grammar.

Solution:

a Left-factor the grammar:
$S \rightarrow SS' \quad | \quad (S) \quad | \quad a$
$S' \rightarrow +S \quad | \quad S \quad | \quad *$

Eliminate left recursion:
$S \rightarrow (S)A \quad | \quad aA$
$A \rightarrow S'A \quad | \quad \epsilon$
$S' \rightarrow +S \quad | \quad S \quad | \quad *$

b

Problem 4 - 10 points
Give the bottom-up parses for the following input string and grammar: $aaa*a++$ and $S \longrightarrow SS+|SS*|a$.

Solution:
$\rightarrow \underline{a}aa*a++$
$\rightarrow S\underline{a}a*a++$
$\rightarrow SS\underline{a}*a++$
$\rightarrow S\underline{SS*}a++$
$\rightarrow SS\underline{a}++$
$\rightarrow S\underline{SS+}+$
$\rightarrow \underline{SS+}$
$\rightarrow S$

Problem 5 - 20 points
Construct the SLR sets of items for the (augmented) grammar $S \longrightarrow SS+|SS*|a$. Compute the GOTO function for these sets of items. Show the parsing table for this grammar. Is this grammar SLR?

Solution:

Problem 6 - 20 points
Construct the canonical parsing table for the following augmented grammar:
$S' \longrightarrow S$
$S \longrightarrow AA$
$A \longrightarrow aA|b$

Solution: