

1. Eliminate any left-recursion or common left-factors in the following grammars:

a)

$S \rightarrow 0A$

$S \rightarrow 1B$

$A \rightarrow AB$

$A \rightarrow A1$

$A \rightarrow 01$

$B \rightarrow 01$

$B \rightarrow 00$

Solution:

$S \rightarrow 0A$

$S \rightarrow 1B$

$A \rightarrow N_1 A_{tail}$

$N_1 \rightarrow 01$

$N_1 \rightarrow A1$

$A_{tail} \rightarrow B A_{tail}$

$A_{tail} \rightarrow \lambda$

$B \rightarrow 0N_2$

$N_2 \rightarrow 1$

$N_2 \rightarrow 0$

b)

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow Id$

Solution:

$E \rightarrow N E_{tail}$

$N \rightarrow E * E$

$N \rightarrow (E)$

$N \rightarrow Id$

$E_{tail} \rightarrow + E E_{tail}$

$E_{tail} \rightarrow \lambda$

2. Which of the following grammars are LL(1)? Explain why. Show First, Follow, and Predict sets.

a)

$S \rightarrow ABBA$

$A \rightarrow a$

$A \rightarrow \lambda$

$B \rightarrow b$

$B \rightarrow \lambda$

First

$\text{First}(\langle S \rangle) = \text{First}(ABBA)$	$\{a, \lambda\}$
$\text{First}(\langle A \rangle) = \text{First}(a \mid \lambda)$	$\{a, \lambda\}$
$\text{First}(\langle B \rangle) = \text{First}(b \mid \lambda)$	$\{b, \lambda\}$

Follow

$\text{Follow}(S)$	$\{\lambda\}$
$\text{Follow}(A)$	$\{b, \lambda\}$
$\text{Follow}(B)$	$\{b, a, \lambda\}$

Predict

$\text{Predict}(S \rightarrow ABBA)$	$\{a\}$
$\text{Predict}(A \rightarrow a \mid \lambda)$	$\{a, b\}$
$\text{Predict}(B \rightarrow b \mid \lambda)$	$\{b\}$

This is LL(1) because the grammar does not contain a left recursive production or a common prefix production.

b)

$S \rightarrow aSe$

$S \rightarrow B$

$B \rightarrow bBe$

$B \rightarrow C$

$C \rightarrow cCe$

$C \rightarrow d$

This is LL(1) because the grammar does not contain a left recursive production or a common prefix production.

First

$\text{First}(\langle S \rangle) = \text{First}(aSe \mid B)$	$\{a, b, c, d\}$
$\text{First}(B) = \text{First}(bBe \mid C)$	$\{b, c, d\}$
$\text{First}(C) = \text{First}(cCe \mid d)$	$\{c, d\}$

Follow

$\text{Follow}(S)$	$\{e\}$
$\text{Follow}(B)$	$\{e\}$
$\text{Follow}(C)$	$\{e\}$

Predict

Predict($S \rightarrow aSe B$)	{a,b,c,d}
Predict($B \rightarrow bBe C$)	{b,c,d}
Predict($C \rightarrow cCe d$)	{c,d}

3. Implement a Grammar Analyzer. The input of this program is a set of productions written in the form of Micro Grammar. This means that all the nonterminals are quoted by $\langle \rangle$. The output of this program should be a list of productions, a list of all the symbols involved, a list of nonterminals, a list of terminal symbols, a list of right hand sides (RHS) of productions, and a list of left hand sides (LHS) of productions. You can create your own data structures. Grammar Analysis Algorithms, which we will develop soon, would have to use your Grammar Analyzer. Their implementation will be requested in the subsequent assignments.

Apply your Grammar Analyzer to the Micro Grammer and to another grammer of your choice; print the results