German University in Cairo
Department of Computer Science
Assoc. Prof. Haythem O. Ismail

CSEN 1003 Compiler, Spring Term 2019
Practice Assignment 5

Discussion: 26.02.19 - 03.03.18

Exercise 5-1

Consider the following CFG:

$$\begin{aligned} S &\rightarrow \texttt{0}T\texttt{1}S \mid \varepsilon \\ T &\rightarrow \texttt{0}T\texttt{1} \mid \varepsilon \end{aligned}$$

a) Compute the FIRST and FOLLOW sets.

Solution:

$\text{FIRST}(S) = \{\texttt{0}, \varepsilon\}$

$\text{FIRST}(T) = \{\texttt{0}, \varepsilon\}$

$\text{FOLLOW}(S) = \{\$\}$

$\text{FOLLOW}(T) = \{\texttt{1}\}$

b) Compute the parsing table.

Solution:

Non-terminal	Input Symbol		
	0	1	$
S	$\texttt{0}T\texttt{1}S$		ε
T	$\texttt{0}T\texttt{1}$	ε	

c) Prove that this grammar is LL(1).

Solution:

The grammar is LL(1) because every entry in parsing table contains at most one rule.

d) What language does this grammar recognize?

Solution:

$\{\texttt{0}^n\texttt{1}^n | n \geq 1\}^*$

Exercise 5-2

Consider the following CFG:

$$\begin{aligned} S &\rightarrow SAB \mid SBC \mid \varepsilon \\ A &\rightarrow \texttt{a}A\texttt{a} \mid \varepsilon \\ B &\rightarrow \texttt{b}B \mid \varepsilon \\ C &\rightarrow \texttt{c}C \mid \varepsilon \end{aligned}$$

a) Compute FIRST and FOLLOW sets for each non-terminal.

Solution:

Non-terminal	FIRST	FOLLOW
S	$\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \varepsilon\}$	$\{\$, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$
A	$\{\mathtt{a}, \varepsilon\}$	$\{\$, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$
B	$\{\mathtt{b}, \varepsilon\}$	$\{\$, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$
C	$\{\mathtt{c}, \varepsilon\}$	$\{\$, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$

b) Build the parsing table.

Solution:

Non-terminal	Input Symbol			
	a	b	c	$
S	SAB SBC ε	SAB SBC ε	SAB SBC ε	SAB SBC ε
A	aAa ε	ε	ε	ε
B	ε	bB ε	ε	ε
C	ε	ε	cC ε	ε

c) From the parsing table, show why the grammar is not LL(1).

Solution:

The grammar is not LL(1) because some entries of the parsing table contain more than one rule.

Exercise 5-3

Construct a parsing table for the following grammar. Is the grammar LL(1)?

$$
\begin{array}{lcl}
S & \rightarrow & AB \\
A & \rightarrow & \textbf{id}\ A \mid \textbf{num} \\
B & \rightarrow & CA \\
C & \rightarrow & \textbf{0}C \mid \textbf{1}
\end{array}
$$

Solution:

Non-terminal	Input Symbol				
	id	**num**	**0**	**1**	$
S	AB	AB			
A	**id** A	**num**			
B			CA	CA	
C			**0**C	**1**	

The grammar is LL(1) because every position in the parsing table contains at most one rule.

Exercise 5-4

Consider the following CFG:

$$\begin{array}{lcl} S & \to & \texttt{(}L\texttt{)} \mid \texttt{a} \\ L & \to & L\texttt{,}S \mid S \end{array}$$

a) Eliminate left recursion.

Solution:

$$\begin{array}{lcl} S & \to & \texttt{(}L\texttt{)} \mid \texttt{a} \\ L & \to & \texttt{(}L\texttt{)}L' \mid \texttt{a}L' \\ L' & \to & \texttt{,}SL' \mid \varepsilon \end{array}$$

b) Compute FIRST and FOLLOW sets for each non-terminal.

Solution:

$\text{FIRST}(S) = \{\texttt{(}, \texttt{a}\}$

$\text{FIRST}(L) = \{\texttt{(}, \texttt{a}\}$

$\text{FIRST}(L') = \{\texttt{,}, \varepsilon\}$

$\text{FOLLOW}(S) = \{\texttt{)}, \texttt{,}, \$\}$

$\text{FOLLOW}(L) = \{\texttt{)}\}$

$\text{FOLLOW}(L') = \{\texttt{)}\}$

c) Build the parsing table.

Solution:

Non-terminal	Input Symbol				
	a	,	(	)	$
S	a		(L)		
L	aL'		$(L)L'$		
L'		$,SL'$		ε	

d) Trace the operations of a predictive top-down parser on the string `(a,(a,a))`.

Exercise 5-5

Consider the following CFG:

$$S \to SS\texttt{+} \mid SS\texttt{*} \mid \texttt{a}$$

a) Eliminate left recursion and left factor the grammar.

Solution:

$$\begin{array}{lcl} S & \to & \texttt{a}S' \\ S' & \to & SS'' \mid \varepsilon \\ S'' & \to & \texttt{+}S' \mid \texttt{*}S' \end{array}$$

b) Compute FIRST and FOLLOW sets for each non-terminal.

Solution:

$\text{FIRST}(S) = \{\texttt{a}\}$

$\text{FIRST}(S') = \text{FIRST}(S) \cup \{\varepsilon\} = \{\texttt{a}, \varepsilon\}$

$\text{FIRST}(S'') = \{\texttt{+}, \texttt{*}\}$

$\text{FOLLOW}(S) = \text{FIRST}(S'') \cup \{\$\} = \{\texttt{+}, \texttt{*}, \$\}$

$\text{FOLLOW}(S') = \text{FOLLOW}(S) = \{\texttt{+}, \texttt{*}, \$\}$

$\text{FOLLOW}(S'') = Follow(S') = \{\texttt{+}, \texttt{*}, \$\}$

c) Build the parsing table.

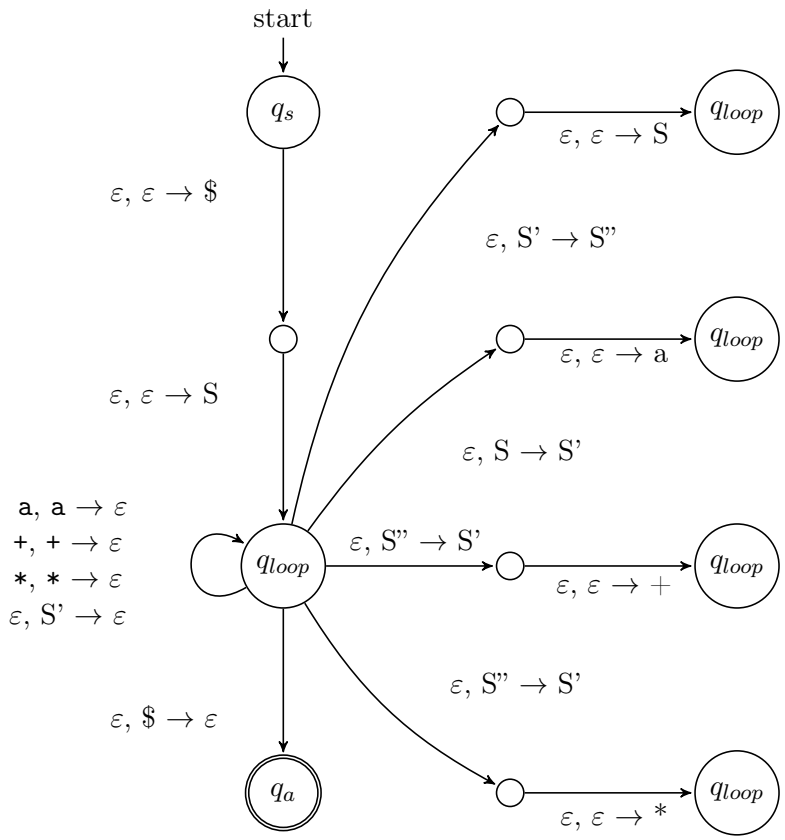
Solution:

Non-terminal	Input Symbol			
	a	+	*	$
S	$\texttt{a}S'$			
S'	SS''	ε	ε	ε
S''		$\texttt{+}S'$	$\texttt{*}S'$	

[1]

d) Draw the corresponding PDA for the above grammar.

Solution:



[1]Note that all the states labelled q_{loop} are the same state. They are only drawn as separate states to avoid clutter.

e) Trace the operations of a predictive top-down parser on the string `aa+`.