

Digital Assignment - Module 2

Derivation

1) Perform leftmost derivation and draw parse tree

$$S \rightarrow AIB$$

$$A \rightarrow 0A/\epsilon$$

$$B \rightarrow 0B/1B/\epsilon$$

output string: 1001

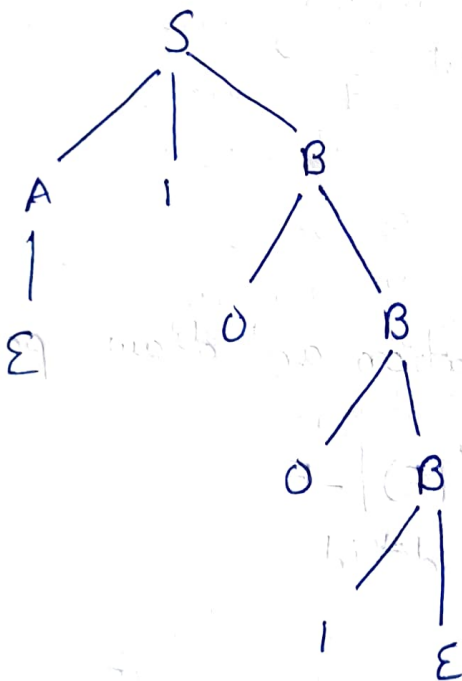
Sol. $S \rightarrow AIB$

$$S \rightarrow A10B \quad (B \rightarrow 0B)$$

$$S \rightarrow A100B \quad (B \rightarrow 0B)$$

$$S \rightarrow A1001B \quad (B \rightarrow 1B)$$

$$S \rightarrow 1001$$



2) Leftmost derivation and draw parse tree.

$$S \rightarrow 0S1/01$$

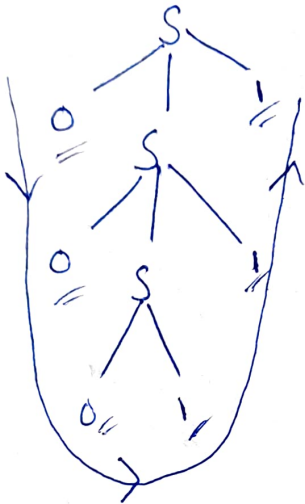
output string: 000111

sol

$$S \rightarrow 0S1$$

$$S \rightarrow 00S11 \quad (S \rightarrow 0S1)$$

$$S \rightarrow 000111 \quad (S \rightarrow 01)$$



3) Perform rightmost derivation and draw parse tree

$$E \rightarrow E+E/E \neq E/id/(E)/-E$$

output string: id + id * id

sol

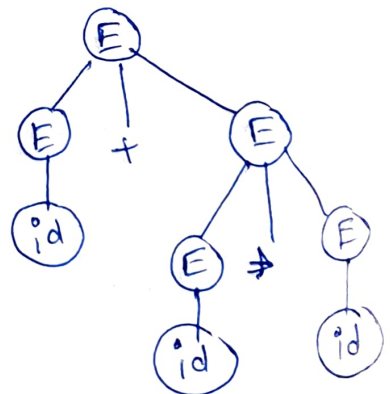
$$E \rightarrow E + E$$

$$\rightarrow E + E \neq E$$

$$\rightarrow E + E \neq id$$

$$\rightarrow E + id \neq id$$

$$\rightarrow id + id \neq id$$



4) $S \rightarrow as/sa/\epsilon$ (output string: aaaa)

$\rightarrow aSa \quad (S \rightarrow aS)$

→ aaSa (s → as)

→ 9995a (s → as)

→ a a a a ($s \rightarrow \epsilon$)


$$\rightarrow Sa a \quad (s \rightarrow Sa)$$

$\rightarrow Saqa$ ($S \rightarrow Sa$)

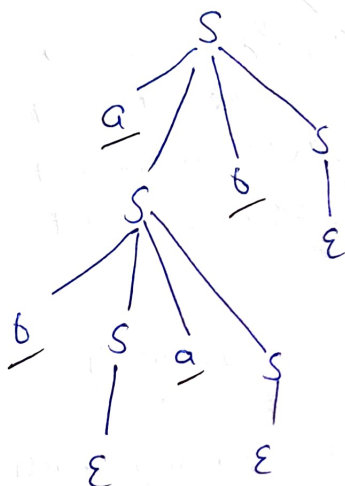
→ S a a a a (S → Sa)

$\rightarrow aaaa \quad (s \rightarrow \epsilon)$



5) $S \rightarrow aSbS / bSaS / \epsilon$ (output string: abab)

$S \rightarrow aSbS$
 $\rightarrow a**S**aSbS$
 $\rightarrow a**a**SbS$
 $\rightarrow a**a**bS$
 $\rightarrow a**a**b$



Left Recursion and Left Factoring

6 $A \rightarrow Abd / Aa / a$
 $B \rightarrow Be / b$

sol
 Left Factoring
 $A \rightarrow Abd / Aa / a$
 $A \rightarrow AA' / a$
 $A' \rightarrow bd / a$
 $B \rightarrow Be / b$

Left Recursion
 $A \rightarrow aA'$
 $A' \rightarrow bdA' / aA' / \epsilon$
 $B \rightarrow bB'$
 $B' \rightarrow eB' / \epsilon$

$$7 \quad A \longrightarrow AB/AC/a/b$$

Left factoring

$$A \longrightarrow AA'/a/b$$

$$A' \longrightarrow B/C$$

Left Recursion

$$A \longrightarrow aA'/bA'$$

$$A' \longrightarrow B/C/E$$

$$B \longrightarrow AB$$

$$C \longrightarrow AC$$

$$8 \quad S \longrightarrow iEtS/iE+sEs/a$$

Left Factoring

$$S \longrightarrow iEtSS'/a$$

$$S' \longrightarrow eS/E$$

Left Recursion

$$S \longrightarrow iEtSS'/as'$$

$$S' \longrightarrow eS/E$$

Top Down Parsing

9

$$S \rightarrow E$$

$$E \rightarrow id \mid TF$$

$$F \rightarrow +id \mid F/E$$

$$T \rightarrow (E) \mid \epsilon$$

Sol.

First

$$\text{First}(S) = \{id\}$$

$$\text{First}(E) = \{id\}$$

$$\text{First}(F) = \{+, E\}$$

$$\text{First}(T) = \{ (, \epsilon \}$$

Follow

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

$$\text{Follow}(E) = \{ \$,) \}$$

$$\text{Follow}(F) = \{ \$,) \}$$

$$\text{Follow}(T) = \{ \$,), + \}$$

Predictive Parsing Table

Non-Terminal	First	Follow
S	{id}	{ \$ }
E	{id}	{ \$,) }
F	{+, E}	{ \$,) }
T	{ (, E }	{ \$,), + }

Non-Terminal	Input Symbol				
	id	()	\$	+
S	$S \rightarrow E$				
E	$E \rightarrow id \mid TF$				
F			$F \rightarrow E$	$F \rightarrow E$	$F \rightarrow +id \mid TF$
T		$T \rightarrow (E)$	$T \rightarrow \epsilon$	$T \rightarrow \epsilon$	$T \rightarrow \epsilon$

As, there is only one entry in the table for terminal and non-terminal.

Therefore, the given grammar is LL(1).

Stack Implementation

Stack	Input	Output
S \$	id + id + id + id \$	$S \rightarrow E$
E \$	id + id + id + id \$	$E \rightarrow idTF$
id TF \$	id + id + id + id \$	Pop(id)
TF \$	+ id + id + id \$	$T \rightarrow E$
F \$	+ id + id + id \$	$F \rightarrow +idF$
+ id F \$	+ id + id + id \$	Pop(+)
id F \$	id + id + id \$	Pop(id)
F \$	+ id + id \$	$F \rightarrow +idF$
+ id F \$	+ id + id \$	Pop(+)
id F \$	id + id \$	Pop(id)
F \$	+ id \$	Pop $F \rightarrow +id$
+ id \$	+ id \$	Pop(+)
id \$	id \$	Pop(id)
\$	\$	accepted

Bottom Up Parsing

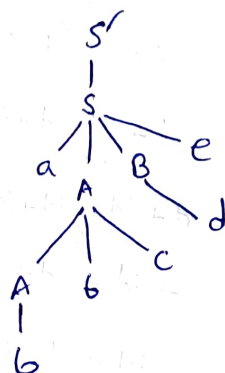
10 $S \rightarrow aABe$
 $A \rightarrow Abc/b$
 $B \rightarrow d$

Reduce a string abbcd to start symbol
 using LR(0), SLR(1), LALR(1), and CLR(1).

sol

LR(0) parsing

$S \rightarrow S$
 $S \rightarrow aABe$
 $A \rightarrow ~~Abc~~ Abc$
 $A \rightarrow b$
 $B \rightarrow d$



SLR(1) parsing:

	a	b	c	d	e	\$
S	1					
A	2	3	2			
B				4		

LALR(1) parsing:

	a	b	c	d	e	\$
S	1					
A	2	3	2			