

NEW HORIZON

COLLEGE OF ENGINEERING

FACD ASSIGNMENT- II

ASSIGNMENT NO:	2
GROUP NO:	1

SUBMISSION DATE	13/11/2020
-----------------	------------

STUDENT NAME	KRITESH POKHREL
USN	1NH18CS207

Q.1)

→

T ₁	0	0	1	1	2	2	B	B
T ₂	B	B	B	B	B	B	B	B

$$\delta(q_0, [0, B]) \rightarrow (q_1, [0, *], R)$$

$$\delta(q_1, [0, B]) \rightarrow (q_1, [0, B], R)$$

$$\delta(q_1, [1, B]) \rightarrow (q_2, [1, *], R)$$

$$\delta(q_2, [1, B]) \rightarrow (q_2, [1, B], R)$$

$$\delta(q_2, [2, B]) \rightarrow (q_3, [2, *], L)$$

$$\delta(q_2, [1, B]) \rightarrow (q_3, [1, B], L)$$

$$\delta(q_3, [1, *]) \rightarrow (q_3, [1, *], L)$$

$$\delta(q_3, [0, B]) \rightarrow (q_3, [0, B], L)$$

$$\delta(q_3, [0, *]) \rightarrow (q_0, [0, *], R)$$

T ₁	0	0	1	1	2	2	B	B
T ₂	*	B	*	B	*	B	B	B

$$\delta(q_0, [0, B]) \rightarrow (q_1, [0, *], R)$$

$$\delta(q_1, [1, *]) \rightarrow (q_1, [1, *], R)$$

$$\delta(q_1, [1, B]) \rightarrow (q_2, [1, *], R)$$

$$\delta(q_2, [2, *]) \rightarrow (q_2, [2, *], R)$$

$$\delta(q_2, [2, B]) \rightarrow (q_3, [2, *], L)$$

$$\delta(q_3, [2, *]) \rightarrow (q_3, [2, *], L)$$

$$\delta(q_3, [1, *]) \rightarrow (q_3, [1, *], L)$$

$$\delta(q_3, [1, *]) \rightarrow (q_3, [1, *], L)$$

$$\delta(q_3, [0, *]) \rightarrow (q_0, [0, *], R)$$

$$\delta(q_0, [1, *]) \rightarrow (q_0, [1, *], R)$$

$$\delta(q_0, [2, *]) \rightarrow (q_4, [2, *], R)$$

$$\delta(q_4, [B, B]) \rightarrow (q_f, [B, B], R)$$

T ₁	0	0	1	1	2	2	B	B
T ₂	*	*	*	*	*	*	B	B

Q-2)

→ The process of constructing the parse tree which starts from root and goes down to the leaf is Top-Down parsing. Top-down parsers construct from the Grammar which is free ambiguity and left recursion. It uses leftmost derivation to construct a parse tree. It allows a grammar which is free from left factoring.

Classification

- 1) with backtracking
- 2) without backtracking

e.g. i) $S \rightarrow aABe$

$A \rightarrow bc$

$B \rightarrow d$

$w \rightarrow abcde$

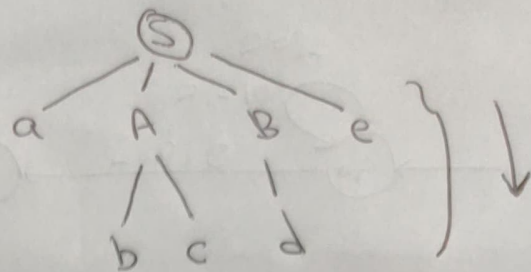


Fig: without backtracking

ii) $S \rightarrow rxz \mid rzd$

$x \rightarrow oa \mid ea$

$z \rightarrow a$

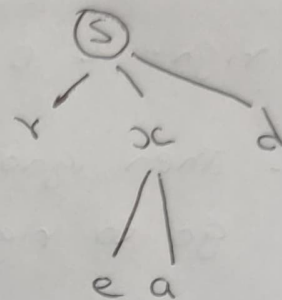
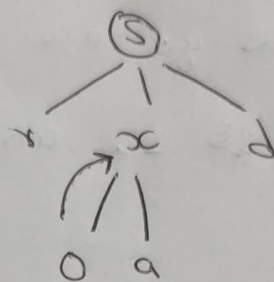


Fig: with backtracking

Q 3)

→

I/p: B 1 0 1 1 1 1 1 0 B

O/p: B 0 1 0 0 0 0 0 1 B

$\delta(q_0, 0) = (q_0, 1, R)$

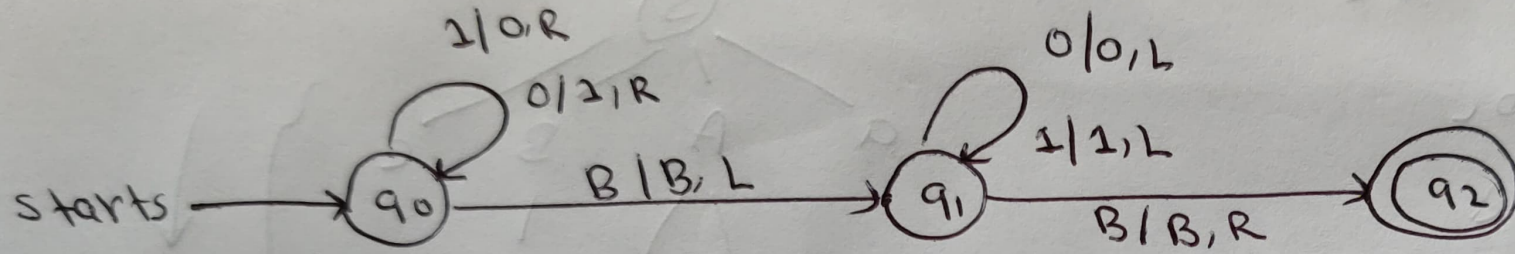
$\delta(q_0, 1) = (q_0, 0, R)$

$\delta(q_0, B) = (q_1, B, L)$

$\delta(q_1, 0) = (q_1, 0, L)$

$\delta(q_1, 1) = (q_1, 1, L)$

$\delta(q_1, B) = (q_2, B, R)$

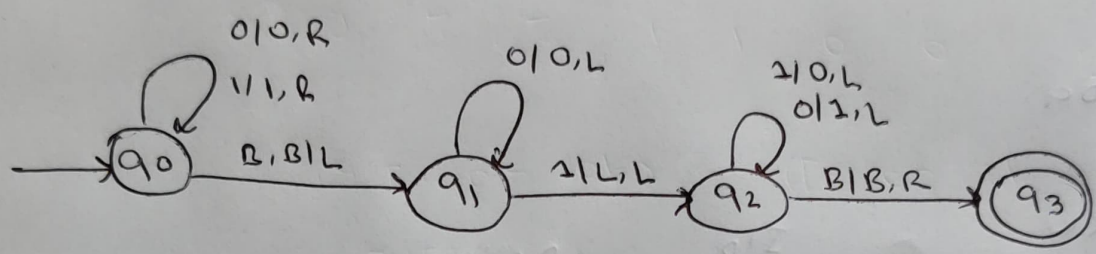


Steps.

1. Convert all 0's and 2's and all of 1's into 0's and go right if B found go left.
2. Then ignore 0's & 1's and go left and if B found go to right.
3. Stop the machine.

2's complement of a binary number is added to 1's complement of the binary

- i. Scanning input string from left to right
- ii. pass all consecutive 0's
- iii. For first 1 comes, do nothing.
- iv. After that, converting 1's into 0's & 0's into 1's
- v. Stop when B is reached.



4.

1. No left recursion in given production.
2. No left factoring in the given production.
3. First & follow of the production.

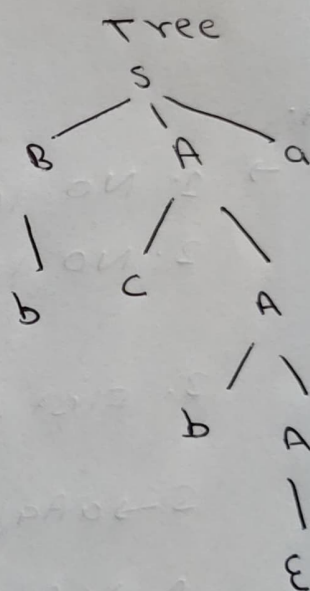
	First	Follow
$S \rightarrow aAa \mid BAa \mid \epsilon$	$\{a, b, \epsilon\}$	$\{\$ \}$
$A \rightarrow cA \mid bA \mid \epsilon$	$\{c, b, \epsilon\}$	$\{a\}$
$B \rightarrow b$	$\{b\}$	$\{a\}$

4. Parsing table

	a	b	c	\$
S	$S \rightarrow aAa$	$S \rightarrow bB$		$S \rightarrow \epsilon$
A	$A \rightarrow \epsilon$	$A \rightarrow bA$	$A \rightarrow cA$	
B		$B \rightarrow b$		

5. bcba

Stack	Top	Action
\$s	bcba\$	$S \rightarrow bAa$
\$aAB	bcba\$	$B \rightarrow b$
\$aAB	bcba\$	POP
\$aA	cba\$	$A \rightarrow cA$
\$aAc	cba\$	POP
\$aA	ba\$	$A \rightarrow bA$
\$aAb	ba\$	POP
\$aA	a\$	$A \rightarrow \epsilon$
\$a	a\$	POP
\$	\$	String Accepted



= bcb ϵ a
= bcba