Q. 7 For the following grammar

$D \rightarrow T\ L\ ;$

$L \rightarrow L\ ,\ id \mid id$

$T \rightarrow int \mid float$

1) Remove left recursion (if required)
2) Find first and follow for each non terminal for Resultant grammar
3) Construct LL(1) parsing table
4) Parse the following string (show stack actions clearly) and draw parse input: int id, id;

Ans. :

1. Remove left recursion

$L \rightarrow id\ L'$

$L' \rightarrow ,\ id\ L' \mid \in$

2. FIRST and FOLLOW for each non-terminal for resultant grammar.

$D \rightarrow TL\ ;$

$L \rightarrow id\ L'$

$L' \rightarrow ,\ id\ L' / \in$

$T \rightarrow int \mid float \mid$

$FIRST\ (T) = \{int, float\}$

$FIRST\ (D) = FIRST\ (T) = \{int, float\}$

FIRST (L) = {id}

FIRST (L′) = {, , ∈}

FOLLOW (D) = {$}

FOLLOW (L) = {;}

FOLLOW (T) = {id}

FOLLOW (L′) = {;}

3. L L (1) parsing table

	int	float	id	,	;	$
D	D → TL ;	D → TL ;				
L			L → idL′			
L′		C → bC		L′ → , idL′	L′ → ∈	
T	T → int	T → float				

4. Parsing of given input string int id, id;

$ D	int id, id ;	D → TL;
$; LT	int id, id ;	T → int
$; L int	int id, id ;	
$; L	id, id ;	L → idL′
$; L′ id	id, id ;	
4 ; L′	, id ;	
$; L′ id ;	, id ;	
$; L′	id ;	L → idL′
$; L′ id ;	id ;	
$; L′	;	L → ∈
$;	;	
$	;	

Accepted

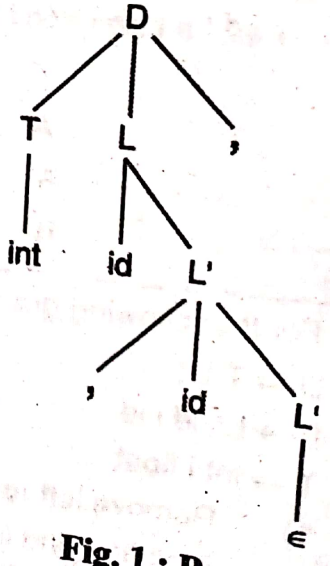

Fig. 1 . D

Test whether grammar is LL(1) or not & construct parsing table.

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \varepsilon$

$B \rightarrow \varepsilon.$

$FIRST(AaAb) = FIRST(A) - \{\varepsilon\} \cup FIRST(aAb) = \{a\}$

$FIRST(BbBa) = FIRST(B) - \{\varepsilon\} \cup FIRST(bBa) = \{b\}$

$FIRST(AaAb) \cap FIRST(BbBa) = \{a\} \cap \{b\} = \phi$

Hence the grammar is LL(1).

For parsing table,

$$\text{FIRST}(S) = \text{FIRST}(AaAb) \cup \text{FIRST}(BbBa) = \{a\} \cup \{b\} = \{a, b\}$$

$$\text{FIRST}(A) = \{\varepsilon\}$$

$$\text{FIRST}(B) = \{\varepsilon\}$$

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

Using $S \rightarrow AaAb$ we get :

$$\text{FOLLOW}(A) = \text{FIRST}(aAb) = \{a\} \text{ and}$$

$$\text{FOLLOW}(A) = \text{FIRST}(b) = \{b\}$$

$$\text{FOLLOW}(A) = \{a, b\}$$

$$S \rightarrow BbBa$$

$$\text{FOLLOW}(B) = \text{FIRST}(bBa) = \{b\}$$

$$\text{FOLLOW}(B) = \text{FIRST}(a) = \{a\}$$

$$\text{FOLLOW}(B) = \{a, b\}$$

Parsing table for the grammar

	a	b	$
S	$S \rightarrow AaAb$	$S \rightarrow BbBa$	
A	$A \rightarrow \varepsilon$	$A \rightarrow \varepsilon$	
B	$B \rightarrow \varepsilon$	$B \rightarrow \varepsilon$	

Q. 13 Test whether the following grammar is LL (1) or not. Construct predictive parsing tab

$S \rightarrow 1AB \mid \in$

$A \rightarrow 1AC \mid 0C$

$B \rightarrow 0S$

$C \rightarrow 1$

Ans. :

$$\begin{aligned} FIRST(S) &= \{1, \in\} \\ FIRST(A) &= \{1, 0\} \\ FIRST(B) &= \{0\} \\ FIRST(C) &= \{1\} \\ FOLLOW(A) &= \{FIRST(C) \cup FIRST(B)\} = \{1, 0\} \\ FOLLOW(B) &= FOLLOW(S) = \{\$\} \\ FOLLOW(C) &= FOLLOW(A) = \{1, 0\} \end{aligned}$$

The parsing table is constructed from above FIRST and FOLLOW functions.

	1	0	$
S	S →1AB		S→ε
A	A → 1AC	A→0C	
B		B →0S	
C	C→1		

No duplicate entries in parsing table Hence the given grammar is LL(1).

Q. 14 Find out FIRST and FOLLOW set for all the Nonterminals

$S \rightarrow AcB \mid cbB \mid Ba$

$A \rightarrow da \mid BC$

$B \rightarrow g \mid \epsilon$

$C \rightarrow h \mid \epsilon$

Ans. :

$$FIRST(C) = \{h, \epsilon\}$$

$$FIRST(B) = \{g, \epsilon\}$$

$$FIRST(A) = \{d \cup FIRST(B) \cup FIRST(C)\} = \{d, g, h, \epsilon\}$$

$$FIRST(S) = \{FIRST(A) \cup c \cup FIRST(B) \cup a\} = \{a, c, d, g, h, \epsilon\}$$

$$FOLLOW(A) = \{c\}$$

$$FOLLOW(C) = FOLLOW(A) = \{c\}$$

$$FOLLOW(B) = \{a \cup FIRST(c) \cup \$\} = \{a, h, \$\}$$

$$\mathbf{FOLLOW(S) = FOLLOW(B) = \{a, h, \$\}}$$

Q. 15 Draw parsing table for Table Driven Parser for the given grammar. Is the grammar LL(1

$A \rightarrow AaB \mid x \qquad B \rightarrow BCb \mid Cy \qquad C \rightarrow Cc \mid \epsilon$

Ans. : Grammar is not suitable for LL(1) parsing.

As Left recursive grammar cannot be LL(1) grammar.

After removing left recursion the grammar will be

$A \rightarrow x\ A'$	First (A) ={x}	Follow (A) ={$}
$A' \rightarrow aBA' \mid \epsilon$	First (A') ={a, ∈ }	Follow (A')= {$}
$B \rightarrow CyB'$	First (B) ={c,y, ∈ }	Follow (B)= {a,$}
$B' \rightarrow CbB' \mid \epsilon$	First (B') ={c,b, ∈ }	Follow (B') ={a,$}
$C \rightarrow C'$	First (C) ={c, ∈ }	Follow (C)= {b, y}
$C' \rightarrow cC' \mid \epsilon$	First (C') = {c,∈ }	Follow (C')= {b,y}

The parsing table :

	a	b	c	x	y	$
A				$A \rightarrow xA'$		
A'	$A' \rightarrow aBA'$					$A' \rightarrow \epsilon$
B	$B \rightarrow CyB'$		$B \rightarrow CyB'$		$B \rightarrow CyB'$	$B \rightarrow CyB'$
B'	$B' \rightarrow CbB'$ $B' \rightarrow \epsilon$	$B' \rightarrow CbB'$	$B' \rightarrow CbB'$			$B' \rightarrow CbB'$ $B' \rightarrow \epsilon$
C		$C \rightarrow C'$	$C \rightarrow C'$		$C \rightarrow C'$	
C'		$C' \rightarrow cC'$ $C' \rightarrow \epsilon$	$C' \rightarrow cC'$		$C' \rightarrow cC'$ $C' \rightarrow \epsilon$	

As there are multiple entries in each column the given grammar is not LL(I).

Q. 23 Construct an LALR (1) parsing table for the following grammar :

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow d.$

Ans. : Augmented grammar is :

0 $S' \rightarrow S$

1 $S \rightarrow Aa$

2 $S \rightarrow bAc$

3 $S \rightarrow dc$

4 $S \rightarrow bda$

5 $A \rightarrow d$

I_0 : $S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot Aa, \$$

$S \rightarrow \cdot bAc, \$$

$S \rightarrow \cdot dc, \$$

$S \rightarrow \cdot bda, \$$

$A \rightarrow \cdot d, a$

I_1 : goto $(I_0, S) : S' \rightarrow S \cdot, \$$

I_2 : goto $(I_0, A) : S \rightarrow A \cdot a, \$$

I_3 : goto (I_0, b)

$S \rightarrow b \cdot Ac, \$$

$S \rightarrow b \cdot da, \$$

$A \rightarrow \cdot d, c$

I_4 : goto (I_0, d)

$S \rightarrow d \cdot c, \$$

$A \rightarrow d \cdot, a$

I_5 : goto $(I_2, a) : S \rightarrow Aa \cdot, \$$

I_6 : goto $(I_3, A) : S \rightarrow bA \cdot c, \$$

I_7 : goto (I_3, d)

$S \rightarrow bd \cdot a, \$$

$A \rightarrow d \cdot, c$

I_8 : goto $(I_4, c) : S \rightarrow dc \cdot, \$$

I_9 : goto $(I_6, c) : S \rightarrow bAc \cdot, \$$

I_{10} : goto $(I_7, a) : S \rightarrow bda \cdot, \$$

LALR (1) parsing table, there is no canonical collection that have identical LR (1) items.

Parsing Table for the grammar :

	Action					Goto	
	a	b	c	d	$	S	A
0		S_3		S_4		1	2
1					Accept		
2	S_5						
3				S_7		6	
4	R_5		S_8				
5					R_1		
6			S_9				
7	S_{10}		R_5				
8					R_3		
9					R_2		
10					R_4		

Q. 27 Write SLR parsing table for : $S \to T$, $T \to CC$, $C \to cC$, $C \to d$.

Ans. :

I_0 : $S \to \cdot T$

$T \to \cdot CC$

$C \to \cdot cC$

$C \to \cdot d$

I_1 : $S \to T \cdot$

I_2 : $T \to C \cdot C$

$C \to \cdot cC$

$C \to \cdot d$

I_3 : $C \to c \cdot C$

$C \to \cdot cC$

$C \to \cdot d$

I_4 : $C \to d \cdot$

I_5 : $T \to CC \cdot$

I_6 : $C \to cC \cdot$

$C \to C \cdot C, C \to \cdot cC, C \to \cdot d$

Parsing Table

	Action			go to		
	c	d	$	S	T	C
I_0	S_3	S_4			1	2
I_1			Accept			
I_2	S_3	S_4				5
I_3		S_4				6
I_4		r_4				
I_5			r_2			
I_6	S_3	S_4	r_3			5

Q. 3(b) Construct LL(1) Parsing table for the following grammar. Also show moves made by input string : abba. **(7 Marks)**

$S \rightarrow aBa$

$B \rightarrow bB \mid \varepsilon$

Ans. : Consider a LL (1) Parsing Table for the given grammar

$S \rightarrow aBa$

$B \rightarrow bB \mid \varepsilon$

Parsing table :

	A	B	$
S	$S \rightarrow aBa$		
B	$B \rightarrow \varepsilon$	$B \rightarrow bB$	

Sequence of moves made by the parser for input "abba".

Stack	Input	Output
$S	abba$	$S \rightarrow aBa$
$aBa	abba$	
$aB	bba$	$B \rightarrow bB$
$aBb	bba$	
$aB	ba$	$B \rightarrow bB$
$aBb	ba$	
$aB	a$	$B \rightarrow \varepsilon$
$a	a$	
$	$	Accept, successful completion

Outputs : $S \rightarrow aBa \quad B \rightarrow bB \quad B \rightarrow bB \quad B \rightarrow \varepsilon$

Derivation (left-most) : $S \Rightarrow aBa \Rightarrow abBa \Rightarrow abbBa \Rightarrow abba$

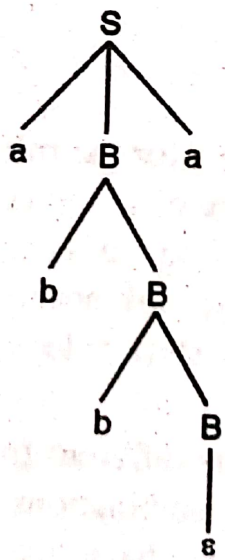

Fig. 1-Q. 3(b) : Parse tree for string "abba"

Q. 2(b)(OR) Construct CLR parsing table for following grammar.

S -> aSA | €

A -> bS | c

(7 Marks)

Ans. :

Step 1 : Augument grammer

$R_0 : S^1 \rightarrow S$

$R_1 : S \rightarrow aSA$

$R_2 : S \rightarrow \in$

$R_3 : A \rightarrow bS$

$R_4 : A \rightarrow C$

Closure :

I_0

$S^1 \rightarrow \cdot S$	$
$S \rightarrow \cdot aSA$	$
$S \rightarrow \cdot$	$

I_1 : goto (I_0, S)

$S^1 \rightarrow S \cdot$	$ R_0

I_2 : goto (I_0, a)

$S \rightarrow a \cdot SA$	b, c
$S \rightarrow .aSA$	b, c
$S \rightarrow \cdot$	b, c

I_3 : goto (I_2, S)

$S \rightarrow aS \cdot A$	$
$A \rightarrow \cdot bs$	$
$A \rightarrow \cdot c$	$

$I_2 \approx$: goto (I_2, a)

$S \rightarrow a \cdot SA$	b, c
$S \rightarrow \cdot aSA$	b, c
$S \rightarrow \cdot$	b, c

I_4 : goto (I_3, A)

$S \rightarrow aSA \cdot$	$	R_1

I_5 : goto (I_3, b)

$A \rightarrow b \cdot s$	$
$S \rightarrow \cdot aSA$	$
$S \rightarrow \cdot$	$

I_6 : goto (I_3, C)

$A \rightarrow C \cdot$	$	R_4

I_7 : goto (I_5, S)

$A \rightarrow bS \cdot$	$	R_3

$I_2 \approx$ goto (I_5, a)

$S \rightarrow a \cdot SA$	b, c
$S \rightarrow \cdot aSA$	b, c
$S \rightarrow \cdot$	b, c

	a	b	c	$	S	A
0	S_2				1	
1				Rol accept		
2	S_2				3	
3		S_5	S_6			4
4				R_1		
5	S_2				7	
6				R_4		
7				R_3		

Given grammer is CLR because in CLR table each cell contain single entry.