S.T.U.S. Mandal's

Sangola Mahavidyalaya, Sangola

Lab Work of compiler construction

	Luo Work of compiler construction
Assignment	Name: Shinde Proja Bapuras Class: Batch: A2 Roll No.: 418/
No.: 1	Name of the Programme - Assignment No 1
Programme	
No.: L	(0.1) Convert following Grammer to LL Grammer
Date :	A -1 a A A -1 E B -3 bbB
	$\begin{array}{c} \mathcal{B} \to \epsilon \\ C \to \mathcal{B} \mathcal{A} \end{array}$
	In this given grammer their is no any left recurssion so we don't have to need to convert it into right recursion grammer.
	for converting grammer into U grammer into the grammer in
	Rule to calculate First & Follow as Follows,
	() Rule for fIRST: ;) Rule No 1:- If terminal symbol a then
	FIRST(D) = {a}
Remarks	ii) Rule No 2: - If their is rule x producing of then FIRST (4) = E
	Then FIRST (A) = FIRST (Yi) U FIRST (Yz) UFIRST (Y)
Signature	(2) Rule for FOLIOW: i) Rule No 1: for start symbol 5 place \$ in the FOLIO(S) ii) Rule No 2: If their production A ii) Rule No 2: If their production A without E
4 7 7 7 7 3	in the FOLLO(S) ii) Rule NO 2: If their production A iii) Rule NO 2: If their production A without E

FOLLOW(B) = { FIRST (B) - E}

FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

FIRST (S) = \(\alpha \, \beta \, \c) \\
FIRST (B) = \(\alpha \, \c) \\
FIRST (B) = \(\beta \, \c) \\
FIRST (C) = \(\beta \, \c) \\
FOLLIOW(C) = \(\delta \, \c) \\
FOLLIOW(C) = \(\delta \, \d

These are FIRST & FOLLOW

No we have to draw a table for ll grammer, as follows.

Ī		a	6	<u> </u>	t
	S	S-> AB.C	S-) AB((S-) AB.C	9
	Α	$A \rightarrow a A$	A → ∈	$A \rightarrow \in$	A → €
	B	4	B→66B1c	3-7881c B-7881c	
	<i>/</i>		C → BA	c→ BA	

(2) Build LL(1) parse table for following Grammer and find out LL(1) or not.

1) 5 → A A → aa A A → b

To build LICID parse Table first we have to Calculate FIRST & POLLOW

Rule for finding FIRST & FOLLOW

- 1 Rules for FIRST
 - DRUIP No. 1: If terminal Symbol a then FIRST (a) = 2 ay
 - ii) Rule No. 2: If their is rule or producing of then
 FIRST (2) = E
 - Then FIRST (A) = PIRST (71) V FIRST (Y2) & --- FURST (Y)

kules for Follow

0

- 1) Rule I; for the start symbol s place \$ in the FOLLOWES)
- 2) Rule 2: If their production A producing &BB everything in FIRST (B) without & is to be placed in FOLLOW(B)
- 3) Rule No 3: If their is production A producing &BB or A producing &B & FIRST (B) = E then follow(A) = POLLOW(B) OY FOLLOW(B) = FOLLOW:(A)

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

LL(1) Parse Table as follows.

A-aaAlb s-aaAlb

Rules for checking given grammer is (LC1) or not.

· A grammer without E is LLLD if · for every production of the A-1/1/2/43/--Mn, the set First (M,), First (M2), First (M3), --- first (Mn) are mutually disjoint.

· FIRST(4,) n FIRST (42) n FIRST (43) n --- n FIRST (4, D=0

· A grammer with E is LLUD if · for every production of the A-ale

· FIRST (&) n FOLLOW (A) = \$ S -> A

FIRST (A) = { a, b} but this production is single so their is no any chance to get common entires. FIRST (OOA) = {a} same as above

FIRST (b) = {b} same as above.

so in this example, we can see that their is no any common entries i.e. it have of therefore given grammer is U(1) grammer.

```
2) F -> E + T
    ナーカナメド
     F -) (E)
     F-) a
 -> E-> E+T In above grammers E-> E+7 & T-> T+;
     ore left recurssion
      First we have to remove left recurssion a
      E \rightarrow E + T
      T - TIF
      F -> (E)
      F \rightarrow a
     After eliminating left recursion grammor is
     A -> +TA | +T
     B→ * FBI * F
    To build [[[]) parse table first we have to
 colculate FIRST and FOLLOW
    FIRST (E) = { +}
                             FOLLOW(E)={$,)}
    FIRST (A) = {+}
                              FOLLOW(A) = { $, )}
    FJRST (T) = {x}
                              FOL(OW(T) = { + , $ , )}
    FIRST (B) = {*}
                               FOL(OW(B) = {+,$,)}
     FIRST (F) = { (, a}
                               FOLLOW(F)={+,$,)}
Explanation =>
   E - A
    for this production we will go for first ( ) that is
  First (E)
  A -+ + TA / +T
      for this production first(A) = {+}
  B -> * FB1 * F
                               7 7 B
      FIRST (B) = { $ }
                               FIRST (T) = PIRST (B) = { d}
   F \rightarrow (E) | a \qquad FJRST(F) = \langle (, a \rangle
```

E→ A, A→+TA | +T, T→B, B→*FB|*F, F→(E), F→A
for this production we will go for FOLLOW(€)
will be {\$,}}

In this follow we have to add & because this is starting symbol.

* LL	(1)	parsing.	Table a	s follows	,		
		.+	4	()	9	P
	E	$E \rightarrow A$,
	A	$\begin{array}{c} A \rightarrow +TA \\ A \rightarrow +T \end{array}$					
		/	T-)B			F 7 1 12	
	B		B→* FB B→* F				-1-1
	F.		ü	f → (E) F → a	- a : 1	F + CE) F -> a	

Pules for Checking given grammer is LLC1) or not

i) A grammer without & is LLC1) if

ofor every production of A -> ×1/2/3/---

the set FIRST(2,), FIRST(2) --- Pirst(2n) are

mutually disjoint.

First(<1) n first(<2) - - - n first(<n) = \$

2) A grammer with & is LL(1) if

for every production of A + a | &

FIRST (2) n FOLLOW(A) = \$

For the production E A, T AB, F=>-(E their is no need to check rules because that are already single production.

SO, for A → + TA (+T)
FIRST (~;) N FIRST (~2)

FIRST (+TA) N FIRST(+T)

{+} Their is no \$ that's why we can say that this grammer is not LL(1) grammer.

also for B → * FB | * F FIRST (Y,) D FIRST (Y2)

FIRST (*FB) O FIRST (*F) (本)

This is not & that's why we can say that this given grammer is not LL(1) grammer.

- 3) 5 A A 5 -> 0 A-a
- To build LL(1) parsing table first we have to calculate FIRST & FOLLOW

Rules for finding FIRST & FOLLOW

- O Rules for FIRST : i) Rule No 1: If terminal symbol a then FIRST(1)=(0)
 - is Rule No. 2: If their is mult or producing of then FIRST (72) = E
 - iii) Rule No 3: For the rule o is producing the YI YZ , --- YN Then FIRST (A) =

FIRST(Y,) U FJRST (Y2) U --- FJRST (YN)

- @ Rules for FOLLOW:
- DRULE NO 2: For the start symbol 5 place \$ in FOLLOW(S)
- 2) Rule No 2: If their production A producing \$ in ' & BB everything in FIRST (B) without & is to be placed in FOLLOW (B)
 - 3) Rule No 3: If their is production A producing &BB or A producing ZB & FIRST (B)= + then FOLLOW(A) = FOLLOW(B) Or FOLLOW(B) = FOLLOW(A)

LL(1) parsing Table

. a

S = JaAla

 $A \rightarrow a$

Rules for checking given grammer is LL(1) or not

• A grammer without & is LL(1) if

• for every production of A + 4, 1 / 31
<n the set FIRST (4,7), FIRST (2,7), FIRST (2,3) ---
FIRST (2,7)

· FIRST (<1) 1 FIRST (<2) D FIRST (<3) D - -- FIRST (<1)=

• A grammer with t is U(1) if • for every production of the A-Tale • FIRST(1) n POLLOW(A) = \$

S-JaAla

FIRST (41) N FIRST (42)
FIRST (AA) N FIRST (A)

{a} ! = {\$\$}

so we can see that this is not \$
so given grammer is not U(1) grammer.

4) $S \rightarrow AS$ $S \rightarrow Q$ $A \rightarrow SA$ $A \rightarrow B$

To build LL(1) passing table first we have to calculate

() Rules for Finding FIRST & FOLLOW

i) Rule No I: If terminal symbol a then FIRST (a) = (0)

ii) Rule No 2: If their is rule or producing & then

FIRST (71) = 6

Pii) Rule No I i If terminal symbol a then FIRST (a) = (a)

- ii) Rule No 2: If their is rule or producing & then FIRST (+)= +
 - iii) Rule. No 3; for the rule a is producing the

Y1 , Y2 - -- YN THEN FIRST (A) = FIRST (Y1) U_--- f&RST(YN)

@Rules for Follow

- DRULE No 2: For the start symbol 5 place & in
- 2) Rule No 2: If their production A producing XBB everything in FIRST(B) without t is to be placed
- 3) Rule No 3: If their is production A producing LBB or A producing 2B & FIRST (B) = E then FOLLOWIA) = FOLLOW(B) Or FOLLOW(B) = FOLLOW(A)

S-1 AS

5 - q

A 7 SA

A - b

FIRST (5) = FIRST (A) = {b, a}

FIRST $(A) = FIRST(5) = \{b, a\}$

FOLLOW(5) = {\$}, 6, a}

FOLLOW (A) = { 'b, a}

LIC	1) Parsi	ing Tab	le
-	q	6	\$
5	5-1 AS	S-AS	
	3-19	S → q	
A	A → SA	A->sA	
	$A \rightarrow b$	A -> b	
0 . 1			

Rules for checking given grammer is LL(1) or not 1) A grammer without & is U(1) if for every production of A -> <1/2/3/-- In

```
se FIRST (LI), FIRST (K2) - -- FIRST (Kn) are
  The
       routually disjoint.
 FIRST (41) \cap FIRST (42) - - - \cap FIRST (40) = \varphi
   2) A grammer with 6 is LLL1) iR
         · for every production of A-lake
          · FIRST (4) N FOLLOW(A) = $
   For the production 5-1 As | a
       FIRST (4,) n FIRST (42)
       FIRST (AS) N PIRST (a)
            2 a } ! = 2 p }
      So we can see that given granomer
    is not (111) grammer.
5) 5→ BAC
 A - aA
  A-1a
  B-AB
  BABB
  B-d
+ To build LL(1) passing table first we have to calculate
  FIRST & FOLLOW
 Brules for finding PIRST
   i) Rule No 1: If terminal symbol a then FIRST (4) = 297
   ii) Rule NO 2: If their is rule or producing & then
     FIRST (x) = E
@ Rules for finding FOLLOW
   i) Rule 1: for the start symbol s place & in Follows
   ii) Rule 2: If their production A producing XBB everything
     in FIRST(B) - Vithout & is to be placed in following
   111) Rule No 3: If their is production A producing KBB or
   A producing XB & FIRST (B) = E then FOLLOW (A) = FOLLOW (B)
    OY FOLLOW(B) = FOLLOW(A)
   FIRST(S) = { b, d, a}
```

FIRST(A) = { a}

ull	parsing	Table			
	a	b	C	4	4
5.	S-> BAC	S -BAC		S-BAC	7
Ą	A -> a A	- /			
В	B → AB B → D B → D	B-1 AB B-1 6B		B → BB B → B	1

Rules for checking given grammer is LL(1) or

1) A grammer without e is LL(1) if for every production of A-1/2/3/--- The set FIRST(x1), FIRST(x2) --- FIRST(x1) are mutually disjoint

FIRST (4,) N FIRST (42) --- N FIRST (4h) = \$

- 2) A grammer with + is LL(1) if
 - · for every production of A-1916
 - · FIRST (Y) D FOLLOW(A) = \$
-) For the production A-1 aA 1 4

ii) For the production B -> 6BIABID

FIRST (KB) N FIRST (Y2) N FIRST (Y3)

FIRST (BB) N FIRST (AB) N FIRST (d)

{b} N {a} n{d}

= {b}

we can see that above grammer content common entiter so above grammer is not L(1) grammer.

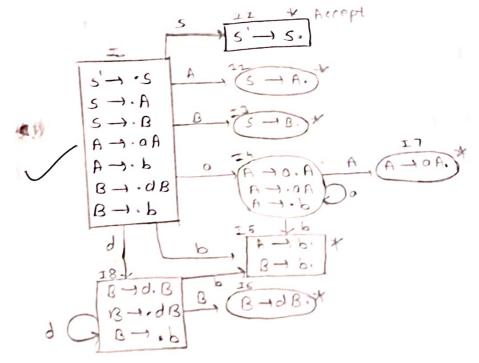
and find out (112) or not.

B-b

solution=Orind augmented grammer

The augmented grammer of given grammer is,

step 2: Find LR(0) collection of items
Below is fig showing the LR(0) collection of
items.



FOLLOW(B) = { \$} FOLLOW(B) = { \$} FIRST(S) = { a, b, d, b}

FIRST(A) = { a, b}

FIRST(B) = { d, b}

	a	Ь	d	\$	5	A	B
10	54	55	5 8		1	2	3
11				Accept			
12				R1			
J 3				R2_			
I 4	54	Ss				7	
I 5				R4 R6			
16				R5			
J7				F3			
81		55	58				6

in SLR(1) parsing Table we have to write reduced production in only follow of left hand side of production.

Rules for checking given grammer is LL(1) or not *A grammer without & is LL(1) if · for every production of A-11/2/31--- 2n

the se FIRST (x1), FIRST (2), FIRST (x3) ---

• FIRST (<1) D FIRST (<12) D FIRST (<3) D -- FIRST (<3) = grammer with e is will if

· for every production of the A-ale · FIRST (Z) n FOLLOW(A) = \$

5 - A I B <1 <2

FIRST (Y,) D FIRST (Y2)

FIRST (A) N FIRST (B)

(b, a) n {d, b}

16) != {p}

as given rules does follows for this production so, we don't need to check for another production I we can say that given grammer is not LL(1) grammer

solution = In above grammer E-E+T & T-T+ F are left recurssion.

First we have to remove left recurssion

For eliminating left recursion from above grammer we have to declare new variable A & B.

After eliminating left recursion grammer is

to Build LL(1) parse table first we have to colculate FIRST and Follow

FIRST ORUles for finding FIRST

- i) RULE NO 1: If terminal symbol a then FIRST(a) = 29}
- (i) Rule NO2: If their is rule & producing of then

 FIRST (2) = 6
- Then FIRST(A) = FIRST(7) UFIRST(72) U --- FIRST (7N)

@ Rules for follow:

- FOLLOW(5)
- 2) Rule No 2: If their production A producting LBB everything in FIRST (B) without & is to be placed in FOLLOW (B)
 3) Rule No 3: If their is production A producing

```
ABB or A producing KB & FIRST(B) = E then FOLLOW(A) =
          FOLLOW(B) Or FOLLOW(B) = FOLLOW(A)
                                 FOLLOW(E)={$ 1}}
        FIRST (E) = {+}
                                 FOLLOW(A) = {$,7}
        FIRST (A) = {+}
        FIRST (7) = { * }
                                 FOLLOW(T) = {+ , $, )}
                                 FOLLOW(B) = {+, $, >}
        7 I RST (F) = { (, a }
        FIRST (B) = < * >
                                 FOLLOW(F) = ? +,$, ?}
    Step 1 find augmented grammer
         The augmented grammer of given grammer is,
         ± → · E [ oth production]
         A - +TE
       A -. + T
                    [3rd]
        I -> .B
                    [4th7
        B - . * FB [sth]
        B-1. *F [6th]
        F > . (E) [7th]
        P - 1. a [sth]
Slep @ Find LRCO) collection of items below is
     the troop collection of items
                                                   fig showing
                      arrept
   E' - · E
                                         F - ( E. )
```

		9	()	*	1 +	\$	E	Α	· B	1	F
	10					5 3		1	2			
	12						Accept	+				
	12			R1			RI					
	13				56					5	4	
The same of the sa	14					59		8	2			
	IS			R4		R4	R4					
	16	SII	57									10
	Jη							12	2_			
	18			R2			R2					
	19				56						13,4	
	Ilo				56					14		
	<u>III</u>			R8		R8	R8					
	I 12			515	-							
	I 13			R3			R3					
	I 14			RS		RS	RS					
	I IS			R7		R7	R7					

Rules for checking given grammer is LCL) or

of grammer without + is (1(1) if

·for every production of A-1<1<2/>
the set FIRST(<1), FIRST(<2), FIRST(<3)FIRST(<1)

*FIRST (4,) N FIRST (42) N FIRST (43) N -- FIRST (4n) = d

· A grammer with f is LL(1) if · for every prod, of A -1 a 16 · first(x) n follow(A) = \$

 $A \rightarrow + TA \{+T\}$ first(z) n = first(z)first(+TA) n = first(+T) so given grammer is $\{+\} = \{0\}$ not LL(1)

3) $S \rightarrow AA$ $S \rightarrow AA$ $A \rightarrow b$

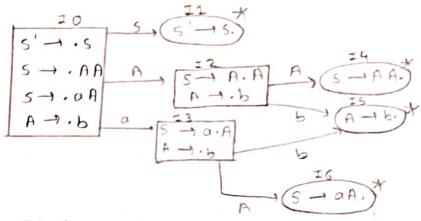
solution step () find augmented grammer . The augmented grammer of given grammer is,

5-1.5 coth production]

5-1. aA [2nd production]

A -1. b [3rd production]

Step @ Find LP(0) collection of items below is fig showing the LP(0) collection of items



First we have to findow FIRST & FOLLOW

- i) Rule No I: If terminal symbol a then FIRSTCO) = {a7
- ii) Rule No 2: If their is rule or producing & then

 FIRST (x) = E
- Then FIRST (A) = FIRST (Y) U FIRST (Y2) U--- FIRST (YN)

ORules for follow;

i) Rule No 2: for the start Symbol 5 place of in FOLLOWS)

ii) Rule No 2: If their production A producing XBB everything
in FIRST (B) without & is to be placed in FOLLOW(B)

iii) Rule No 3: If their is production A producing ZBB or A

producing XB & FIRST (B) = E then POLLOW(A) = FOLLOW(B)

or FOLLOW(B) = FOLLOW(A)

 $FIRST(S) = \{4, b\}$ FOLLOW(S) = $\{4, b\}$ $FIRST(A) = \{b\}$ FOLLOW(A) = $\{4, b\}$

١						
		a	Ь		S	Λ
	TO	53			2_	2,
	11			Uccebp	1 A	1477
	12		S ₅	1		4
	13		Ss		1,0	6
	14			R2	17	
	I 5	7	R3	R3	() ()	The set p
	16			R2		11 11
Ť			-			

Rules for checking given grammer is LL(1) or not

· A grammer without e is LL(1) if

· for every production of A-x1/2/31--- < the set FIRST (4,1), FIRST (42), PIRST (43) --- FIRST(4n)

* FIRST (~1) · n FIRST (~2) n FIRST (~3) n - - - FIRST (~n) = 0

* A grammer with & is LLCI) if

· for every production of the A-19/E

· FIRST (2) n FOLLOW(A) = 0

FIRST (41) N FIRST (42)

FIRST (AN) o FIRST (AA) the respective 1 b) n (a)

given grammer is LLCI)

4)
$$S \rightarrow A$$

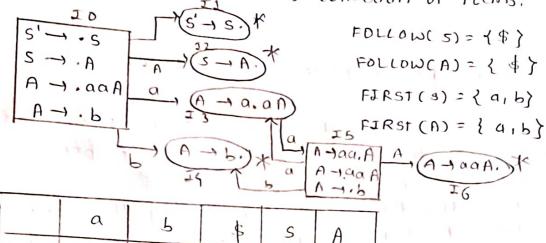
$$A \rightarrow aaA$$

A -> b

solution: step 1 : Find augmented grammer

The augmented grammer of given grammer is, production

Showing the LR(0) collection of items below is fig



		11	1		-1-	
	-	a	5	\$	S	А
	Io	53	54		1	2
	II	8	1	accept) 1
	I2_		1	RI I	1 1	
	13	55	у.			-
	IL		,	R3	1	
	I 5	53	54	RZ		6
1	I6	113		R2		3

Rules for checking given grammer is 1111) or not

· A grammer without & is LC(1) if

the set FIRST (4,), FIRST (42), FIRST (43) ---- FIRST (4n)

· FIRST (41) D FIRST (42) D - - - FIRST (40) = 6

OA grammer with & is well if

- · for every production of n-ale
- · FIRST (Z) N FOLLOW(N) = d

n - aaAlb

FIRST (<1) D FIRST (<2)

FIRST (aaA) N FIRST (b)

10>

given grammer is LL(1)

Party Ch

5) 5 - Ab A -1 aA A -1 ab A WE

For calculating FIRST & FOLLOW following Rules.

(1) FIRST :-DRUIE No 1: If terminal symbol a then FIRST (9) - Lay ii) Rule No 2: If their is rule or producing & then FIRST(X) = (

iii) Rule No 3: For the rule o is producing the Mi, Yz--YN Then FIRST (A) = FIRST (Y1) UFIRST (Y2) U· - FIRST (YN)

@ FOLLOW

i) Rule No 1; for the start symbols place \$ in.

ii) Rule No 2; It their production A producing XBB everything in FIRST (B) without & is to be placed in FOLLOWIBJ

iii) Rule No 3: It their is production A producing LBB or A producing KB & FIRST (B) = E then FOLIOW(A) = FOLIOW(B) OF FOLIOW(B) = FOLIOW(A)

FIRST(S) = { a, b, E} FOLLOW(s) = (\$) FIRST (A) = { a, e} FOLLOW(A) = { b}.

Step O find augmented grammen The augmented grammer of given grammer is

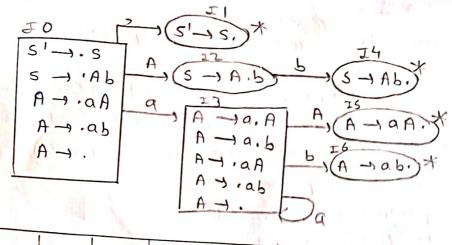
S'T'S Eoth production] 5-1.Ab [161]

A -1. aA [2nd]

A-1.ab [3rd7]

A -1.6 [uth]

step @ Find LR(0) collection of items below is fig showing the LRCO) collection of Items.



	1					
		4	Ь	\$	S	A
	To	53	- Inches		1	2
	51	- 4	Ä	Accept		
4	JZ		54		1)	1 1
	F 3	53	56		,	5
d d	I4			RI		
1	II		R2			
	I6		R3			
	•					

The non-terminals of this grammer are (a,b)

The non-terminals of this grammer are (S,A)

\$ is by defoult a non-terminal that takes accepting state.

- IO gives A in I so I is added to A column to Dw
 - · similary 5 is written in A column & 4 now.
 - · Io gives a in I3,50 53 (Shift 3) is added to a column & o row.

Rules for checking given grammer is LL(1) or not

of grammer without e is LL(1) if

ofor every production of A -> 1 /2 /3 -- 20

The set FJRST (21), PJRST (22), PJRST (23) --- FJRST (2

oFJRST (21) n FJRST (22) n FJRST (23) n --- FJRST (24) = 0

· A grammer with & is LLLI) if

· for every production of A-1ale

· FIRST (X) n FOLLOW (A) = \$

 $A \rightarrow aA \mid ab \mid \epsilon$ $\langle 1 \mid \langle 2 \mid \langle 3 \rangle$ FIRST ($\langle 1 \rangle \mid n \mid FIRST (<math>\langle 2 \rangle \mid n \mid FOLLOW (\langle 3 \rangle)$ $\langle a \rangle \mid n \mid \langle a \rangle \mid n \mid FOLLOW (A \rightarrow \epsilon) = \langle b \rangle$ $\langle a \rangle \mid = \langle a \rangle$

given grammer is not LL(1) grammer,

27/02/24