S.T.U.S. Mandal's

Sangola Mahavidyalaya, Sangola

Lab Work of compiler construction

Assignment No.: 1	Name : Shinde Pooja Bapurao Class : BSc ECS-III Batch : A2 Roll No.: 4181
Programme No.: 1	Name of the Programme - Assignment No 1
Date :	

Q.1) Convert following Grammer to LL Grammer.

I) $S \rightarrow ABcC$
$A \rightarrow aA$
$A \rightarrow \epsilon$
$B \rightarrow bbB$
$B \rightarrow \epsilon$
$C \rightarrow BA$

→ 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 LL grammer first we have to calculate first & follow.

Rule to calculate First & Follow as follows,

① Rule for FIRST :

i) Rule No 1 :- If terminal symbol a then FIRST(a) = {a}

ii) Rule No 2 :- If their is rule x producing α then FIRST(α) = ε

iii) for rule a is producing the $Y_1, Y_2 --- Y_k$ Then FIRST(A) = FIRST(Y_1) ∪ FIRST(Y_2) --- ∪ FIRST(Y_N)

② Rule for FOLLOW:

i) Rule No 1 : for start symbol s place $ in the FOLLO(S)

ii) Rule No 2 : If their production A producing αBβ everything in FIRST(β) without ε

is LL(1) grammer.

Remarks

Signature

is to be placed in FOLLOW(B)

FOLLOW(B) = { FIRST(β) − ε }

iii) Rule No 3 : If their is production A producing αBβ or A producing αB & FIRST(β) = ε then

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

FIRST(S) = {a, b, c}　　FOLLOW(S) = {$}

FIRST(A) = {a, ε}　　FOLLOW(A) = {b, c, $}

FIRST(B) = {b, c}　　FOLLOW(B) = {c, b, a}

FIRST(C) = {b, c}　　FOLLOW(C) = {$}

These are FIRST & FOLLOW

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

	a	b	c	$
S	S → ABcC	S → ABcC	S → ABcC	
A	A → aA	A → ε	A → ε	A → ε
B		B → bbB/c	B → bbB/c	
		C → BA	C → BA	

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

1) S → A
A → aaA
A → b

→ To build LL(1) parse Table first we have to Calculate FIRST & FOLLOW

Rule for finding FIRST & FOLLOW

① Rules for FIRST

i) Rule No. 1: If terminal symbol a then FIRST(a) = {a}

ii) Rule No. 2: If their is rule x producing α then

FIRST(x) = E

iii) Rule No 3: For the rule a is producing the $Y_1, Y_2, \ldots$

Then FIRST(A) = FIRST(Y_1) ∪ FIRST(Y_2) ∪ ... FIRST(Y

Rules for FOLLOW

1) Rule 1 : for the start symbol S place $ in the FOLLOW(S)
2) Rule 2 : If their production A producing $\alpha B \beta$ everything in FIRST (β) without ϵ is to be placed in FOLLOW(B)
3) Rule No 3 : If their is production A producing $\alpha B \beta$ or A producing αB & FIRST (β) = ϵ then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

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

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

LL(1) Parse Table as follows.

	a	b	$
S	S → A	S → A	
A	A → aaA|b	S → aaA|b	

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

- A grammer without ϵ is LL(1) if
  - for every production of the $A \to \alpha_1 | \alpha_2 | \alpha_3 | \cdots \alpha_n$, the set First($\alpha_1$), First($\alpha_2$), First($\alpha_3$), ---First($\alpha_n$) are mutually disjoint.
  - FIRST(α_1) ∩ FIRST(α_2) ∩ FIRST(α_3) ∩ --- ∩ FIRST(α_n) = ∅

- A grammer with ϵ is LL(1) if
  - for every production of the $A \to \alpha | \epsilon$
  - FIRST (α) ∩ FOLLOW(A) = ∅

S → A

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

A → aaA.

FIRST (aaA) = {a} same as above

A → b

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

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

2) $E \rightarrow E+T$

$T \rightarrow T * F$

$F \rightarrow (E)$

$F \rightarrow a$

→ $E \rightarrow E+T$ In above grammers $E \rightarrow E+T$ & $T \rightarrow T*F$ are left recurssion

First we have to remove left recurssion

$E \rightarrow E \underbrace{\boxed{+T}}_{A}$

$T \rightarrow T \underbrace{\boxed{*F}}_{B}$

$F \rightarrow (E)$

$F \rightarrow a$

After eliminating left recursion grammar is,

$E \rightarrow A$

$A \rightarrow +TA \mid +T$

$T \rightarrow B$

$B \rightarrow *FB \mid *F$

$F \rightarrow (E)$

$F \rightarrow a$

To build LL(1) parse Table first we have to calculate FIRST and FOLLOW

FIRST (E) = { + }

FIRST (A) = { + }

FIRST (T) = { * }

FIRST (B) = { * }

FIRST (F) = { (, a }

FOLLOW(E) = { \$,) }

FOLLOW(A) = { \$,) }

FOLLOW(T) = { +, \$,) }

FOLLOW(B) = { +, \$,) }

FOLLOW(F) = { +, \$,) }

Explanation ⇒

$E \rightarrow A$

For this production we will go for first(A) that is first(A)

= { + }

$A \rightarrow +TA \mid +T$

for this production first(A) = { + }

$B \rightarrow *FB \mid *F$

FIRST (B) = { * }

$T \rightarrow B$

FIRST (T) = FIRST (B) = { * }

$F \rightarrow (E) \mid a$ FIRST (F) = { (, a }

$E \rightarrow A$, $A \rightarrow +TA \mid +T$, $T \rightarrow B$, $B \rightarrow *FB \mid *F$, $F \rightarrow (E)$, $F \rightarrow a$

for this production we will go for FOLLOW(E) will be {$,)}

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

* LL(1) parsing Table as follows.

	+	*	(	)	a	$
E	$E \rightarrow A$					
A	$A \rightarrow +TA$ $A \rightarrow +T$					
T		$T \rightarrow B$				
B		$B \rightarrow *FB$ $B \rightarrow *F$				
F			$F \rightarrow (E)$ $F \rightarrow a$		$F \rightarrow (E)$ $F \rightarrow a$	

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

1) A grammer without ϵ is LL(1) if
   - for every production of $A \rightarrow \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \cdots \alpha_n$ the set FIRST(α_1), FIRST(α_2) --- First(α_n) are mutually disjoint.

$\text{First}(\alpha_1) \cap \text{First}(\alpha_2) \cdots \cap \text{First}(\alpha_n) = \phi$

2) A grammer with ϵ is LL(1) if
   - for every production of $A \rightarrow a \mid \epsilon$
   - FIRST(α) $\cap$ FOLLOW(A) $= \phi$

For the production $E \rightarrow A$, $T \rightarrow B$, ~~$F \rightarrow E$~~. their is no need to check rules because that are already single production.

so, for $A \rightarrow \underset{\alpha_1}{+TA} \mid \underset{\alpha_2}{+T}$

FIRST(α_1) $\cap$ FIRST(α_2)

FIRST(+TA) $\cap$ FIRST(+T)

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

also for $B \rightarrow \underset{\alpha_1}{*FB} \mid \underset{\alpha_2}{*F}$

FIRST(α_1) $\cap$ FIRST(α_2)

FIRST(* F B) $\cap$ FIRST (* F)

{ * }

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

3) $S \rightarrow aA$

$S \rightarrow a$

$A \rightarrow a$

→ To build LL(1) parsing table first we have to calculate FIRST & FOLLOW

Rules for finding FIRST & FOLLOW

① Rules for FIRST :

i) Rule No 1: If terminal symbol a then FIRST(a) = {a}

ii) Rule No. 2 : If their is rule x producing ϵ then FIRST (x) = ϵ

iii) Rule No 3 : For the rule a is producing the $Y_1, Y_2, --- Y_N$ Then FIRST (A) = FIRST(Y_1) $\cup$ FIRST (Y_2) $\cup$ --- FIRST (Y_N)

② Rules for FOLLOW :

1) Rule No 1 : For the start symbol S place \$ in FOLLOW(S)

2) Rule No 2 : If their production A producing \$ in $\alpha B \beta$ everything in FIRST (β) without ϵ is to be placed in FOLLOW (B)

3) Rule No 3 : If their is production A producing $\alpha B \beta$ or A producing αB & FIRST (β) = ϵ then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

FIRST (S) = {a}

FIRST (A) = {a}

FOLLOW(S) = { \$ }

FOLLOW (A) = { \$ }

LL(1) parsing Table

	a
S	$S \to aA \mid a$
A	$A \to a$

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

- A grammer without ϵ is LL(1) if
  - for every production of $A \to \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \ldots \alpha_n$ the set FIRST (α_1), FIRST (α_2), FIRST (α_3) ---- FIRST (α_n)
  - FIRST $(\alpha_1) \cap$ FIRST $(\alpha_2) \cap$ FIRST $(\alpha_3) \cap$ --- FIRST $(\alpha_n) = \phi$

- A grammer with ϵ is LL(1) if
  - for every production of the $A \to \alpha \mid \epsilon$
  - FIRST $(\alpha) \cap$ FOLLOW$(A) = \phi$

$S \to \underset{\alpha_1}{aA} \mid \underset{\alpha_2}{a}$

FIRST $(\alpha_1) \cap$ FIRST(α_2)

FIRST $(aA) \cap$ FIRST (a)

$\{a\} \;\; != \{\phi\}$

So. we can see that this is not ϕ

So given grammer is not LL(1) grammer.

4) $S \to AS$
$S \to a$
$A \to SA$
$A \to b$

To build LL(1) parsing table first we have to calculate FIRST & FOLLOW

① Rules for finding FIRST & FOLLOW

i) Rule No 1 : If terminal symbol a then FIRST $(a) = \{a\}$

ii) Rule No 2 : If their is rule x producing ϵ then FIRST $(x) = \epsilon$

i) Rule No 1 : If terminal symbol a then FIRST (a) = {a}
ii) Rule No 2 : If their is rule x producing ∈ then FIRST (x) = ∈
iii) Rule No 3 : for the rule a is producing the $Y_1, Y_2 --- Y_N$ Then FIRST (A) = FIRST(Y_1) U ---- FIRST(Y_N)

(e) Rules for FOLLOW

1) Rule No 1 : For the start symbol S place $ in FOLLOW (S)
2) Rule No 2 : If their production A producing αBβ everything in FIRST(β) without ∈ is to be placed in FOLLOW(B)
3) Rule No 3 : If their is production A producing αBβ or A producing αB & FIRST (β) = ∈ then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

S → AS
S → a
A → SA
A → b

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

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

LL(1) Parsing Table

	a	b	$
S	S → AS S → a	S → AS S → a	
A	A → SA A → b	A → SA A → b	

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

1) A grammer without ∈ is LL(1) if
• for every production of A → $α_1 | α_2 | α_3 | --- α_n$

The se FIRST (α_1), FIRST (α_2) --- FIRST (α_n) are mutually disjoint.

FIRST (α_1) $\cap$ FIRST (α_2) --- $\cap$ FIRST (α_n) = ϕ

2) A grammer with ϵ is LL(1) if

- For every production of $A \to \alpha \,|\, \epsilon$
- FIRST (α) $\cap$ FOLLOW(A) = ϕ

For the production $S \to \underset{\alpha_1}{AS} \,|\, \underset{\alpha_2}{a}$

FIRST (α_1) $\cap$ FIRST (α_2)

FIRST (AS) $\cap$ FIRST (a)

$\{a\}$ $!= \{\phi\}$

so we can see that given grammer is not LL(1) grammer.

5) $S \to BAc$
$A \to aA$
$A \to a$
$B \to AB$
$B \to bB$
$B \to d$

$\to$ To build LL(1) parsing table first we have to calculate FIRST & FOLLOW

① Rules for Finding FIRST

i) Rule No 1: If terminal symbol a then FIRST (a) = {a}
ii) Rule No 2: If their is rule x producing ϵ then FIRST (x) = ϵ

② Rules for finding FOLLOW

i) Rule 1: for the start symbol S place \$ in FOLLOW(S)
ii) Rule 2: If their production A producing $\alpha B \beta$ everything in FIRST(β) without ϵ is to be placed in FOLLOW(B)
iii) Rule No 3: If their is production A producing $\alpha B \beta$ or A producing αB & FIRST (B) = ϵ then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

FIRST(S) = { b, d, a }
FIRST(A) = { a }

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

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

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

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

LL(1) parsing Table

	a	b	c	d	$
S	$S \to BAC$	$S \to BAC$		$S \to BAC$	
A	$A \to aA$ $A \to a$				
B	$B \to AB$ $B \to bB$ $B \to d$	$B \to AB$ $B \to bB$ $B \to d$		$B \to AB$ $B \to bB$ $B \to d$	

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

1) A grammer without ϵ is LL(1) if for every production of $A \to \alpha_1 | \alpha_2 | \alpha_3 | \dots \alpha_n$ the set $FIRST(\alpha_1)$, $FIRST(\alpha_2)$ --- $FIRST(\alpha_n)$ are mutually disjoint

$FIRST(\alpha_1) \cap FIRST(\alpha_2) \dots \cap FIRST(\alpha_n) = \phi$

2) A grammer with ϵ is LL(1) if
   - for every production of $A \to a | \epsilon$
   - $FIRST(\alpha) \cap FOLLOW(A) = \phi$

i) For the production $A \to \underset{\alpha_1}{aA} \mid \underset{\alpha_2}{a}$

$FIRST(\alpha_1) \cap FIRST(\alpha_2)$

$\{a\} \; != \{\phi\}$

ii) For the production $B \to \underset{\alpha_1}{bB} \mid \underset{\alpha_2}{AB} \mid \underset{\alpha_3}{d}$

$FIRST(bB) \cap FIRST(\alpha_2) \cap FIRST(\alpha_3)$

$FIRST(bB) \cap FIRST(AB) \cap FIRST(d)$

$\{b\} \cap \{a\} \cap \{d\}$

$= \{\phi\}$

We can see that above grammer content common entries so above grammer is not LL(1) grammer.

3) Build SLR(1) parse table for the following grammer and find out LL(1) or not.

I) $S \rightarrow A$

$S \rightarrow B$

$A \rightarrow aA$

$A \rightarrow b$

$B \rightarrow dB$

$B \rightarrow b$

Solution = ① Find augmented grammer

The augmented grammer of given grammer is,

$S' \rightarrow .S$ [0th production]

$S \rightarrow .A$ [1th production]

$S \rightarrow .B$ [2nd production]

$A \rightarrow .aA$ [3rd]

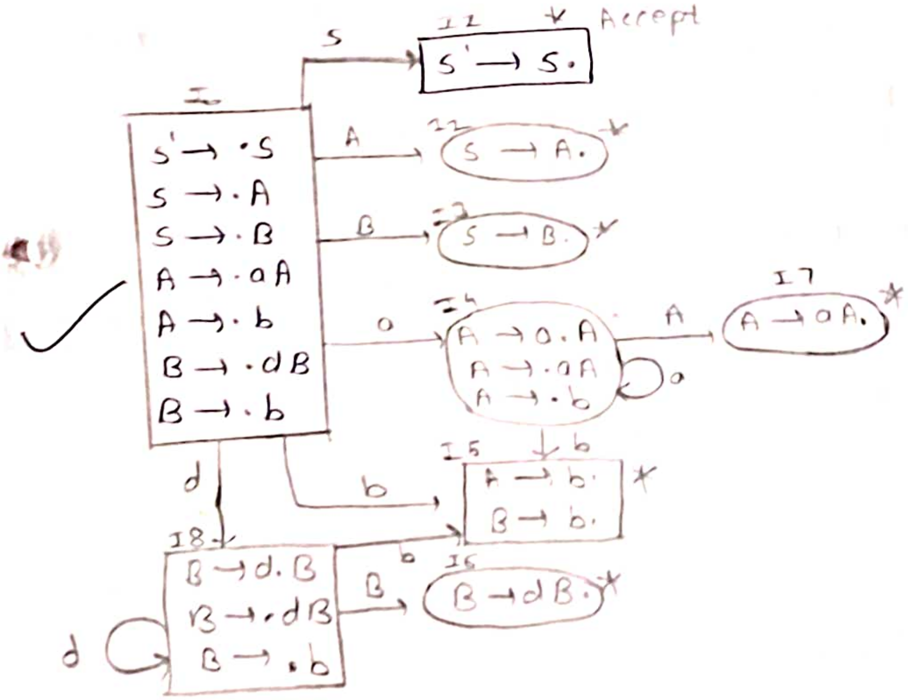
$A \rightarrow .b$ [4th]

$B \rightarrow .dB$ [5th]

$B \rightarrow .b$ [6th]

step ②: Find LR(0) collection of items

Below is fig showing the LR(0) collection of items.



FOLLOW(S) = { \$ }

FOLLOW(A) = { \$ }

FOLLOW(B) = { \$ }

FIRST(S) = { a, b, d, b }

FIRST(A) = { a, b }

FIRST(B) = { d, b }

	a	b	d	$	S	A	B
I0	S_4	S_5	S_8		1	2	3
I1				Accept			
I2				R1			
I3				R2			
I4	S_4	S_5				7	
I5				R4 R6			
I6				R5			
I7				R3			
I8		S_5	S_8				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 ϵ is LL(1) if
  - for every production of $A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 | \ldots \alpha_n$ these FIRST(α_1), FIRST(α_2), FIRST(α_3) ... FIRST(α_n)
  - FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ FIRST(α_3) $\cap \ldots$ FIRST(α_n) $= \phi$
- A grammer with ϵ is LL(1) if
  - for every production of the $A \rightarrow \alpha | \epsilon$
  - FIRST(α) $\cap$ FOLLOW(A) $= \phi$

$S \rightarrow \underset{\alpha_1}{A} \mid \underset{\alpha_2}{B}$

FIRST(α_1) $\cap$ FIRST(α_2)

FIRST(A) $\cap$ FIRST(B)

$\{b, a\} \cap \{d, b\}$

$\{b\} \; != \; \{\phi\}$

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

2) $E \rightarrow E + T$

$T \rightarrow T * F$

$F \rightarrow (E)$

$F \rightarrow a$

Solution ⇒ In above grammer $E \rightarrow E+T$ & $T \rightarrow T * F$ are left recurssion.

First we have to remove left recurssion

$E \rightarrow E \underbrace{\boxed{+T}}_{A}$

$T \rightarrow T \underbrace{\boxed{* F}}_{B}$

$F \rightarrow (E)$

$F \rightarrow a$

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

After eliminating left recursion grammer is

$E \rightarrow A$

$A \rightarrow +TA \mid +T$

$T \rightarrow B$

$B \rightarrow *FB \mid *F$

$F \rightarrow (E)$

$F \rightarrow a$

To Build LL(1) parse table First we have to calculate FIRST and FOLLOW

~~FIRST~~ ① Rules for finding FIRST

i) RULE NO 1: If terminal symbol a then FIRST(a) = {a}

ii) Rule No 2: If their is rule x producing α then FIRST (x) = ϵ

iii) Rule No 3: For the rule a is producing the $Y_1, Y_2 --- Y_N$

Then FIRST(A) = FIRST(Y_1) U FIRST(Y_2) U --- FIRST(Y_N)

② Rules for follow:

1) Rule No 1: For the start symbol s place \$ in FOLLOW(S)

2) Rule No 2: If their production A producing $\alpha B \beta$ everything in FIRST (β) without ϵ is to be placed in FOLLOW (B)

3) Rule No 3: If their is production A producing

$\alpha B\beta$ or A producing αB & FIRST(β) = ε then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

FIRST(E) = {+}
FIRST(A) = {+}
FIRST(T) = {*}
FIRST(F) = {(, a}
FIRST(B) = {*}

FOLLOW(E) = {$,)}
FOLLOW(A) = {$,)}
FOLLOW(T) = {+, $,)}
FOLLOW(B) = {+, $,)}
FOLLOW(F) = {+, $,)}

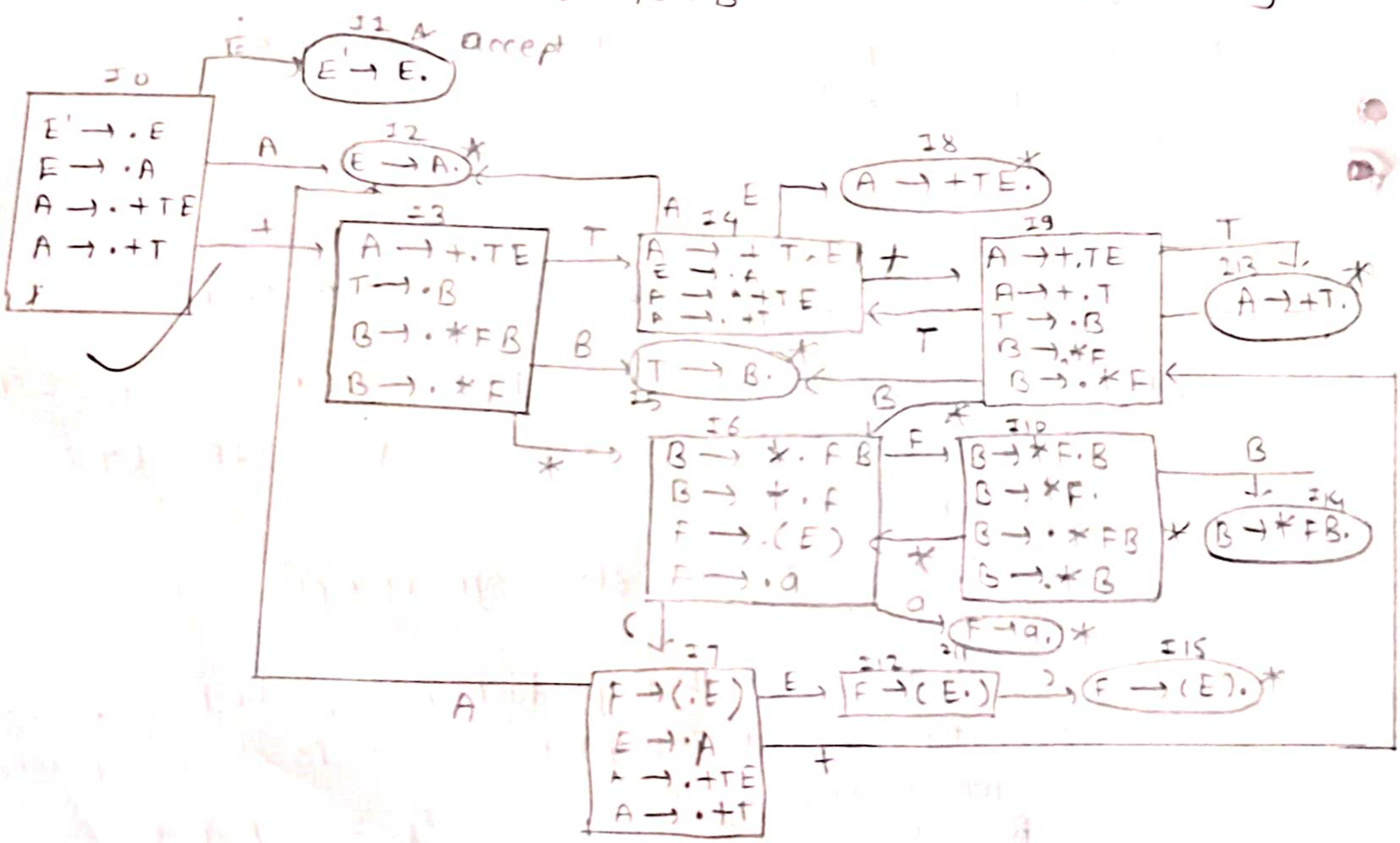
step ① find augmented grammer

The augmented grammer of given grammer is.

E' → .E [0th production]
E → .A [1th]
A → .+TE [2nd]
A → .+T [3rd]
T → .B [4th]
B → .*FB [5th]
B → .*F [6th]
F → .(E) [7th]
F → .a [8th]

Step ② Find LR(0) collection of items

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



	a	(	)	*	+	$	E	A	B	T	F
I0					S3		1	2			
I1						Accept					
I2			R1			R1					
I3				S6					5	4	
I4					S9		8	2			
I5			R4		R4	R4					
I6	S11	S7									10
I7							12	2			
I8			R2			R2					
I9				S6						13, 4	
I10				S6					14		
I11			R8		R8	R8					
I12			S15								
I13			R3			R3					
I14			R5		R5	R5					
I15			R7		R7	R7					

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

- A grammer without ϵ is LL(1) if
  - for every production of $A \to \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \dots \alpha_n$ the set FIRST(α_1), FIRST(α_2), FIRST(α_3) - FIRST(α_n)
  - FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ FIRST(α_3) $\cap$ --- FIRST(α_n) = ϕ
- A grammer with ϵ is LL(1) if
  - for every prod. of $A \to a \mid \epsilon$
  - first(α) $\cap$ Follow(A) = ϕ

$A \to \underbrace{+TA}_{\alpha_1} \mid \underbrace{+T}_{\alpha_2}$

first(α_1) $\cap$ first(α_2)

First(+TA) $\cap$ first(+T)

{+} != {ϕ}

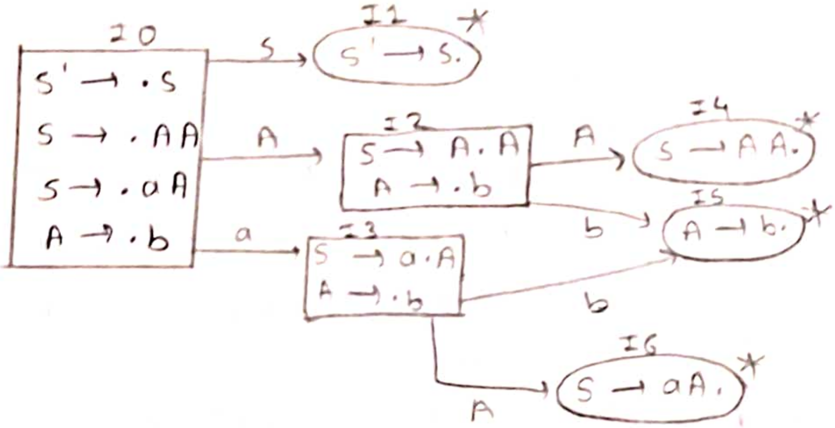
so given grammer is not LL(1)

3) S → A A
S → a A
A → b

solution step ① Find augmented grammer
The augmented grammer of given grammer is,

$S' \to .S$ [0th production]
$S \to .AA$ [1th production]
$S \to .aA$ [2nd production]
$A \to .b$ [3rd production]

step ② Find LR(0) collection of items below is fig showing the LR(0) collection of items



First we have to findout FIRST & FOLLOW

i) Rule No 1: If terminal symbol a then FIRST(a) = {a}
ii) Rule No 2: If their is rule x producing ε then FIRST(x) = ε
iii) Rule No 3: For the rule a is producing the $Y_1 Y_2 --- Y_N$
Then $FIRST(A) = FIRST(Y_1) \cup FIRST(Y_2) \cup --- FIRST(Y_N)$

② Rules for follow:
i) Rule No 1: for the start symbol S place $ in FOLLOW(S)
ii) Rule No 2: If their production A producing $\alpha B \beta$ everything in FIRST(β) without ε is to be placed in FOLLOW(B)
iii) Rule No 3: If their is production A producing $\alpha B \beta$ or A producing αB & FIRST(B) = ε then FOLLOW(A) = FOLLOW(B) or FOLLOW(B) = FOLLOW(A)

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

	a	b	$	S	A
I0	S3			1	2
I1			Accept		
I2		S_5			4
I3		S_5			6
I4			R1		
I5		R3	R3		
I6			R2		

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

- A grammer without ϵ is LL(1) if
  - for every production of $A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 | --- \alpha_n$ the set FIRST(α_1), FIRST(α_2), FIRST(α_3) --- FIRST(α_n)
  - FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ FIRST(α_3) $\cap$ ---- FIRST(α_n) = ϕ
- A grammer with ϵ is LL(1) if
  - For every production of the $A \rightarrow a | \epsilon$
  - FIRST (α) $\cap$ FOLLOW(A) = ϕ

$S \rightarrow \underset{\alpha_1}{AA} \,|\, \underset{\alpha_2}{aA}$

FIRST(α_1) $\cap$ FIRST(α_2)

FIRST(AA) $\cap$ FIRST(aA)

{b} $\cap$ {a}

{ϕ}

given grammer is LL(1)

4) $S \rightarrow A$

$A \rightarrow aaA$

$A \rightarrow b$

Solution : step① : Find augmented grammer

The augmented grammer of given grammer is,

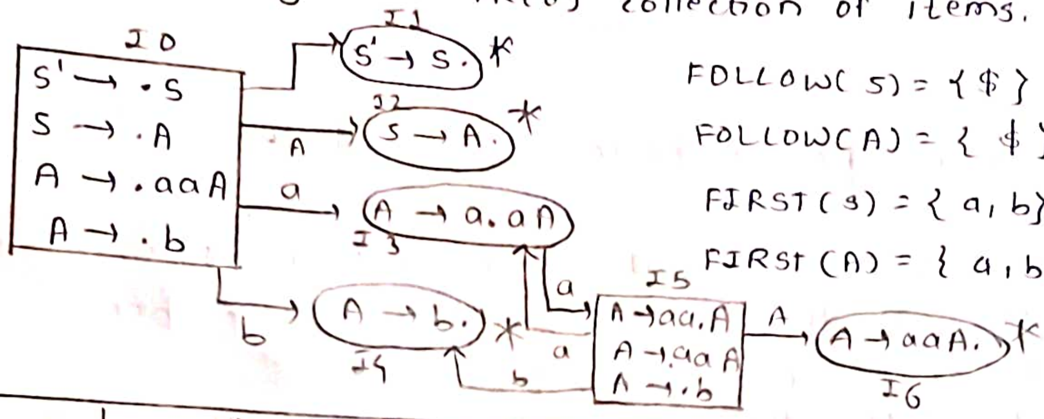
$S' \rightarrow .S$ [0th production]

$S \rightarrow .A$ [1th production]

$A \rightarrow .aaA$ [2nd production]

$A \rightarrow .b$ [3rd production]

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



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

	a	b	\$	S	A
I0	S3	S4		1	2
I1			accept		
I2			R1		
I3	S5				
I4			R3		
I5	S3	S4	R2		6
I6			R2		

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

- A grammer without ϵ is LL(1) if
  - for every production of $A \to \alpha_1 | \alpha_2 | \alpha_3 | \cdots \alpha_n$ the set FIRST(α_1), FIRST(α_2), FIRST(α_3) ---- FIRST(α_n)
  - FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ --- FIRST(α_n) = ϕ
- A grammer with ϵ is LL(1) if
  - for every production of $A \to a | \epsilon$
  - FIRST(α) $\cap$ FOLLOW(A) = ϕ

$A \to aaA \mid b$
$\quad \alpha_1 \quad\; \alpha_2$

FIRST(α_1) $\cap$ FIRST(α_2)
FIRST(aaA) $\cap$ FIRST(b)
{ ϕ }

given grammer is LL(1)

5) $S \to Ab$
$A \to aA$
$A \to ab$
$A \to \epsilon$

→ For calculating FIRST & FOLLOW follow Following Rules.

① FIRST :-

i) Rule No 1 : If terminal symbol a then FIRST(a) = {a}

ii) Rule No 2 : If their is rule x producing α then FIRST(x) = ϵ

iii) Rule No 3 : For the rule a is producing the $Y_1, Y_2 \ldots Y_N$ Then FIRST (A) = FIRST (Y_1) U FIRST (Y_2) U ... FIRST (Y_N)

② FOLLOW

i) Rule No 1 : for the start symbol s place \$ in FOLLOW(S).

ii) Rule No 2 ; If their production A producing $\alpha B \beta$ everything in FIRST (β) without ϵ is to be placed in FOLLOW(B)

iii) Rule No 3 : If their is production A producing $\alpha B \beta$ or A producing αB & FIRST (B) = ϵ then FOLLOW(A) = FOLLOW(B) or FOLLOW (B) = FOLLOW(A)

FIRST(S) = {a, b, ϵ}
FIRST (A) = {a, ϵ}.

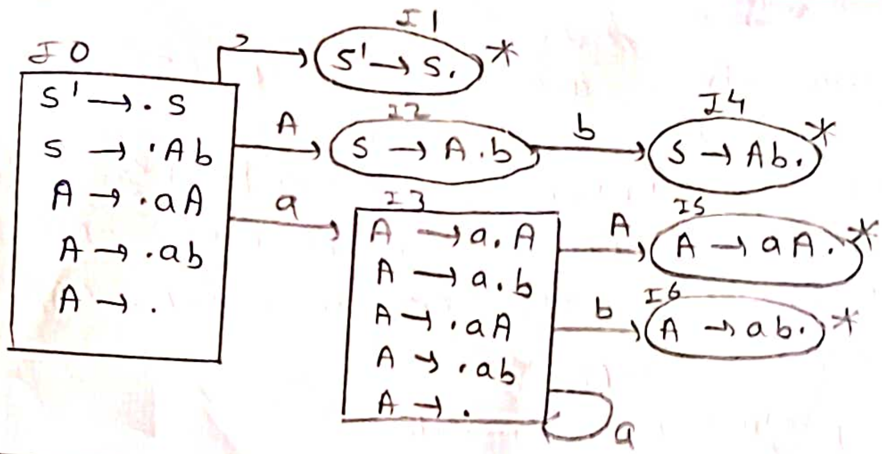
FOLLOW(S) = {\$}
FOLLOW(A) = {b}.

step ① find augmented grammer

The augmented grammer of given grammer is

$S' \to .S$ [0th production]
$S \to .Ab$ [1st]
$A \to .aA$ [2nd]
$A \to .ab$ [3rd]
$A \to .\epsilon$ [4th]

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



	a	b	$	S	A
I0	S3			1	2
I1			Accept		
I2		S4			
I3	S3	S6			5
I4			R1		
I5		R2			
I6		R3			

~~Ru~~. The terminals of thi's grammer are {a, b}
The non-terminals of thi's grammer are {S, A}

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

- I0 gives A in I2 so 2 is added to A column of 0 row
- I0 gives S in I1, so 1 is added to S column of 1 row
- similary 5 is written in A column of 4 row.
- I0 gives a in I3, so S3 (shift 3) is added to a column of 0 row.

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

- A grammer without ϵ is LL(1) if
  - for every production of $A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 --- \alpha_n$

the set FIRST(α_1), FIRST(α_2), FIRST(α_3) --- FIRST(α

- FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ FIRST(α_3) $\cap$ --- FIRST(α_n) $= \phi$

- A grammer with ϵ is LL(1) if
  - for every production of $A \rightarrow a | \epsilon$
  - FIRST(α) $\cap$ FOLLOW(A) $= \phi$

$A \rightarrow aA \mid ab \mid \epsilon$
$\quad\;\; \alpha_1 \quad \alpha_2 \quad \alpha_3$

FIRST(α_1) $\cap$ FIRST(α_2) $\cap$ FOLLOW(α_3)

$\{a\} \cap \{a\} \cap$ FOLLOW($A \rightarrow \cancel{\epsilon}$) $= \{b\}$

$\{a\} \; != \{\phi\}$

given grammer is not LL(1) grammer.

21/02/24