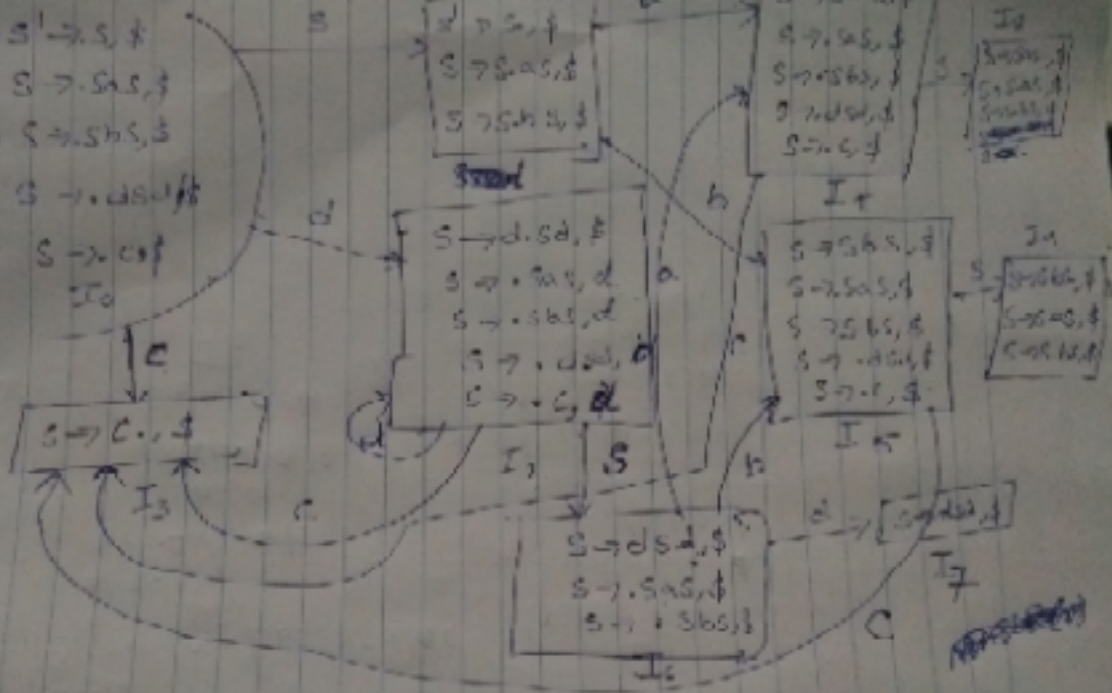


STATE	Action				
	a	b	d	c	\$
0		s_2		s_3	s_1
1	s_4	s_7			
3			R_4		
4		s_6	R_4		B
5		s_6	R_4		q
6	s_4	s_7	R_3		
8	s_4/R_1	s_5/R_1			

- Not a CLR(1) Grammar
- we can not parse the sentence
- at state 8 we find shift & reduce conflict

$S \rightarrow SaS / SbS / dSa / c$

- 0 $S' \rightarrow S, \$$
- 1 $S \rightarrow SaS, \$$
- 2 $S \rightarrow SbS, \$$
- 3 $S \rightarrow dSa, \$$
- 4 $S \rightarrow c, \$$



STATE

Action

Go to

Calc

Symbol / I/P Action

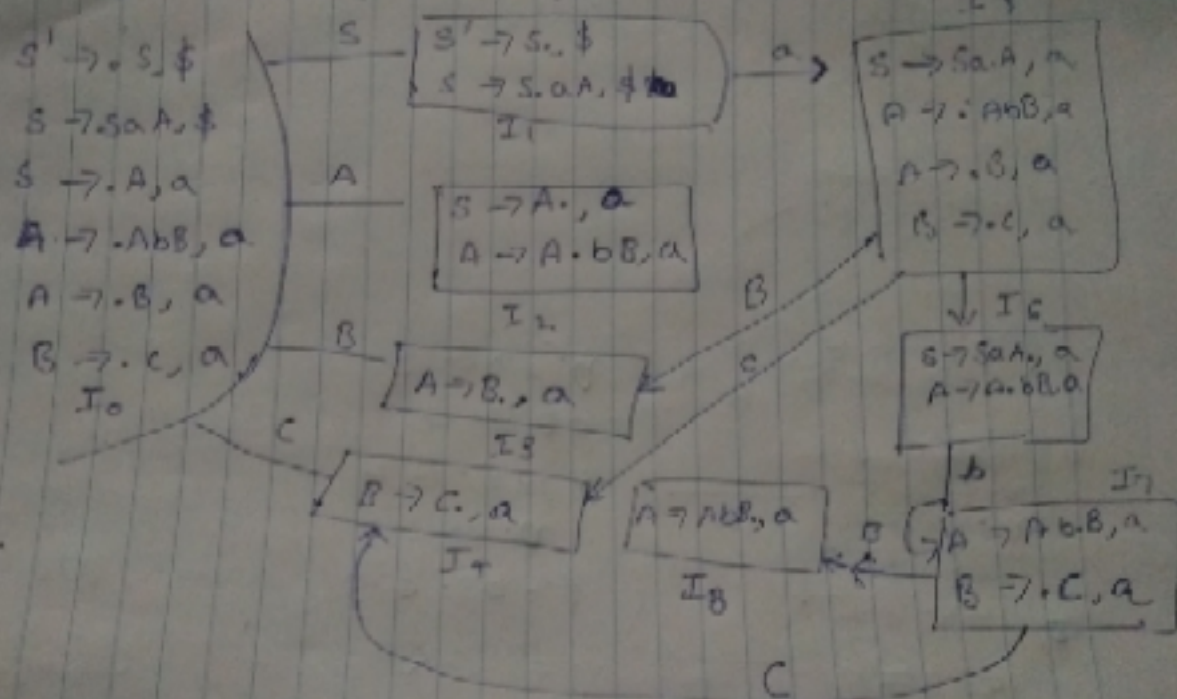
STATE	Action	Go to	Calc	Symbol / I/P Action
0		S, A, R	Stack of ANS	
1	S ₁	1, 2, 3	0	↑ clear shift
2	S ₂			↓ accept
3	r ₃	r ₃		
4	r ₄	r ₄		
5	r ₇ S ₄	8, 3		
6	r ₁ S ₁			
7	r ₄	8		
8	r ₂	r ₂		

No transition from C to h

Can not parse for string.

Q3 Type 1

$S \rightarrow SaA / A, A \rightarrow AbB / B, B \rightarrow C.$



0163	\$ S+AB	25
01637	\$ S+AB	\$
01632	\$ S+AB	\$
01631	\$ S+A	\$
0163	\$ S	\$
01	\$	\$

$S \rightarrow$
 Rewrite $B \rightarrow$
 $A \rightarrow AB$
 $S \rightarrow A$

Q + b a v	Stack	Symbol	Input	Action
	\$0	\$	a b a v \$	S ₁
	01	\$a	+ b a v \$	B → a Reduce
	03	\$B	+ b a v \$	A → B Reduce
	02	\$A	+ b a v \$	S → A Reduce
	01	\$S	+ b a v \$	S ₀
016	\$S+	b a v \$	S ₅	
0165	\$S+b	a v \$	Reduce B → b	
0163	\$S+B	a v \$	Reduce A → B	
01634	\$S+A	a v \$	S ₁	
016341	\$S+A a	v \$	Reduce B → a ^{z₂}	

\$LR

State

0 1 2 3 4 5 6 7

+ Action b, \$

Goto
3 A B

S₆ S₄ S₅ accept

S₄ S₅

S₇R₄ R₄ R₄ R₄ R₄

R₆ R₇

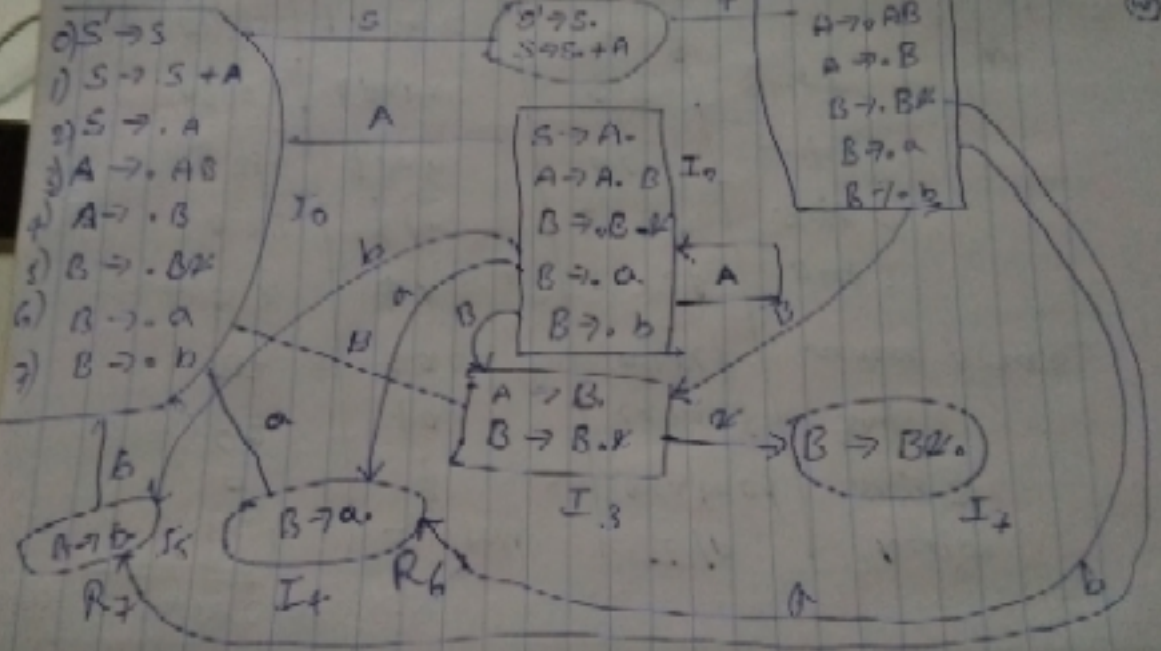
S₇

R₅

2 3

3

$S \rightarrow S+AA$
 $A \rightarrow AB/B$
 $B \rightarrow BA/a/b$



i/p a c a b a c

Stack	Symbols	I/p	Action
0	\$	a c a b a c	Shift 3
0 3	\$ a	c a b a c	Shift 7
0 3 7	\$ a c	a b a c	reduce. <u>A → a c</u>
0 2	\$ A	a b a c	Shift 3
0 2 3	\$ A a	b a c	Shift 6
0 2 3 6	\$ A a b	a c	Shift 1
0 2 3 6 3	\$ A a b a	c	Shift 7
0 2 3 6 3	\$ A a b a c		

→ Can not parse.
the string

ii) SLR

State	Action				Go to	
	a	b	c	\$	S	R
I ₀	S ₃		S ₄		1	2
I ₁	Accept		Accept			
I ₂	S ₃		S ₄			
I ₃		S ₆	S ₇			
I ₄			r ₃			
I ₅				Accept		
I ₆	S ₃		S ₄		8	2
I ₇	r ₄		r ₄			
I ₈				Accept		

Follow(S) = {b}

Follow(*) = {a, c}

Yes
Grammar is
SLR(1)



iii) abacbb

Line	Stack	Symbols	Input	Action
1) 0	0	\$	abacbb\$	S ₃
2) 03	03	\$ab	bacbb\$	S ₈
3) 038	038	\$ab	acbb\$	reduced A → ab
4) 01	01	\$A	acbb\$	S ₃
5) 013	013	\$Am	acbb\$	S ₄
6) 0134	\$Aa_c	cb\$		No action defined Rejected

The string abacbb is not SLR.

CFG: $S \rightarrow AS | acs | c$

$A \rightarrow aab$

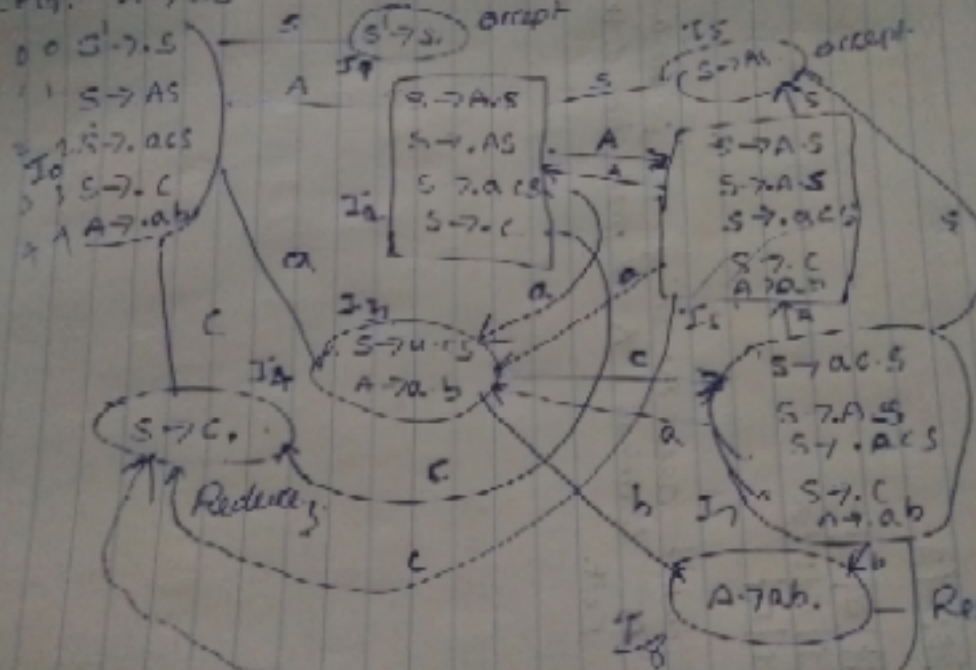
SLR

State	Action				Goto	
	a	b	c	\$	S	A
I_0	S_3		S_4		I_1	I_2
I_1	Accept					
I_2	S_3		$R_5 S_4$		5	6
I_3		S_8	S_7			
I_4				δ_3		
I_5	Accept					
I_6	S_3		S_4		5	1
I_7	S_3	S_8	S_4		5	6
I_8	δ_4	δ_4				

follows(S) = $\{$

follow(A) = $\{a, b\}$

CFG: $S \rightarrow AS | acS | a$
 $A \rightarrow ab$



NO
Shift
Reduce
conflict
if LR(0)
Grammar

②

String: $\alpha \wedge (a b v c) \$$

\$	D	B
----	---	---

\$	D	E	C
----	---	---	---

\$	D	E	α
----	---	---	----------

\$	D	
----	---	--

Can't parse the string as for ~~product~~
~~terminal~~ \wedge production doesn't
 Exit.

$$\text{Follow } B = \text{First}(D) = \{\wedge\}$$

$$\text{Follow } E = \text{Follow}(B) = \{\wedge\}$$

$$\text{Follow } C = \text{First}(E) = \{v\}$$

IV

LL(1)

	V	N	\wedge	()	a	b	c	\$
S			$S \rightarrow BD$	$S \rightarrow BD$		$S \rightarrow BD$	$S \rightarrow BD$	$S \rightarrow BD$	

D	$D \rightarrow \wedge BD$							$D \rightarrow E$	
---	---------------------------	--	--	--	--	--	--	-------------------	--

B		$B \rightarrow CE$	$B \rightarrow CE$		$B \rightarrow CE$	$B \rightarrow CE$	$B \rightarrow CE$	$B \rightarrow CE$	
---	--	--------------------	--------------------	--	--------------------	--------------------	--------------------	--------------------	--

E $E \rightarrow CE$

C		$C \rightarrow \wedge C$	$C \rightarrow CS$		$C \rightarrow a$	$C \rightarrow b$	$C \rightarrow c$		$E \rightarrow E$
---	--	--------------------------	--------------------	--	-------------------	-------------------	-------------------	--	-------------------

iii)

First:

$$\text{First}(S) = \text{First}(B) = \text{First}(C) \\ = \{n, c, a, b, e\}$$

$$\text{First}(D) = \{\Lambda, e\}$$

$$\text{First}(B) = \{n, c, a, b, e\}$$

$$\text{First}(E) = \{v, e\}$$

$$\text{First}(C) = \{n, c, a, b, e\}$$

Follow:

$$\text{Follow}(S) = \text{First}(B) = \text{First}(C) \\ = \{n, c, a, b, e\}$$

$$\text{Follow}(D) = \{\Lambda, e\} \cup \{\$ \}$$

Follow:

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

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

type 2 (Q1)

(Q1)

$$S \rightarrow SA \wedge B / B$$

$$B \rightarrow BV C / C$$

$$C \rightarrow \wedge C / (S) / a / b / c.$$

① left recursion

$$S \rightarrow BD$$

$$D \rightarrow \wedge BD / \epsilon$$

$$B \rightarrow CE$$

$$E \rightarrow VCE / \epsilon$$

$$C \rightarrow \wedge C / (S) / a / b / c.$$

② left Factoring.

$$S \rightarrow BD$$

$$D \rightarrow \wedge BD / \epsilon$$

$$B \rightarrow CE$$

$$E \rightarrow VCE / \epsilon$$

$$C \rightarrow \wedge C / (S) / a / b / c$$

(v)

string: $\phi \neq \phi \neq \phi \neq \phi \neq \phi$

$\$ | D | B$

$\$ | D | E | C$

$\$ | D | E | \phi$

$\$ | D | E | \phi | C | X$

$\$ | D | E | C$

~~String is Not a LL(1) Grammar as no production for E on terminal C~~

$\$ | D | E | \phi | S | \phi$

$\$ | D | E | \phi | D | B$

$\$ | D | E | \phi | D | E | C$

$\$ | D | E | \phi | D | E | C | \phi$

$\$ | D | E | C | D | E | \phi$

String is
Not a LL(1)
Grammar as
No production
on E @
terminal ϕ

$$\text{First}(D) = \{v, b\}$$

$$\text{First}(B) = \{n, c, a, b, c\}$$

$$\text{First}(E) = \{\wedge, E\}$$

$$\text{First}(C) = \{n, c, a, b, c\}$$

Follow:

Follow of S: $\{\$$

Follow of D: $\{\$$

Follow of B: $\text{First}(D) \Rightarrow \{v\}$

Follow of E: $\text{follow}(B) \Rightarrow \{v\}$

Follow of C: $\text{First}(E) \Rightarrow \{\wedge\}$

IV

LL(1): G_2

	v	\wedge	n	c		a	b	c	\$
S			$S \rightarrow BD$	$S \rightarrow BD$		$S \rightarrow BD$	$S \rightarrow BD$	$S \rightarrow BD$	
D	$D \rightarrow vBD$								$D \rightarrow C$
B			$B \rightarrow CE$	$B \rightarrow CE$		$B \rightarrow CE$	$B \rightarrow CE$	$B \rightarrow CE$	
E		$E \rightarrow \wedge EE$							$E \rightarrow B$
C			$C \rightarrow nC$	$C \rightarrow c$		$C \rightarrow a$	$C \rightarrow b$	$C \rightarrow c$	

Q1 type 1

CD
Mid: 0

- ① $S \rightarrow SVB/B$
 $B \rightarrow BAC/C$
 $C \rightarrow \omega C/(S)/a/b/C$

① Remove left recursion:

$$G_1 = \begin{aligned} S &\rightarrow BD \\ D &\rightarrow VBD/E \\ B &\rightarrow CE \\ E &\rightarrow \omega E/E \\ C &\rightarrow \omega C/(S)/a/b/C \end{aligned}$$

② NO Common prefixes $(A \rightarrow a\alpha_1/a\alpha_2/a\alpha_3)$
 $A = aA'$
 $A' = \alpha_1/\alpha_2/\alpha_3$

$$G_2 = \begin{aligned} S &\rightarrow BD \\ D &\rightarrow VBD/E \\ B &\rightarrow CE \\ E &\rightarrow \omega E/E \\ C &\rightarrow \omega C/(S)/a/b/C \end{aligned}$$

③ First

$$\begin{aligned} \text{First}(S) &= \{\text{First}(B) - \epsilon\} \cup \{\text{First}(D)\} \\ &= \{\text{First}(C)\} \cup \{V\} \\ &= \{\omega, c, a, b, e\} \cup \{V\} \\ &= \{\omega, c, a, b, e, V\} \end{aligned}$$