SLR parsing

PAGE NO.:
DATE: / /

Grammer :-

0th : S' → S
1st : S → L=R
2nd : S → R
3rd : L → *R
4th : L → id
5th : R → L

⇒ Grammar is already augmented by S'

⇒ new start symbol is S'.

compute "C" first :-

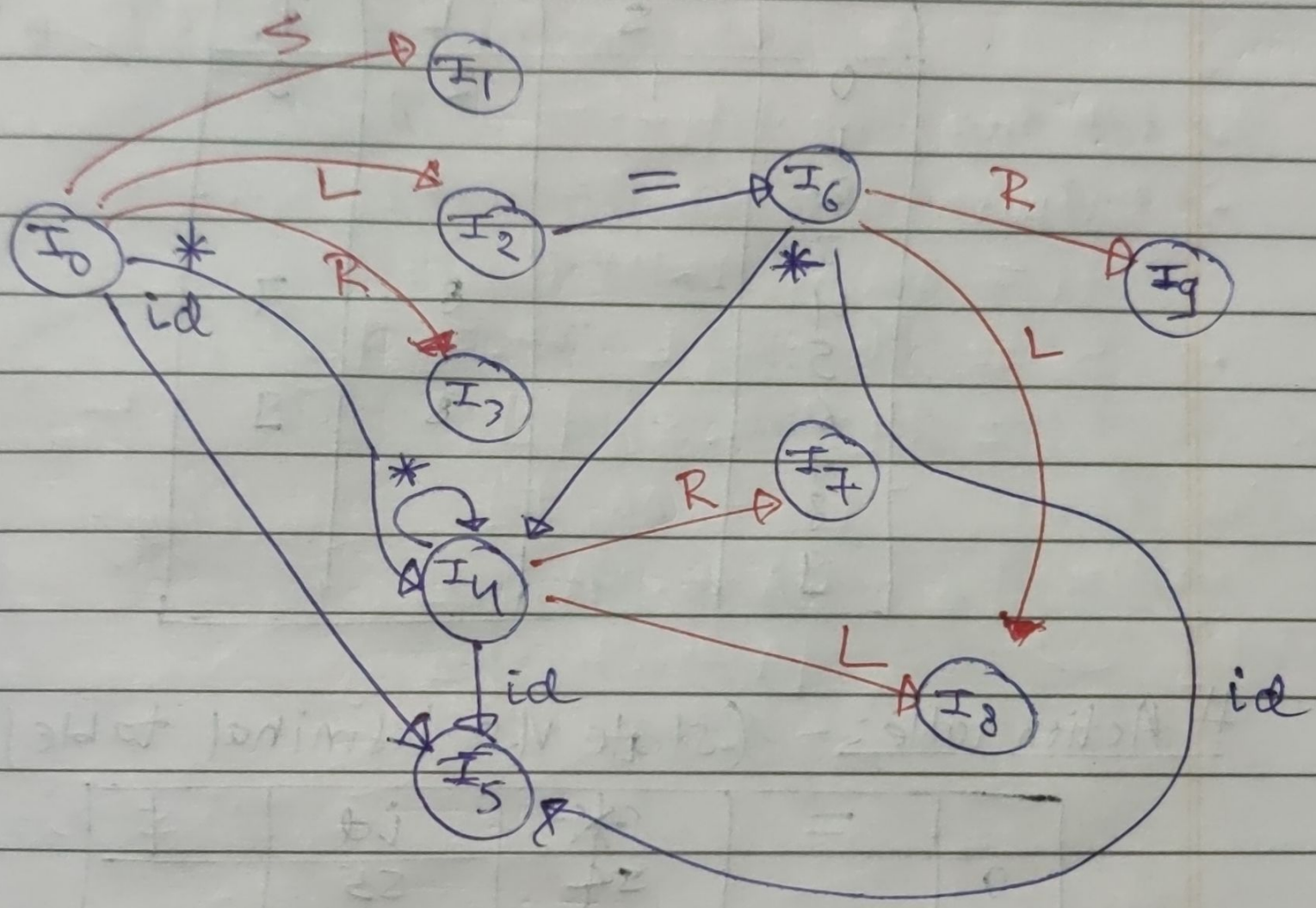
- Let C = { $I_0, I_1, I_2 \ldots I_n$ } where
  I_i = set of LR(0) item.
- I_0 = closure (S' → .S)
  = { S' → .S, S → .L=R
  S → .R
  L → .*R
  L → .id
  R → .L }
- I_1 = goto (I_0, S)
  = closure (S' → S.)
  = { S' → S. }
- I_2 = goto (I_0, L)
  = closure (S → L. =R) ∪ closure (R → L.)
  = { S → L. =R,
  R → L. }
- I_3 = goto (I_0, R)
  = closure (S → R.) = { S → R. }
- I_4 = goto (I_0, *)
  = closure (L → *. R) = { L → *. R, L → .id
  R → .L, L → .*R }

- $I_5 = goto(I_0, id) = closure(L \to id.)$
  $= \{L \to id.\}$
- $I_6 = goto(I_2, =)$
  $= closure(S \to L = .R)$
  $= \{S \to L = .R$
  $R \to .L$
  $L \to .*R$
  $L \to .id\}$
- $I_7 = goto(I_4, R) = closure(L \to *R.)$
  $= \{L \to *R.\}$
- $I_8 = goto(I_4, L) = \{R \to L.\}$
- $goto(I_4, id) = I_5$
  $goto(I_4, *) = I_4$
- $goto(I_6, *) = I_4$
  $goto(I_6, id) = I_5$
  $goto(I_6, L) = I_8$
- $goto(I_6, R) = closure(S \to L = R.)$
  $= \{S \to L = R.\} = I_9$

Hence, there will be total 10 states in SLR parsing grammar :-
[0, 1, 2 ..... 9]

Making the automata :-

- if goto $(I_i, X) = I_j$, then automata will have $(I_i \xrightarrow{X} I_j)$.



Computing Follow sets :-

- Follow(S') = Follow(S) = {$}
  Follow(S) = {$}
  Follow(L) = {=, $}
  Follow(R) = {=, $}

	S'	S	L	R
rule 1 :	$	$		
rule 2 :			=	
rule 3 : and 4			$	$, =

PAGE NO.:
DATE: / /

Goto table:- (state V/S Non-terminal).

- If goto(I_i, X) = I_j, then goto[i, X] = j
  ∀ non-terminal "X".

	S	L	R
0	1	2	3
1			
2			
3			
4		8	7
5			
6		8	9
7			
8			
9			

[easily made from automata]

Action table:- (state V/S terminal tables)

	=	*	id	$
0		S4	S5	
1				accept
2	S6, R5			R5
3				R2
4		S4	S5	
5	R4			R4
6		S4	S5	
7	R3			R3
8	R5			R5
9				R1

For making action table,

(i) Consider all terminal symbol transition in the automata to write "shift operation" in the table.

(ii) For reduce operations :

(i) Go to each I_i and look for a production of the form $A \to \alpha$. ("." at last)

(ii) look for follow(A)

(iii) action[i, follow(A)] = reduce no [$A \to \alpha$]

{Note that [$S' \to S.$] on encountering $, will be accept.}

Example:- In I_2, $[R \to L.] \Rightarrow$ follow(R) = {=, $}.

hence, action[2, =] = R5

action[2, $] = R5.

Note :- If the table have multiple entries, the grammar is not SLR(0) grammar.