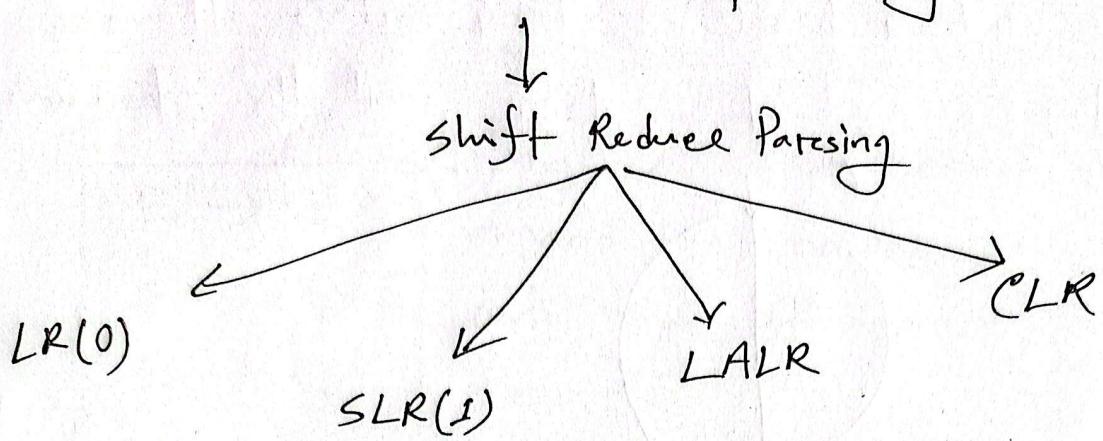


Types of bottom up parsing



LR(0) Parsing

↳ L means left to write scan

R produce a "leftmost derivation in reverse."

0 means zero lookahead symbols

LR(0) Parsing table,

↳ Need to find canonical items.

$$E \rightarrow T + E/T$$

$$T \rightarrow id$$

State	Action			Goto
	id	+	\$	
				E T

P.T.O

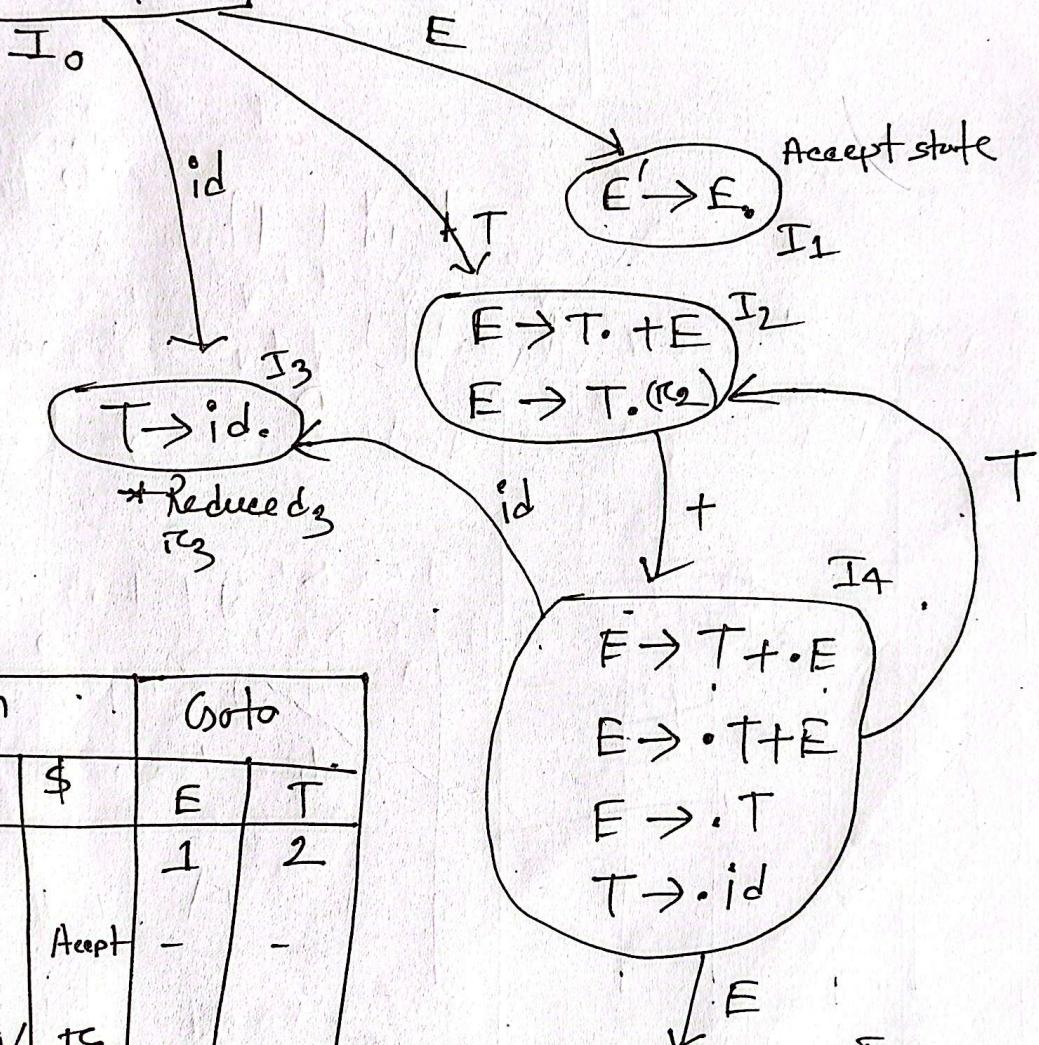
$$E \rightarrow \frac{1}{T+E/T^2}$$

$$T \rightarrow id$$

Augment.

$$\begin{array}{l} E' \rightarrow \cdot E \\ E \rightarrow \cdot T + E \\ E \rightarrow \cdot T \\ T \rightarrow \cdot id \end{array}$$

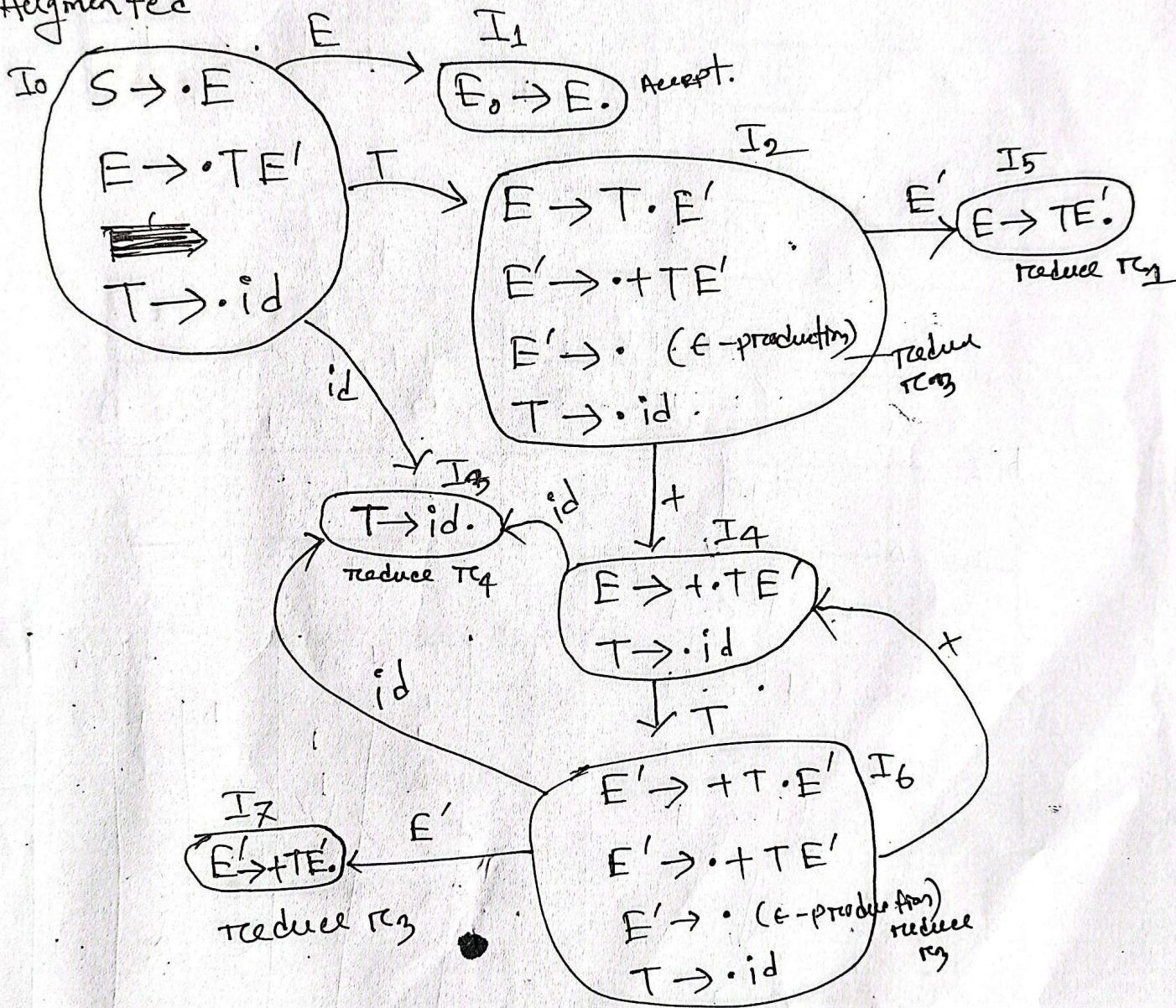
If you see a dot before a nonterminal, expand that nonterminal's productions until the dot appears before a terminal.



state	Action			Goto	
	id	$+$	$\$$	E	T
0	s_3	-	-	1	2
1	-	-	Accept	-	-
2	r_2	s_4/r_2	$+r_2$	-	-
3	r_3	r_3	r_3	-	-
4	s_3	-	-	5	2
5	r_1	r_1	r_1	-	-

- 1) $E \rightarrow TE'$
- 2) $E' \rightarrow +TE'$
- 3) $E' \rightarrow \epsilon$
- 4) $T \rightarrow id$

Augmented



LR(0) Parsing Table.

State	Action			GOTO		
	id	+	\$	E	T	E'
0	s_3			1	2	
1	-	-	Accept	-	-	-
2	s_3	s_4	r_{33}			5
3	r_4	r_4	r_4			
4	s_3				6	
5	r_{11}	r_{11}	r_{11}			
6	s_3	s_4	r_{33}			7
7	r_2	r_2	r_2			

Parsing the input id + id / id \$

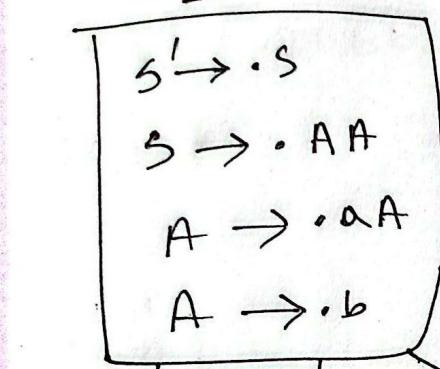
Stack	input	Action	Goto
0	id + id \$	$A[0, id] = S_3 \{ shift \}$ push id 3	
0 id 3	+ id \$	$A[3, +] = T C_4 \{ reduce \}$ $T \rightarrow id$ pop id 3 push T	
0 T	+ id \$		$goto(0, T) = 2$ push 2
0 T 2	+ id \$	$action[2, +] = S_4 \{ shift + \}$ push + 4	
0 T 2 + 4	id \$	$action[9, id] = S_3 \{ shift id \}$ push id 3	
0 T 2 + 4 id 3	\$	$action[3, \$] = T C_4 \{ reduce \}$ $T - id \}$ pop id 3 push T	
0 T 2 + 4 T	\$		$goto(4, T) = 6$ push 6
0 T 2 + 4 T 6	\$	$action[6, \$] = T C_3 \{ reduce E \rightarrow E' \}$ pop nothing push E'	
0 T 2 + 4 T 6 E'	\$	$action[7, \$] = R_2 \{ reduce E' \rightarrow + T E' \}$	$goto(6, \$) = 7$
0 T 2 + 4 T 6 $E' \times$	\$	$\overline{\text{pop } 2 * + T E' = 2 + 3 = 1 \text{ entries}}$ $\therefore \text{so pop } + 4 T 6 E' \times$ push E'	

stack	input	Action	goto
0T2E'	\$		goto [2, E'] = 5
0T2E'5	\$	Action [5, \$] = τ_1 reduce $E \rightarrow TE'$ pop $2 + TE' = 2 + 2 = 4$ entries so pop T2E'5 push E	
0E	\$		goto [0, E] = 1 push 1
0E1	\$	Action [1, \$] = Accept.	

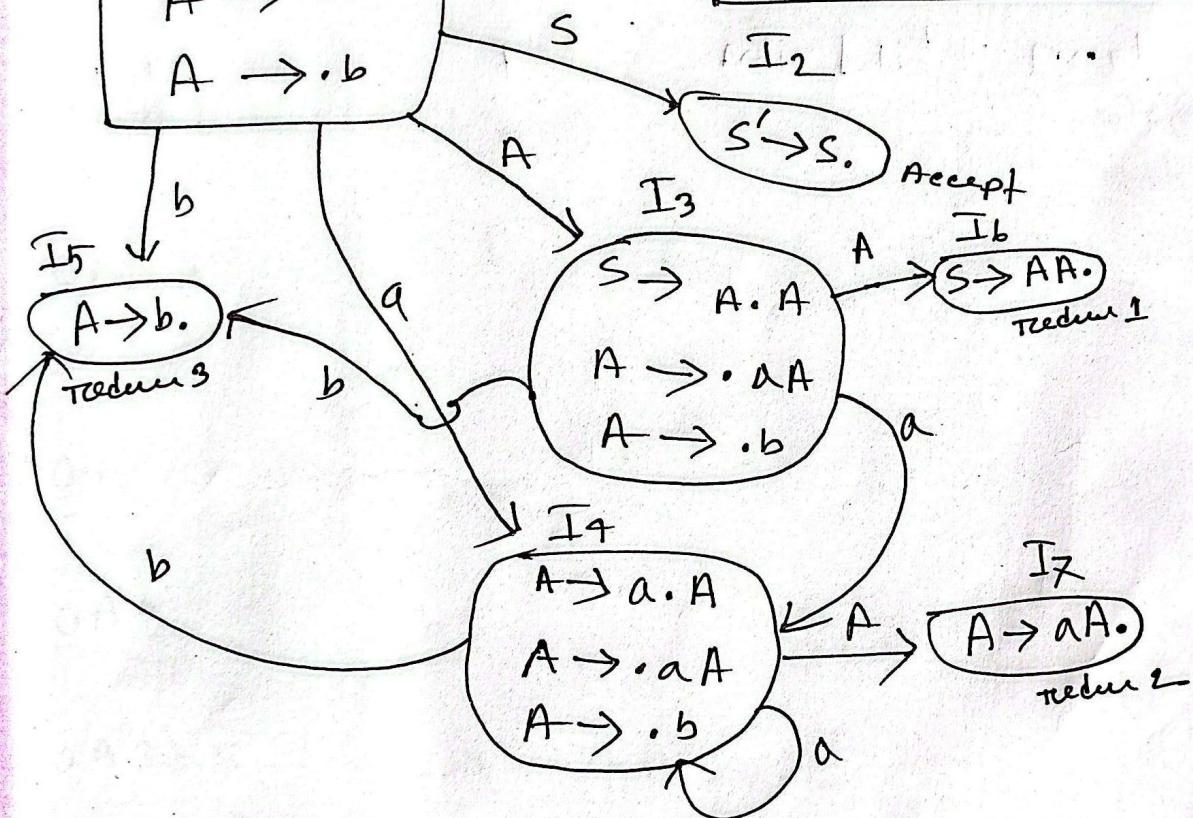
Check whether the given grammar is LR(0) or Not?

- i) $S \rightarrow AA$
- ii) $A \rightarrow aA$
- iii) $A \rightarrow b$

I_0



state	Action	goto
-	a b \$	S A
0	$s_4 s_5$	2 3
2	... Acc.	- -
3	$s_4 s_5$	6
4	$s_4 s_5$	7
5	$r_{c_3} r_{c_3} r_{c_3}$	
6	$r_{c_4} r_{c_4} r_{c_4}$	
7	$r_{c_2} r_{c_2} r_{c_2}$	



Parsing ~~aabb~~ aabb\$

stack	input	Action	goto
0	aabb\$	$(0, a) = S_4$ shift a ..	
0a4	bb\$	$(4, b) = S_5$ shift b	
0a4b5	b \$	$(5, b) = T_3$ reduce $A \leftarrow b$ pop b, 5 push A	
0a4A	b \$		goto $(q, A) = 7$ push 7
0a4A7	b \$	$(7, b) = T_2$ reduce $A \rightarrow aA$ pop a4A7 push A	
0A	b \$		goto $(0, A) = 3$ push 3
0A3	b \$	$(3, b) = S_5$ shift b	
0A3b5	\$	$(5, \$) = T_3$ reduce $A \rightarrow b$ pop b5 push A	
0A3A	\$		goto $(3, A) = 6$ push 6
0A3A1	\$	$(1, \$) = T_1$ reduce $S \rightarrow AA$ pop A3A6 push S	
0S	\$		goto $(0, S) = 2$
0S2	\$	$(2, \$) = \text{acc}$	✓