

$$1) S \rightarrow S + T$$

$$2) S \rightarrow S - T$$

$$3) S \rightarrow T$$

$$4) T \rightarrow T * F$$

$$5) T \rightarrow T / F$$

$$6) T \rightarrow F$$

$$7) F \rightarrow \text{num}$$

$$8) F \rightarrow (S)$$

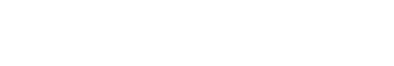
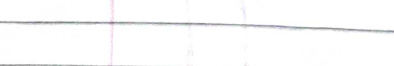
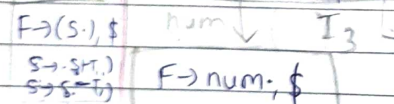
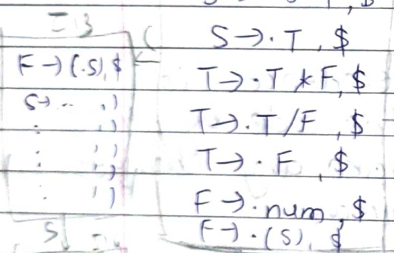
Grammar for
Calculator

To check whether this is LALR(1) or not:

DFA \Rightarrow

for

LR(1)



States	Action								GOTO		
	+	-	*	/	num	()	\$	S	T	F
0					S3	S13			1	2	12
1	S4	S5									
2			S7	S6				acc			
3							R3	R3			
4							R7	R7			
5					S3	S13				8	12
6					S3	S13				9	12
7					S3	S13					10
8					S3	S13					11
9			S7	S6			R1	R1			
10			S7	S6			R2	R2			
11							R5	R5			
12							R4	R4			
13							R6	R6			
14	S4	S5			S3	S13			14	2	12
15							S15				
							R8	R8			

As LR(1) is too big, I directly drew LALR(1) parse table.

From this, we can see that there is no LR or RR conflict. Hence grammar is LALR(1).

Attribute Grammar

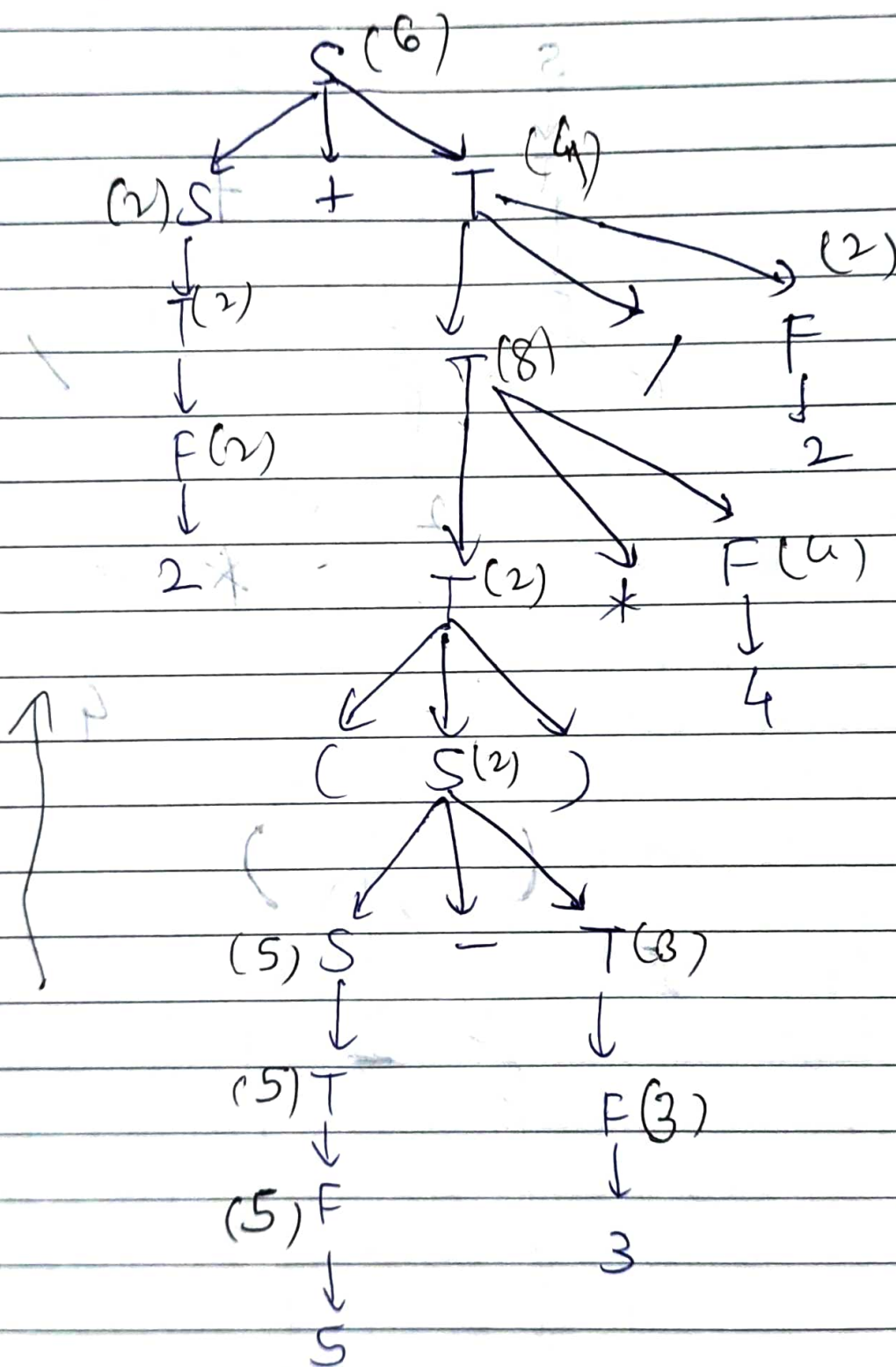
We consider a synthesized attribute 'value' which gives value till each node.

- 1) $S \rightarrow S + T$, $\{ S.value = S_1.value + T.value \}$
- 2) $S \rightarrow S - T$, $\{ S.value = S_1.value - T.value \}$
- 3) $S \rightarrow T$, $\{ S.value = T.value \}$
- 4) $T \rightarrow T * F$, $\{ T.value = T_1.value * F.value \}$
- 5) $T \rightarrow T / F$, $\{ T.value = T_1.value / F.value \}$
- 6) $T \rightarrow F$, $\{ T.value = F.value \}$
- 7) $F \rightarrow num$, $\{ F.value = value(num) \}$
- 8) $F \rightarrow (S)$, $\{ F.value = S.value \}$

Note: - Operator Precedence is maintained in the code by YACC.
No need to make any changes in attribute grammar.

$- < + < * < / < ($

eg $2 + (5 - 3) \times 4 / 2$



Ans: - 6