$\begin{array}{c} 1) & S \rightarrow S + T \\ 2) & S \rightarrow S - T \\ 3) & S \rightarrow T \end{array}$ Grammar for	
3) 5 -> T Grammar for	
4) T > T *F Calculator	
$5) T \rightarrow T/F$	
$\begin{array}{c} 6) \uparrow \rightarrow F \\ 7) \not\models \rightarrow \text{num} \end{array}$	
S) F→ (S) To check whether this is LALR(1) or not: To check whether this is LALR(1) or not: TO CHECK whether this is LALR(1) or not:	
$DFA \Rightarrow IO S' \rightarrow S., $ + S \rightarrow S+.T, $ S \rightarrow S+T. $$	
(RU) S - S + T, \$ S - T, \$ T - T/F, \$ T - T./F, \$	
5-7.5 -12 F-7-num,\$	
$T \rightarrow T/F$ $S \rightarrow S-T$	
T-> T/F, \$ T-> T./F \$	
F-)(5.), \$ num T3	
16 S > S - T . T	
Fy(s).\$ Total Tot	
$T \rightarrow T/F \cdot , 4$ $T \rightarrow T/F \cdot , 4$	

			111	mille 1						Date		
e de la composition della comp	0.010	Action								5000		
_	STOTE	3 12+	1 -	*	/	num	1)	3	5	T	10
_	1	1 54	55	00		33	513		(2.3)	- No.	2	12
_	4	2	33	27			32	12	acc		5	
	1	12		57	ISG			R3	R3		5	
	1	, p		+-		53		R7	R7			
_	1					S3	513		=1 _[11]		8	12
-	+ 6			and the same of th	200	<u>S3</u>	213		y. 19		9	12
_	1 7	-			-	<u>S3</u>	213					16
_	8			107	C	S ₃	513	7.	39.5		The second second	7)
	q			S7	SG		.,,2	PIC	RI		8	
Whates may	16			57	56			R2	R2			
State of the last	1.	1	7	1 7 2			5	RS	RS	3.7		
Chicken Spirit	12		4	1 03			-	K4	R4		-57	
100								R6	RG		- 1	
	13	C	1 P	an		53	SI3			14	2	
1	19	54	55	O13"	15 376	12C N	2452	S15	9170	2		
+	<u></u> <u>LS</u>		The second		69			K8	R8	9/4/		
1	3	1911	J 0	5+ 1	1 4		Sui	- / 0//	روم الإلا	ud		
	15.	A.9-1	12 + 2		11:00	P. SOL	9 (71).	SILVI	5	3 8		
_		As LR(1) is too big I directly drew										
_		AS LR(1) is too big, I directly drew LALRID parse table.										
_	- 51	12 ad www slower same and sixon.										
_		From this, we can see that there is no										
_		bases of RR conflict. Hence grammon										
· ·		is LALRU).										
								1.00		of setting		

Attribute Grammar We consider a synthesized attribute 'value' which gives value fill each node. 1) S > S+T, S. value = S. value + T. value' 2) S > S - T, S. value = S. value - T. value' 3) S - T (S. value = T. value' 4) T > T * F (T. value = T. value * F. value') 5) T - T * F (T. value = T. value * F. value') 6) T - F (T. value = F. value') 7) F - num (F. value = Au value(num)) 8) F - (S) (F. value = S. value)

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

- < + < * 两 < / ()

2+(5-3)*4/2 F(2) (2) 5(2) (B) (5) 5 F(3) Ans: - 6