Assignment 10 – 29 - May - 2023

Implementation of LR parser

LR Parsing algorithm:

Input: An input string *w* and an LR parsing table with functions *action* and *goto* for grammar G.

Output: If *w* is in L(G), a bottom-up-parse for *w*; otherwise, an error indication.

Method: Initially, the parser has s_0 on its stack, where s_0 is the initial state, and *w*$ in the input buffer. The parser then executes the following program :

```
set ip to point to the first input symbol of w$;
repeat forever  begin
    let s be the state on top of the stack and
       a  the symbol pointed to by ip;
    if action[s, a] = shift s' then begin
       push a then s' on top of the stack;
       advance ip to the next input symbol
    end
    else if action[s, a] = reduce A→β then begin
         pop 2* | β | symbols off the stack;
         let s' be the state now on top of the stack;
         push A then goto[s', A] on top of the stack;
         output  the production A→ β
    end
    else if action[s, a] = accept then
         return
    else error( )
end
```

Write a Program to implement the LR parser. Consider 2 input Strings (to show accept and reject/error) for the Grammar given below & show the sequence of moves by input String using LR Parser.

[NOTE : Accept input via file]

1. **Roll_No: 2110342211IT001 - 2110972211IT016**

Grammar

S → E

E → E + T | T

T → T * F | F

F → id

States	Action				Go to		
	id	+	*	$	E	T	F
I_0	S_4				1	2	3
I_1		S_5		Accept			
I_2		R_2	S_6	R2			
I_3		R_4	R_4	R4			
I_4		R_5	R5	R5			
I_5	S4					7	3
I_6	S4						8
I_7		R1	S6	R1			
I_8		R3	R3	R3			

2. **Roll_No:2110491211IT017 - 2110491211IT033**

Grammar

R → S

S → (L) L

S → ε

L → int L

L → S

	ACTION				GOTO	
State	(	)	int	$	S	L
0	s2	r3		r3	1	
1				acc		
2	s2	r3	s4	r3	5	3
3		s6				
4	s2	r3	s4	r3	5	7
5		r5		r5		
6	s2	r3	s4	r3	5	8
7		r4		r4		
8		r2		r2		

3. Roll_No:2110491211IT034 - 2110491211IT051

Grammar

E → E + T

E →T

T → T * F

T →F

F →(E)

F → id

	Action						Goto		
State	id	+	*	(	)	$	E	T	F
0	S5			S4			1	2	3
1		S6				accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

4. Roll_No:2110491211IT052 - 2110491211IT068

Grammar

S' → S

S → A S

S → b

A → S A

A→ c

	ACTION			GOTO	
State	c	b	$	S	A
0	s4	s3		1	2
1	s4	s3	acc	6	5
2	s4	s3		7	2
3	r3	r3	r3		
4	r5	r5			
5	s4/r4	s3/r4		7	2
6	s4	s3		6	5
7	s4/r2	s3/r2	r2	6	5

5. **Roll_No:2110491211IT069 - 2110491211IT087**

Grammar

S → x

S → (L)

L → S

L → L , S

	ACTION					GOTO	
	(	)	x	,	$	S	L
1	s 3		s 2			4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			6	5
4					Acc		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			9	
9	r 4	r 4	r 4	r 4	r 4		