

$$E' \rightarrow E$$

$$\textcircled{1} E \rightarrow E + T$$

$$\textcircled{2} E \rightarrow T$$

$$\textcircled{3} T \rightarrow T * F$$

$$\textcircled{4} T \rightarrow F$$

$$\textcircled{5} F \rightarrow (E)$$

$$\textcircled{6} F \rightarrow id$$

Step 4: Construct Parsing Table

Action table says \rightarrow what to do if we get this token (terminal symbol)
 \rightarrow whether to shift or reduce
 \rightarrow whether to accept or error

GoTo table says \rightarrow where to go after a reduction

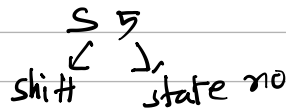
Rows \rightarrow for each of the states.

Columns \rightarrow all the tokens including \$

state	Action Table						GoTo Table		
	id	+	*	()	#	E	T	F
I_0	S5			S4			1	2	3
I_1		S6				accept			
I_2		R2	S7		R2	R2			
I_3		R4	R4		R4	R4			
I_4	S5			S4			8	2	3
I_5		R6	R6		R6	R6			
I_6	S5			S4				9	3
I_7	S5			S4					10
I_8		S6			S11				
I_9		R1	S7		R1	R1			
I_{10}		R3	R3		R3	R3			
I_{11}		R5	R5		R5	R5			

* It is not possible to see E' since not in body \therefore not in GoTo table.

* whenever we have a transition from one state to another on a terminal symbol, shift the destination state in stack.



* To reduce we have to check if there is any item with the cursor at the end.

* If I have a production rule with cursor at end \rightarrow apply reduction for all the terminal symbols of the head of the production rule.

* For $E' \rightarrow E$ we have only # in followset of E' , so we consider it as special case and do not reduce.

* For I_2 , we have $E \rightarrow T$. \therefore E's follows are

$\$, + ,)$ we will reduce for all of them.

$R \quad G$
 $\swarrow \quad \searrow$ no of production rule
reduce

② $E \rightarrow T$

④ $T \rightarrow F$

* Blank spaces indicate errors, $I_3 \rightarrow$ error
id cannot start with error.

$[S7, R2] \leftarrow$ Shift Reduce conflict
parser does not know whether to shift or
reduce. just by analyzing the grammar.
we need complex parsers for that.

There can also be reduce-reduce conflict.

Goto table

I_0 contains non-terminal and state 1 error.

→ total tokens (terminals)

* NO of columns to SLR(1) → $6P_1$

" " " " " (2) → $6P_2$

{ The automaton is LR(0)
The parse table is SLR(1)