$E \to E^{(1)} + T \mid T^{(2)}$

$T \to T^{(3)} * F \mid F^{(4)}$

$F \to (E)^{(5)} \mid id^{(6)}$

tree_mydesign.docx

test case1: (id+id)*id$

Stack Implementation: input: (id+id)*id$

Step	Stack	input	Action
step1	$0	(id+id)*id$	shift 4
step2	$0(4	id+id)*id$	shift 5
step3	$0(4id5	+id)*id$	Reduce 6: 1 element 2x1=pop 2 and add F. F→id
~~step~~			
step4	$0(4id5	+id)*id$	
step5	$0(4F3	+id)*id$	reduce 4: 1 element 2x1=pop 2 and add T. T→F
step6	$0(4F3	+id)*id$	
step7	$0(4T2	+id)*id$	reduce 2: 1 element 2x1=pop 2 element and add E. E→T
Step8	$0(4T2	+id)*id$	
step9	$0(4E8	+id)*id$	shift 6
step10	$0(4E8+6	id)*id$	shift 5
step11	$0(4E8+6id5	)*id$	reduce 6: pop 2 and add F
step12	$0(4E8+6F3	)*id$	
step13	$0(4E8+6T9	)*id$	reduce 4: pop 2 and add T
8			
step14	$0(4E8	)*id$	
step15	$0(4E8+6T9	)*id$	reduce 1: 3 elements 2x3=pop 6 and add E. E→E+T

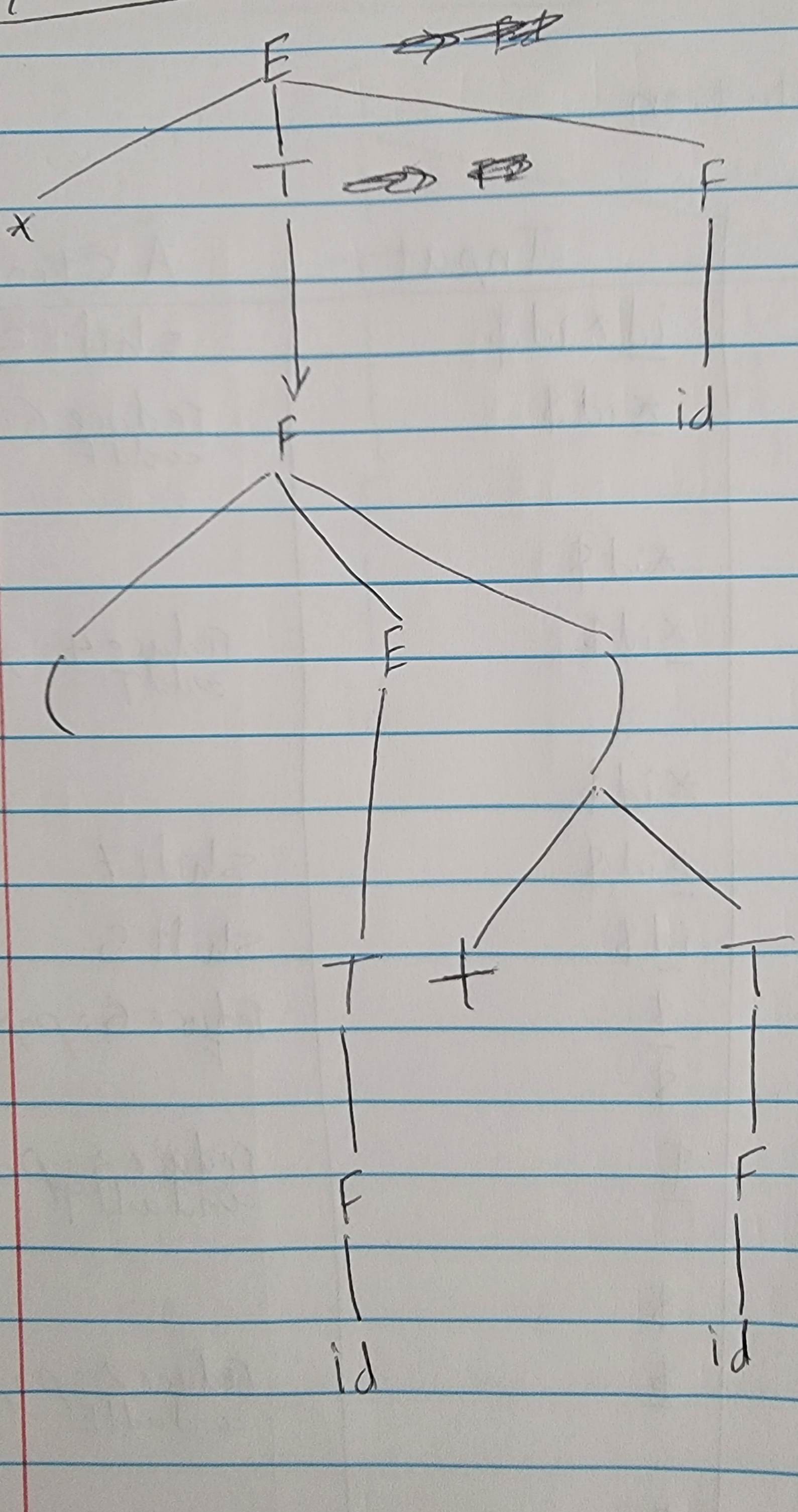
step 16	$0(4E8+6T9	)*id$	shift 11
step 17	$0(4E8	)*id$	F->(E) reduce 5: 3 elements 2x3=pop 6 and add F
step 18	$0(4E8)11	*id$	
step 19	$0(4E8)11	*id$	
Step 20	$0F3	*id$	reduce 4: pop 2 and add T
step 21	$0F3	*id$	
Step 22	$0T2	*id$	shift 7
step 23	$0T2*7	id$	shift 5
step 24	$0T2*7id5	$	reduce 6: pop 2 and add F
step 25	$0T2*7id5	$	
step 26	$0T2*7F10	$	reduce 3: 3 elements 2x3=pop 6 and add T
step 27	$0T2*7F10	$	
Step 28	$0T2	$	reduce 2: pop 2 and add E
step 29	$0T2	$	
step 30	$0E1	$	acc

test case 2: id*id$

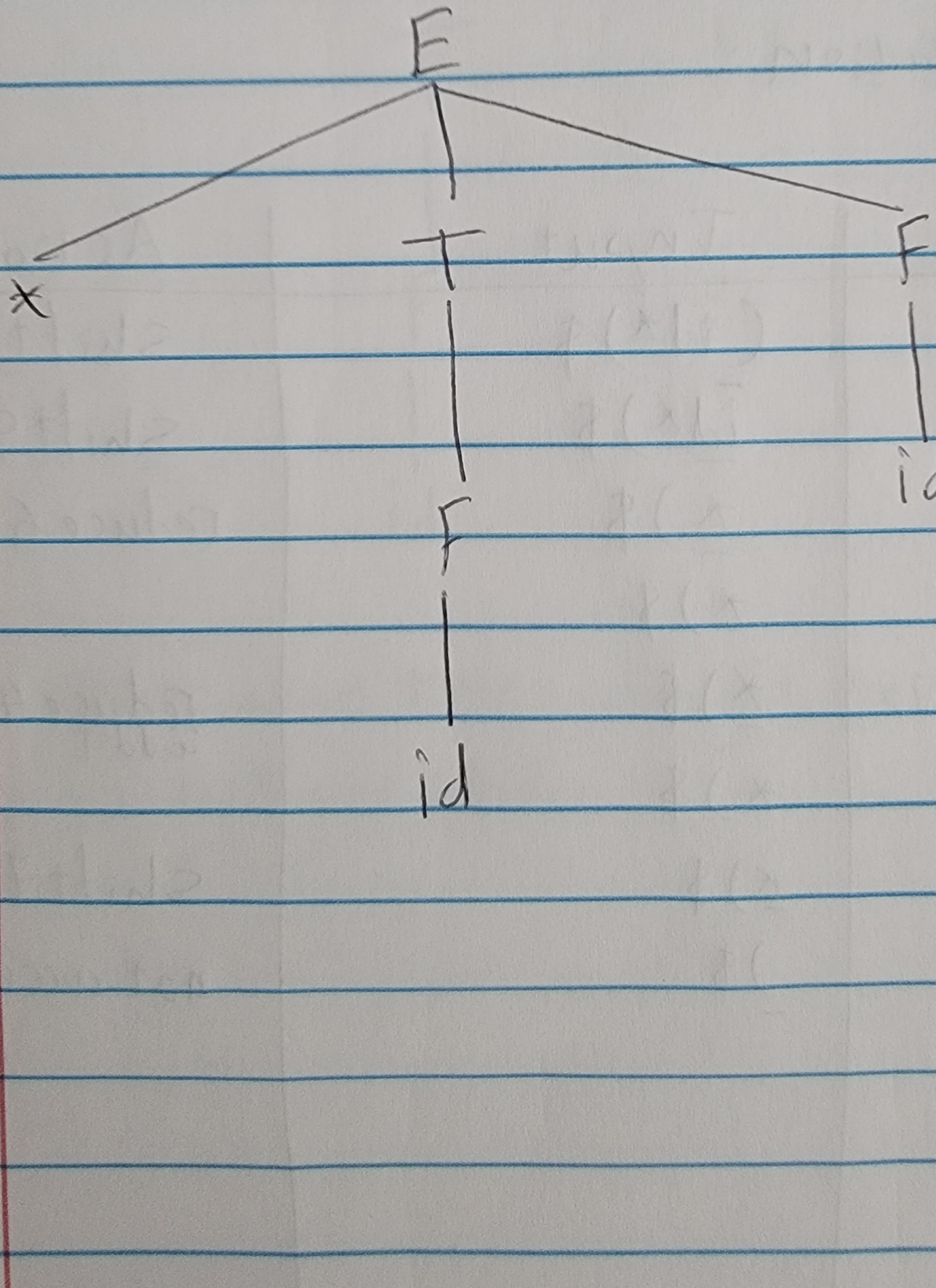
- Stack Implementation

Step	Stack	Input	Action
Step 1	$0	id*id$	shift 5
step 2	$0id5	*id$	reduce 6: pop 2 and add F
step 3	$0F3	*id$	
step 4	$0F3	*id$	reduce 4: pop 2 and add T
step 5	$0T2	*id$	
step 6	$0T2	*id$	shift 7
step 7	$0T2*7	id$	shift 5
step 8	$0T2*7id5	$	reduce 6: pop 2 and add F
step 9	$0T2*7F10	$	
step 10	$0T2*7F10	$	reduce 3: pop 6 and add T
Step 11	$0T2	$	
Step 12	$0T2	$	reduce 2: pop 2 and add E
Step 13	$0E1	$	
Step 14	$0E1	$	acc

- parse tree : (id+id)*id$



parse Tree: id * id $

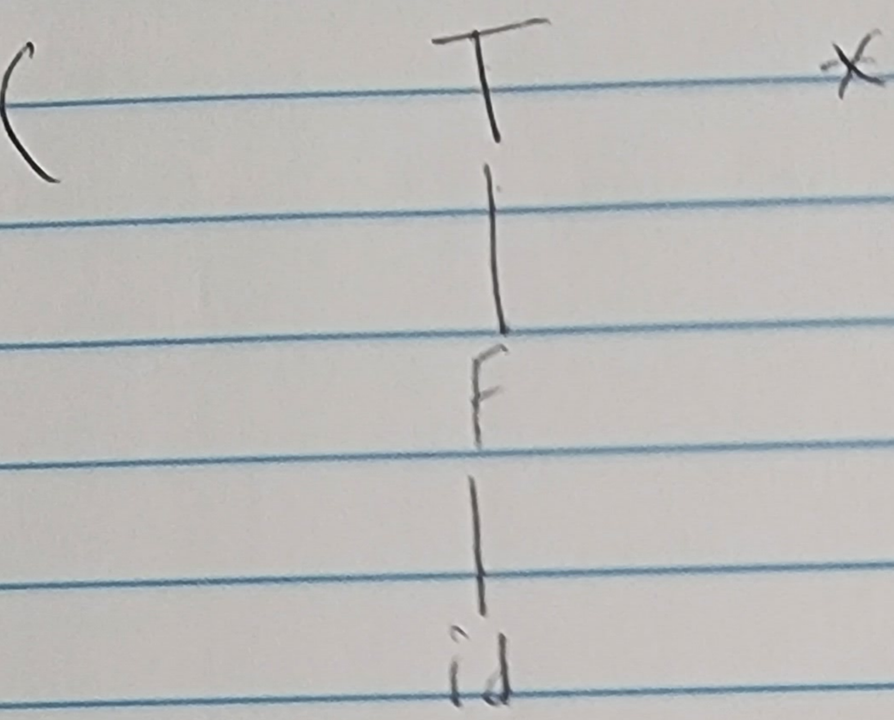


Test case 3: (id*)$

Stack Implementation:

Step	Stack	Input	Action
step 1	$0	(id*)$	shift 4
step 2	$0(4	id*)$	shift 5
step 3	$0(4id5	*)$	reduce 6: Pop 2 and add F
step 4	$0(4F	*)$	
step 5	$0(4F3	*)$	reduce 4: Pop 2 and add T
step 6	$0(4T	*)$	
step 7	$0(4T2	*)$	shift 7
step 8	$0(4T2*7	)$	not accepted

- parse Tree: (id *) $



- the parse tree doesn't exist because the input case is not accepted