Compiler Construction

Assignment #01

LR & SLR

Submitted by:

Sagar Acharya

LR(0) Parsing

① $S \to Aa$

$A \to bA \mid b$

Input: b b b b a

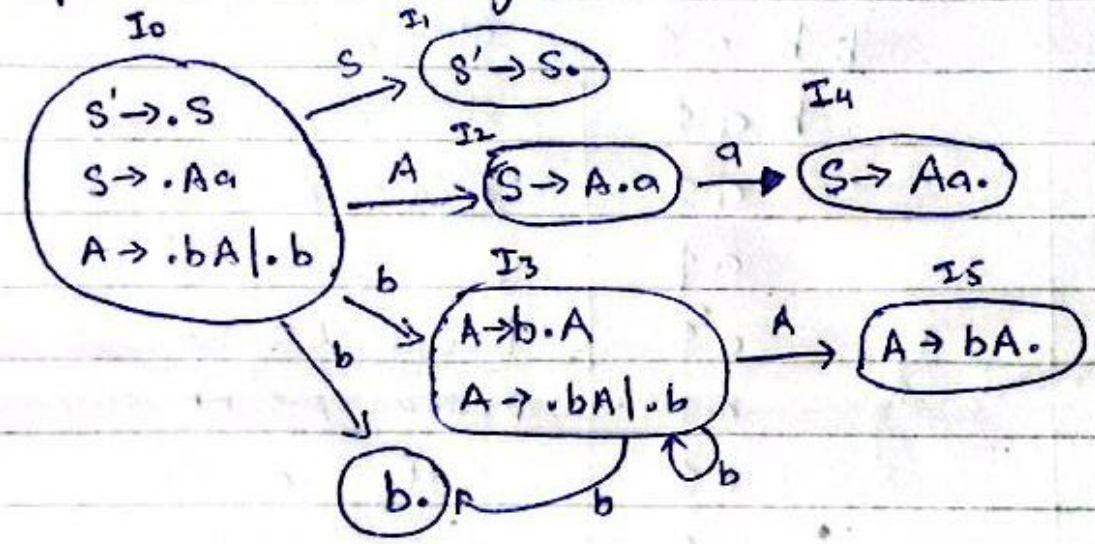
Sol Step1: Augment & Label

$S' \to S$ ⓪

$S \to Aa$ ②

$A \to bA \mid b$ ③ ④

Step2: Draw DFA Diagram



Step3: Parsing Table

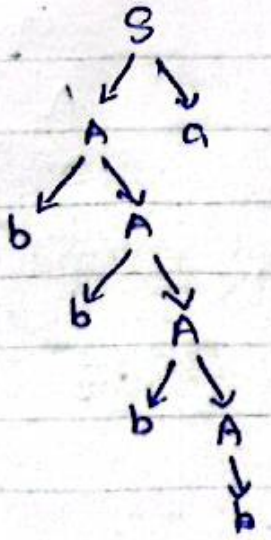
State	Action			GoTo	
	a	b	$	A	S
I_0		S_3		2	1
I_1			accept		
I_2	S_4				
I_3	R_3	S_3		5	
I_4	R_1	R_1	R_1		
I_5	R_2	R_2	R_2		

Step4: Stack Implementation (Stack Parsing Trace)

Input: b b b b a $

Stack (State)	Input Buffer	Action
$0	b b b b a $	Shift b → S_3
$0b3	b b b a $	Shift b → S_3
$0b3b3	b b a $	Shift b → S_3
$0b3b3b3	b a $	Shift b → S_3
$0b3b3b3b3	a $	Reduce $A \to b$ (R_3)
$0b3b3b3A5	a $	Reduce $A \to bA$ (R_2)
$0b3b3A5	a $	Reduce $A \to bA$ (R_2)
$0b3A5	a $	Reduce $A \to bA$ (R_2)
$0A2	a $	Shift a → S_4
$0A2a4	$	Reduce $S \to Aa$ (R_1)
$0S1	$	Accept

Step 5: Passing Tree

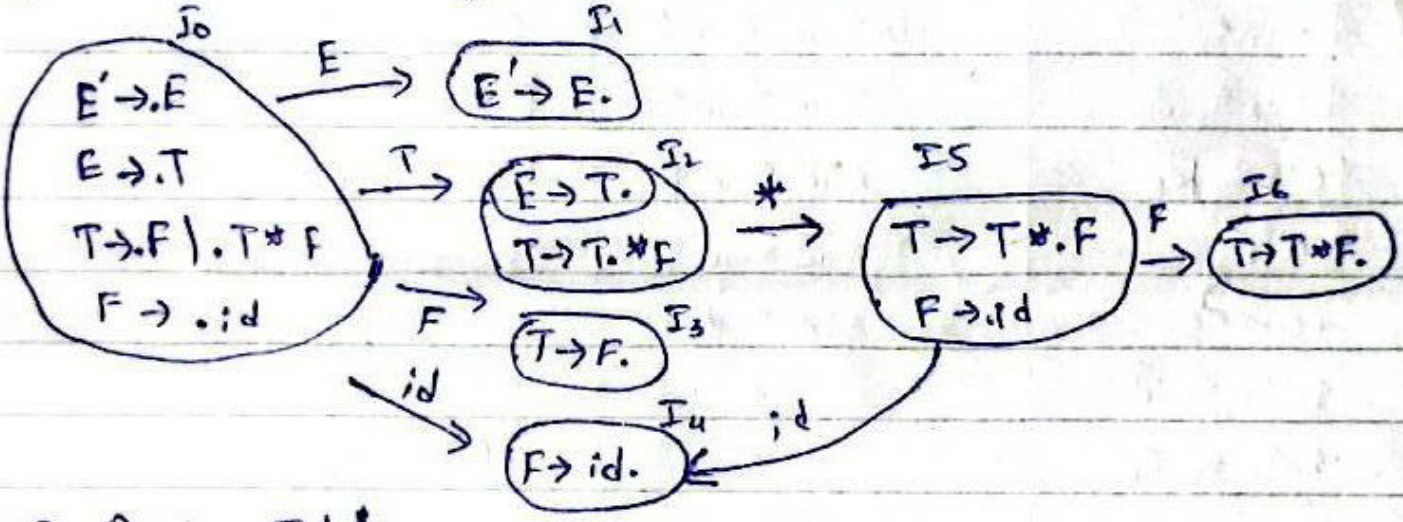


(2) $E \to T$
$T \to F \mid T * F$
$F \to id$
Input: id * id * id * id

Sol Step 1: Augment & Label

$E' \to E$ (0)
$E \to T$ (1)
$T \to F$ (2) $\mid T * F$ (3)
$F \to id$ (4)

Step 2: Draw DFA Diagram



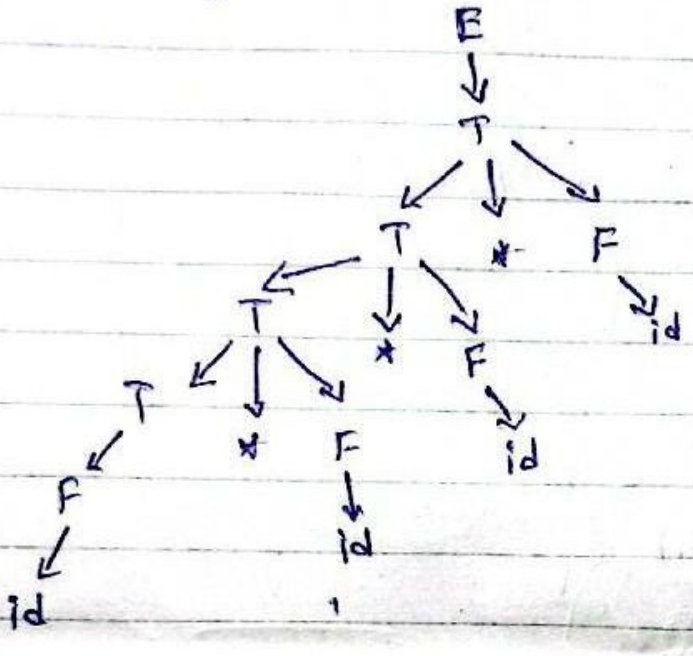
Step 3: Parsing Table

State	Action			Goto		
	*	id	$	E	T	F
I_0		S_4		1	2	3
I_1			accpt			
I_2	S_5	R_1	R_1			
I_3	R_2	R_2	R_2			
I_4	R_4	R_4	R_4			
I_5		S_4				6
I_6	R_3	R_3	R_3			

Step 4: Stack Implementation (Stack Parsing Tree)

Stack (State)	Input Buffers	Actions
$0	id * id * id * id $	Shift id → S_4
$0 id_4	* id * id * id $	Reduce F → id (R_4)
$0 F_3	* id * id * id $	Reduce T → F (R_2)
$0 T_2	* id * id * id $	shift * → S_5
$0 T_2 * $_5$	id * id * id $	Shift id → S_4
$0 T_2 * $_5$ id_4	* id * id $	Reduce F → id (R_4)
$0 T_2 * $_5$ F_6	* id * id $	Reduce T → T*F (R_3)
$0 T_2	* id * id $	shift * → S_5
$0 T_2 * $_5$	id * id $	shift id → S_4
$0 T_2 * $_5$ id_4	* id $	Reduce F → id (R_4)
$0 T_2 * $_5$ F_6	* id $	Reduce T → T*F (R_3)
$0 T_2	* id $	shift * → S_5
$0 T_2 * $_5$	id $	shift id → S_4
$0 T_2 * $_5$ id_4	$	Reduce F → id (R_4)
$0 T_2 * $_5$ F_6	$	Reduce T → T*F (R_3)
$0 T_2	$	Reduce E → T (R_1)
$0 E_1	$	Accept

Step5: Parsing Tree

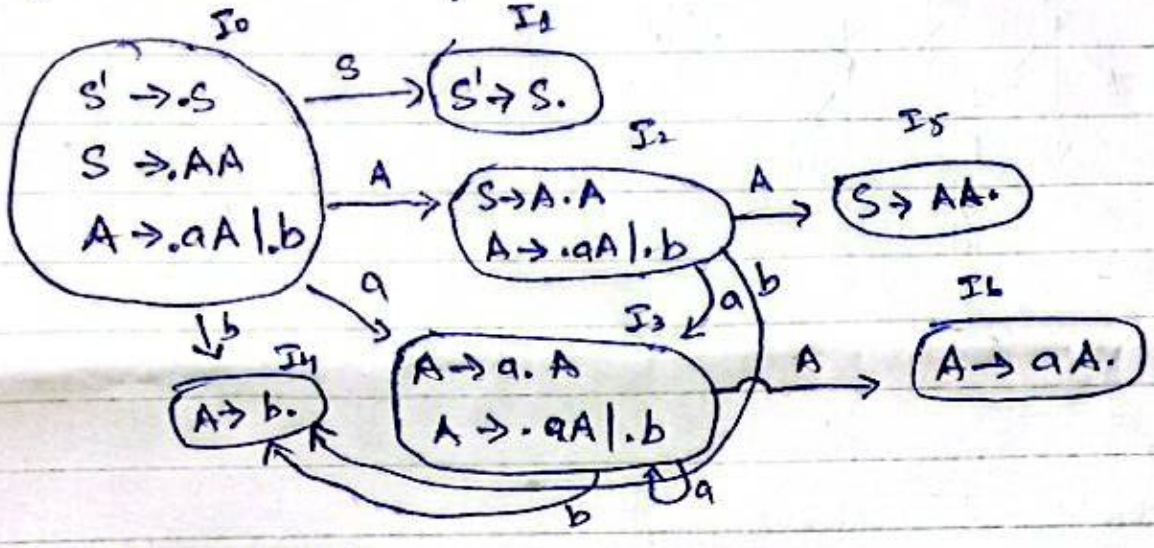


(3) $S \to AA$
$A \to aA \mid b$
Input: aabb

Sol Step 1: Augment & Label

$S' \to S$ (0)
$S \to AA$ (1)
$A \to aA$ (2) $\mid b$ (3)

Step 2: Draw DFA Diagram



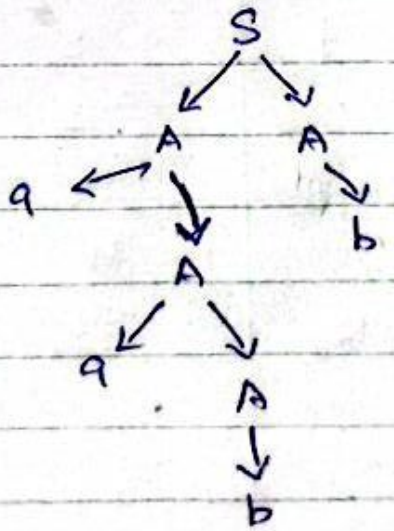
Step 3: Passing Table

State	Action			Goto	
	a	b	$	S	A
I_0	S_3	S_4		1	2
I_1			Accept		
I_2	S_3	S_4			5
I_3	S_3	S_4			6
I_4	R_3	R_3	R_3		
I_5	R_1	R_1	R_1		
I_6	R_2	R_2	R_2		

Step 4: Stack Implementation

Stack	Input (Buffer)	Action
$0	a a b b $	Shift a → S_3
0a_3$	a b b $	Shift a → S_3
0a_3a_3$	b b $	Shift b → S_4
0a_3a_3b_4$	b $	Reduce A → b (R_3)
0a_3a_3A_6$	b $	Reduce A → aA (R_2)
0a_3A_6$	b $	Reduce A → aA (R_2)
$0 A_2	b $	Shift b → S_4
0A_2b_4$	$	Reduce A → b (R_3)
0A_2A_5$	$	Reduce S → AA
0S_1$	$	Accept

Step 5: Parsing Tree



(4) $S \to 01 \mid 0S1$

Input: 000111

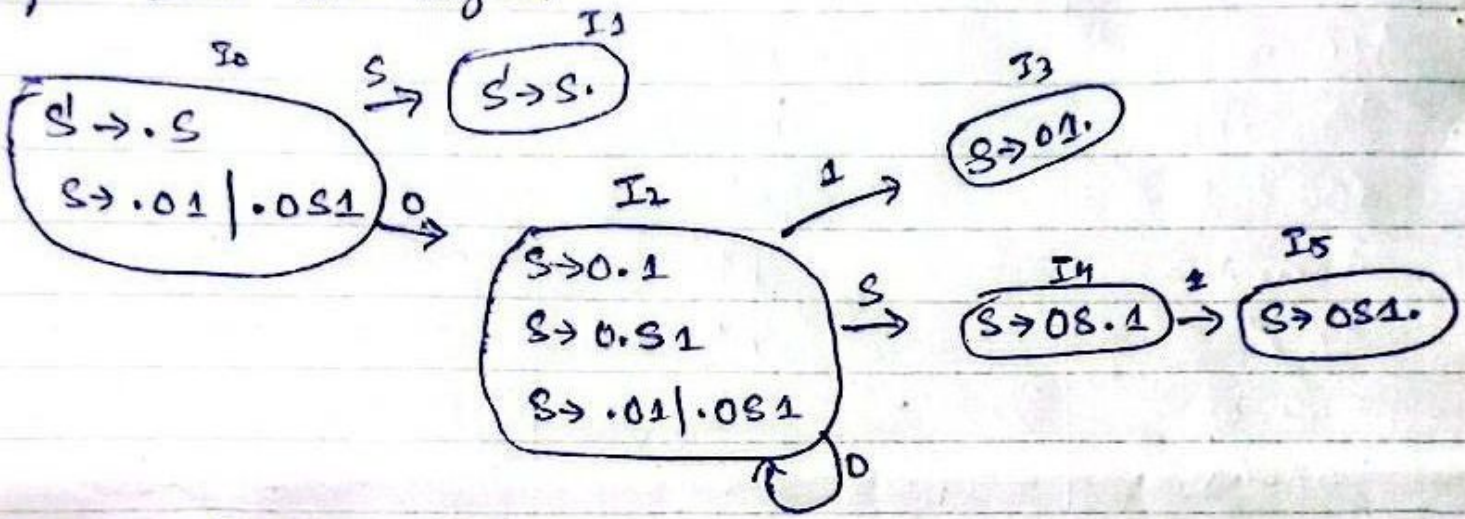
Sol

Step1: Augment & Label

$S' \to S$ (0)

$S \to 01$ (1) $\mid 0S1$ (2)

Step2: Draw DFA Diagram



Step3: Parsing Table

Stack	Action			Goto
	0	1	$	S
I_0	S_2			1
I_1			Accept	
I_2	S_2	S_3		4
I_3	R_1	R_1	R_1	
I_4		S_5		
I_5	R_2	R_2	R_2	

① $S \to XY$

$X \to xX \mid x$

$Y \to yY \mid y$

Input: xxx yyy

Sol Step 1: Augment & Label

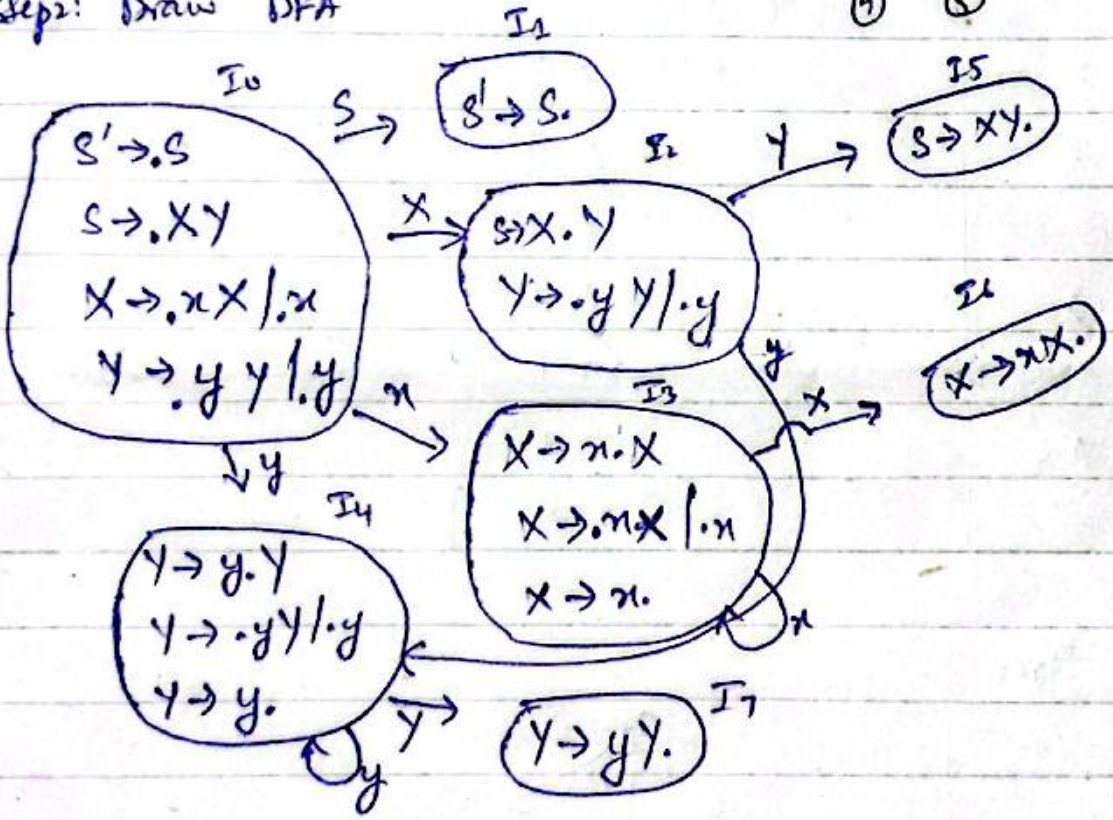
$S' \to S$ ⓪

$S \to XY$ ①

$X \to xX$ ② $\mid x$ ③

$Y \to yY$ ④ $\mid y$ ⑤

Step 2: Draw DFA



Step 3: Parsing Table

State	Action			Goto		
	x	y	$	S	X	Y
I_0	S_3	S_4		1	2	
I_1			Accept			
I_2		S_4				5
I_3	S_3	R_3	R_3		6	
I_4	R_5	S_4	R_5			7
I_5	R_1	R_1	R_1			
I_6	R_2	R_2	R_2			
I_7	R_4	R_4	R_4			

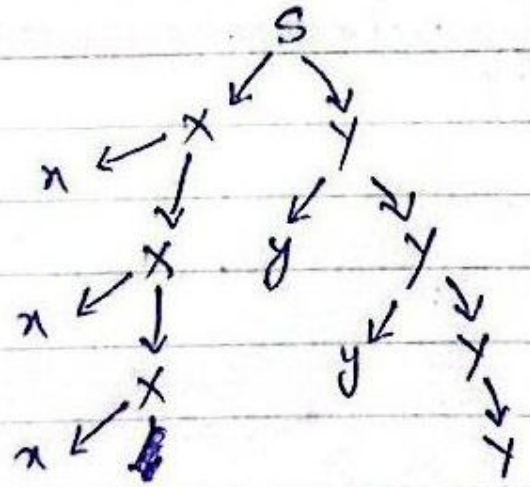
Step 4: Stack Trace

Stack	Input Buffer	Action
$0	x x x y y y $	Shift x → S_3
$0 x 3	x x y y y $	Shift x → S_3
$0 x 3 x 3	x y y y $	Shift x → S_3
$0 x 3 x 3 x 3	y y y $	Reduce X → x (R_3)
$0 x 3 x 3 X 6	y y y $	Reduce X → xX (R_2)
$0 x 3 X 6	y y y $	Reduce X → xX (R_2)
$0 X 2	y y y $	Shift y → S_4
$0 X 2 y 4	y y $	Shift y → S_4

Continued

Stack	Input Buffer	Action
$0 x2 y4	y y $	Shift y → S4
$0 x2 y4 y4	y $	Shift y → S4
$0 x2 y4 y4 y4	$	Reduce Y → y (R3)
$0 x2 y4 y4 Y7	$	Reduce Y → y Y (R4)
$0 x2 y4 Y	$	Reduce Y → y Y (R4)
$0 x2 Y5	$	Reduce S → x Y (R1)
$0 S1	$	Accept

Step 5: Parse Tree



SLR(1) Parser

① $E \to E + T \mid T$

$T \to T * F \mid F$

$F \to id$

Input: id + id * id + id * id

Step1: Augment & Label

$E' \to E$ ⓪

$E \to E + T$ ① $\mid T$ ②

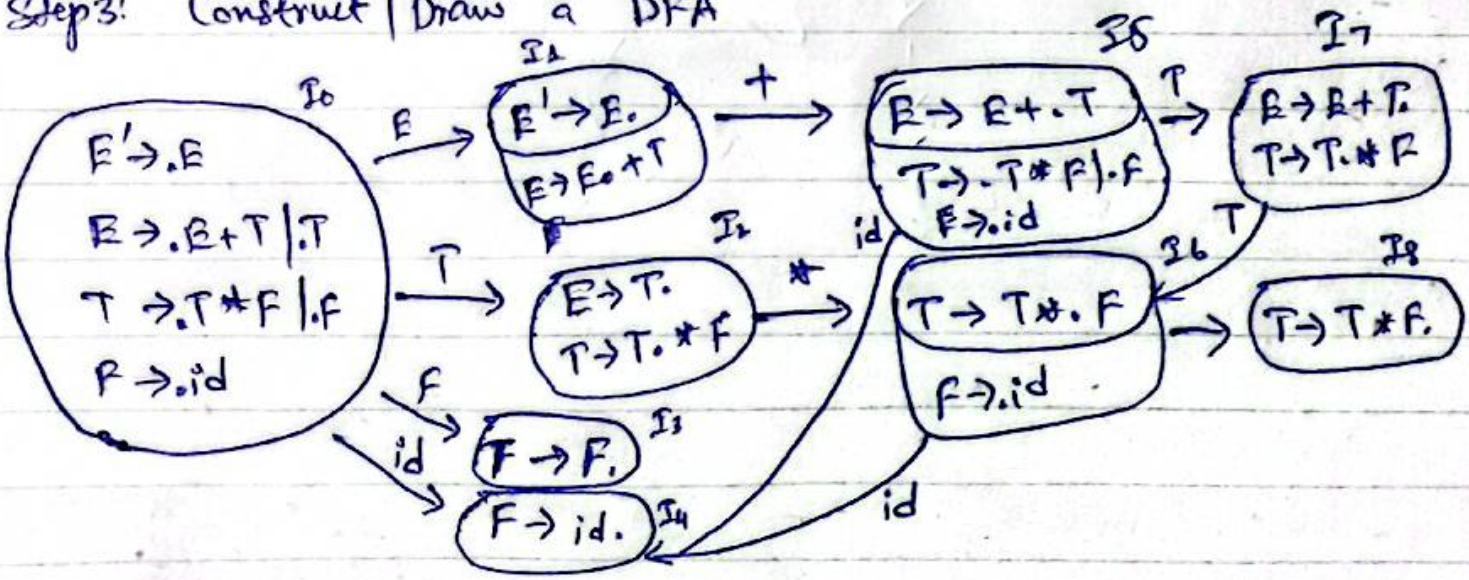
$T \to T * F$ ③ $\mid F$ ④

$F \to id$ ⑤

Step2: First and Follow

CFG	First	Follow
$E' \to E$	id	$
$E \to E+T \mid T$	id	$, +
$T \to T*F \mid F$	id	$, +, *
$F \to id$	id	$, +, *

Step3: Construct / Draw a DFA



Step 4: Parsing Table

State	Action				Goto		
	id	+	*	$	E	F	T
I_0	S_4				1	3	2
I_1		S_5		Accept			
I_2		R_2	S_6	R_2			
I_3		R_4	R_4	R_4			
I_4		R_5	R_5	R_5			
I_5	S_4					3	7
I_6	S_4					8	
I_7		R_1	S_6	R_1			
I_8		R_3	R_3	R_3			

Step 5: Stack Tracing

Stack	Input	Action
$0	id + id * id + id * id	Shift id → S_4
$0 id_4	+ id * id + id * id	Reduce F → id (R_5)
$0 F_3	+ id * id + id * id	Reduce T → F (R_4)
$0 T_2	+ id * id + id * id	Reduce E → T (R_2)
$0 E_1	+ id * id + id * id	Shift + → S_5
$0 $E_1 +_5$	id * id + id * id $	Shift id → S_4
$0 $E_1 +_5 id_4$	* id + id * id $	Reduce F → id (R_5)
$0 $E_1 +_5 F_3$	* id + id * id $	Reduce T → F (R_4)
$0 $E_1 +_5 T_7$	* id + id * id $	Shift * → S_6
$0 $E_1 +_5 T_7 *_6$	id + id * id $	Shift id → S_4
$0 $E_1 +_5 T_7 *_6 id_4$	+ id * id $	Reduce F → id (R_5)
$0 $E_1 +_5 T_7 *_6 F_8$	+ id * id $	Reduce T → T * F
$0 $E_1 +_5 T_7$	+ id * id $	Reduce E → E + T
$0 E_1	+ id * id $	Shift + → S_5
$0 $E_1 +_5$	id * id $	Shift id → S_4
$0 $E_1 +_5 id_4$	* id $	Reduce F → id (R_5)
$0 $E_1 +_5 F_3$	* id $	Reduce T → F (R_4)
$0 $E_1 +_5 T_7$	* id $	Shift * → S_6
$0 $E_1 +_5 T_7 *_6$	id $	Shift id → S_4
$0 $E_1 +_5 T_7 *_6 id_4$	$	Reduce F → id (R_5)
$0 $E_1 +_5 T_7 *_6 F_3$	$	Reduce T → F (R_4)

$0 E2 +5 T7 *6 F8	$	Reduce T→F (R4)
$0 E2 +5 T7 *6 F	$	Reduce T→T*F (R3)
$0 E2 +5 T7	$	Reduce E→E+T (R1)
$0 E2	$	Accept

Step 6: Parse Tree