
Reg.No

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -II

Academic Year 2022-2023(ODD)

Fifth Semester

Computer Science and Engineering

19CSB301 – Automata Theory and Compiler Design

A

Time: 1.5 Hours **Maximum Marks: 50**

Part-A (5 x 2 =10)

		CO	Blooms
1.	Name the rules of computing the Firstpos and Lastpos of concatenation and union operation	CO2	Rem
2.	Construct the Syntax tree for ((a+b)*+(a.c)*) and mark the nullable nodes	CO2	Cre
3.	Define LL (1) grammar and its properties	CO3	Rem
4.	Find the first and follow of the CFG given below: S→ABCD A→a| ε B→b| ε C→c D→d| ε	CO3	App
5.	Define Type checking	CO3	Rem

Part-B (2x13+14=40)

6.	a.	Construct the ε-NFA to DFA for the given regular expression (a|b)*abb	13	CO2	Cre
		or			
	b.	Construct DFA from the given regular expression "ba(a+b)*ab" using Direct Method	13	CO2	Cre
7.	a.	Check whether the given CFG is LL(1) or not. Check whether the string id+id*id is accepted by this CFG E→TE' E'→+TE'| ε T→FT' T'→*FT'| ε F→id|(E)	13	CO3	Ana
		or			

	b.	Construct the SLR parsing table for the following grammar. check whether the string (a) is accepted or not. A → (A)|a	13	CO3	Eva
8.	a.	Construct the DFA from the given Regular Expression (a|b)*abb using Direct Method	14	CO2	Cre
		or			
	b.	Construct the canonical parsing table for the grammar given below. the check whether the string "cdcd " is accepted or not. S->CC C->cC C->d	14	CO3	Eva

Und-Understanding Rem-Remembering App-Applying
Ana-Analysis Cre-Creating Eva-Evaluating

Reg.No | | | | | | | | | | |

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -II

Academic Year 2022-2023(ODD)

Fifth Semester

Computer Science and Engineering

19CSB301 – Automata Theory and Compiler Design

B

Time: 1.5 Hours **Maximum Marks: 50**

Part-A (5 x 2 =10)

		CO	Blooms
1.	Draw the ε-NFA for the regular expression (0|1) (0|1)	CO2	Rem
2.	Construct the Syntax tree for (a | b)*a and find the first pos and last pos of each node	CO2	App
3.	Eliminate immediate left recursion for the following Grammar E -> E+T T->T*F F->(E)|id	CO3	App
4.	Perform left factoring for the grammar. S → iEtS / iEtSeS / a	CO3	APP
5.	Mention the two rules for Type Checking	CO3	Rem

Part-B (2x13+14=40)

6.	a.	Construct the ε-NFA to DFA for the given regular expression (a|b)*a(a|b) using Thompsons Construction	13	CO2	Cre
		or			
	b.	Convert the Regular Expression (a|b)*abb to DFA (Direct Method)	13	CO2	Cre

7.	a.	Construct the predictive parsing table for the following grammar and hence check whether the string (a,a) is accepted or not. S -> (L)|a L-> L,S|S	13	CO3	Ana
		or			
	b.	Construct the CLR parsing table for the following grammar. check whether the string (a) is accepted or not. CFG A → (A)|a	13	CO3	Cre
8.	a.	Construct the DFA from the given Regular Expression (a/b)*a (a/b) using Direct Method	14	CO2	Cre
		or			
	b.	Construct the SLR parsing table for the following grammar and check whether the string "adad" is accepted or not. S->CC C->aC C->d	14	CO3	Ana

Und-Understanding Rem-Remembering App-Applying
Ana-Analysis Cre-Creating Eva-Evaluating

Reg.No

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -II

Academic Year 2022-2023(ODD)

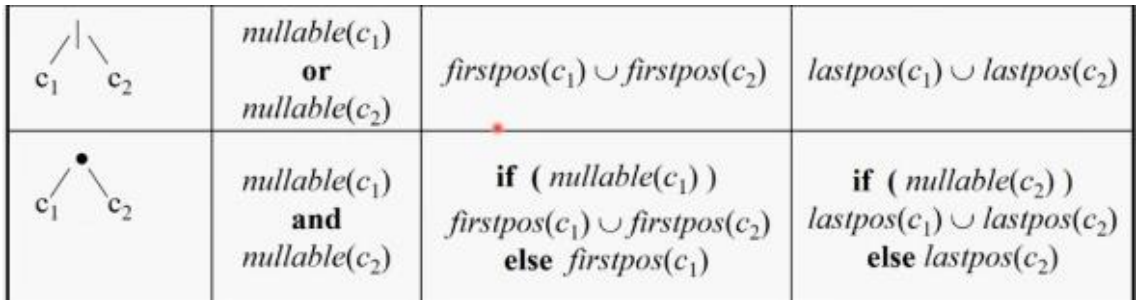
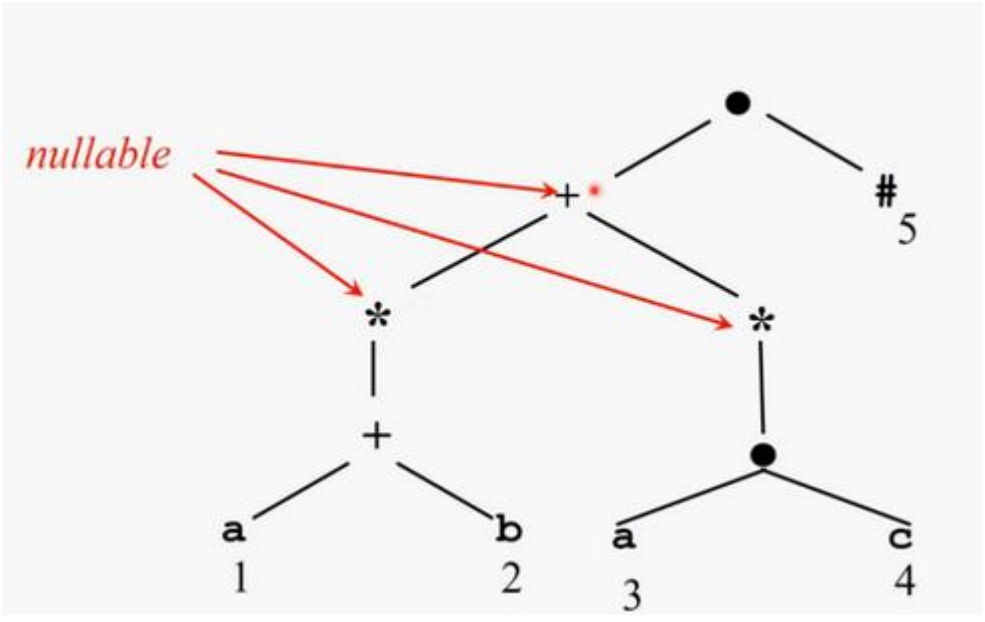
Fifth Semester

Computer Science and Engineering

19CSB301 – Automata Theory and Compiler Design

Time: 1.5 Hours **Maximum Marks: 50**

Part-A (5 x 2 =10)

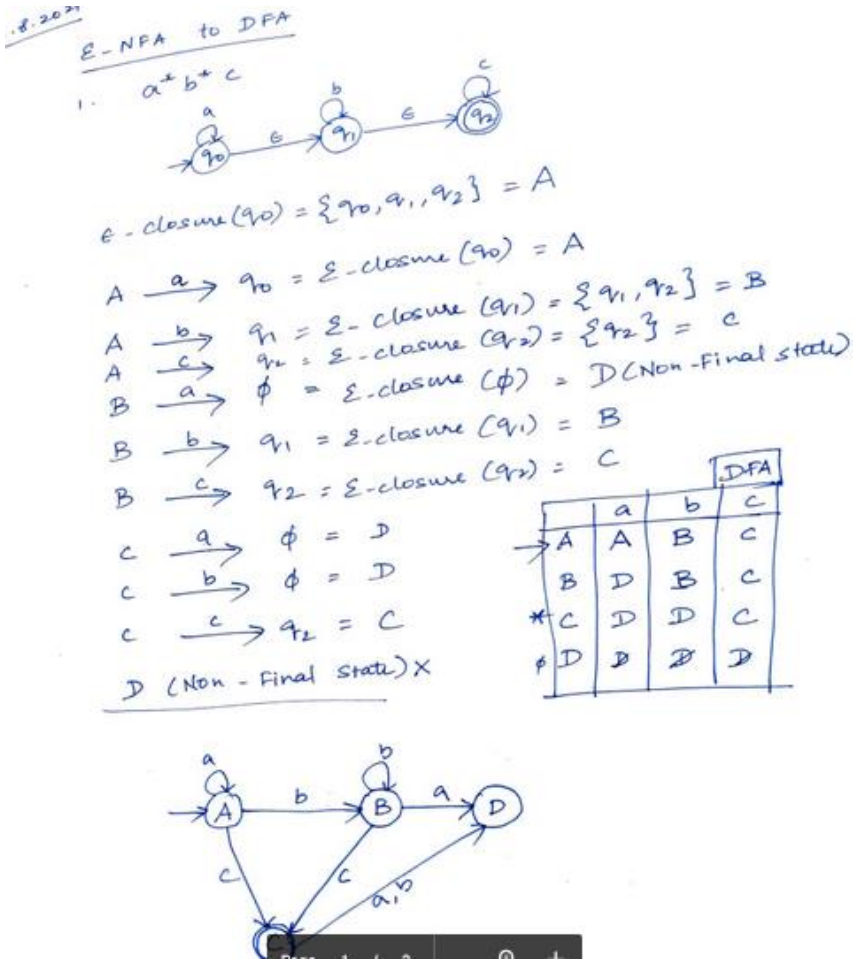
		CO	Blooms
1.	**Give the rules of computing the Firstpos and Lastpos of concatenation and union operation**  	CO2	Rem
2.	**Construct the Syntax tree for ((a+b)*+(a.c)*) and mark the nullable nodes**  	CO2	Und
3.	**Define LL (1) grammar and its properties** LL(1), the first L stands for scanning the input from left to right, the second L stands for producing a leftmost derivation, and the 1 stands for using one input symbol of lookahead at each step to make parsing action decision.	CO3	Und

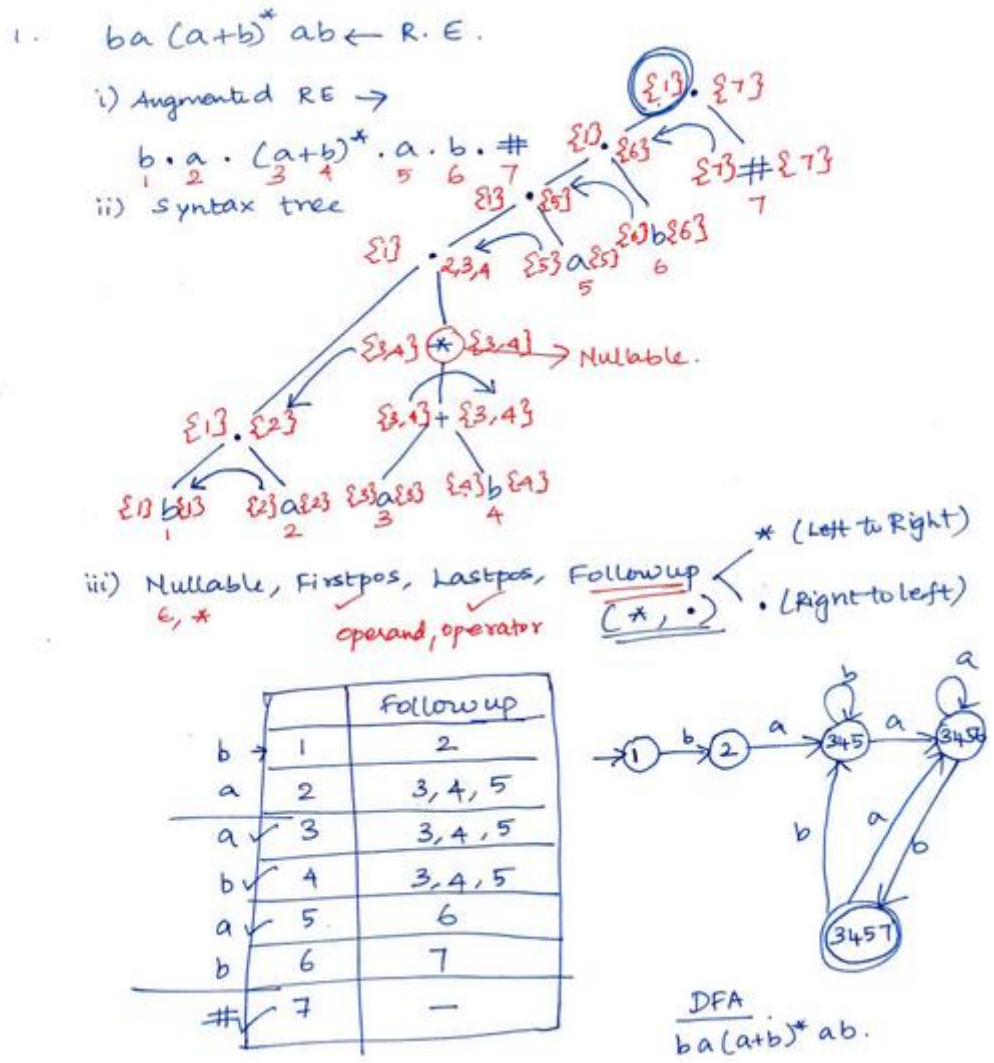
4.	**Find the first and follow of the CFG given below:** S→ABCD A→a| ε B→b| ε C→c D→d| ε	CO3	Ana
5.	**Define Type checking** Type checking is the process of verifying and enforcing constraints of types in values. A compiler must check that the source program should follow the syntactic and semantic conventions of the source language and it should also check the type rules of the language.	CO3	Und

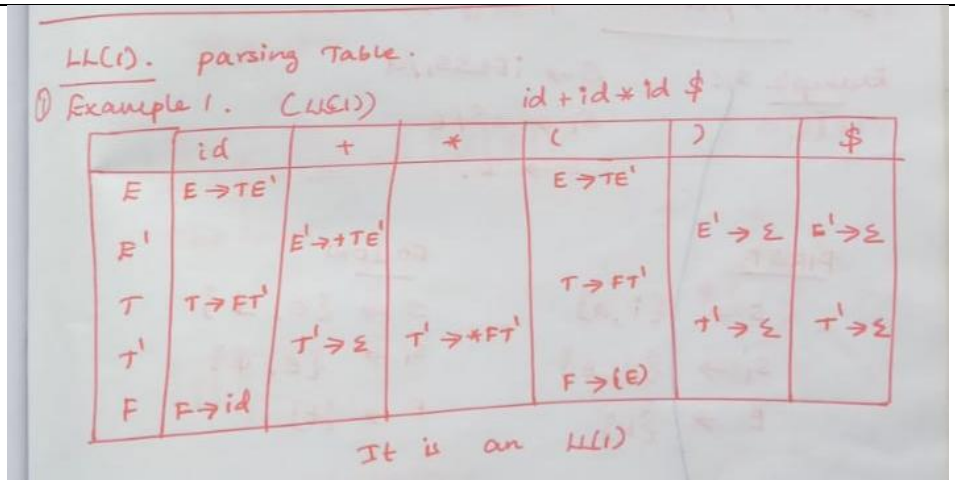
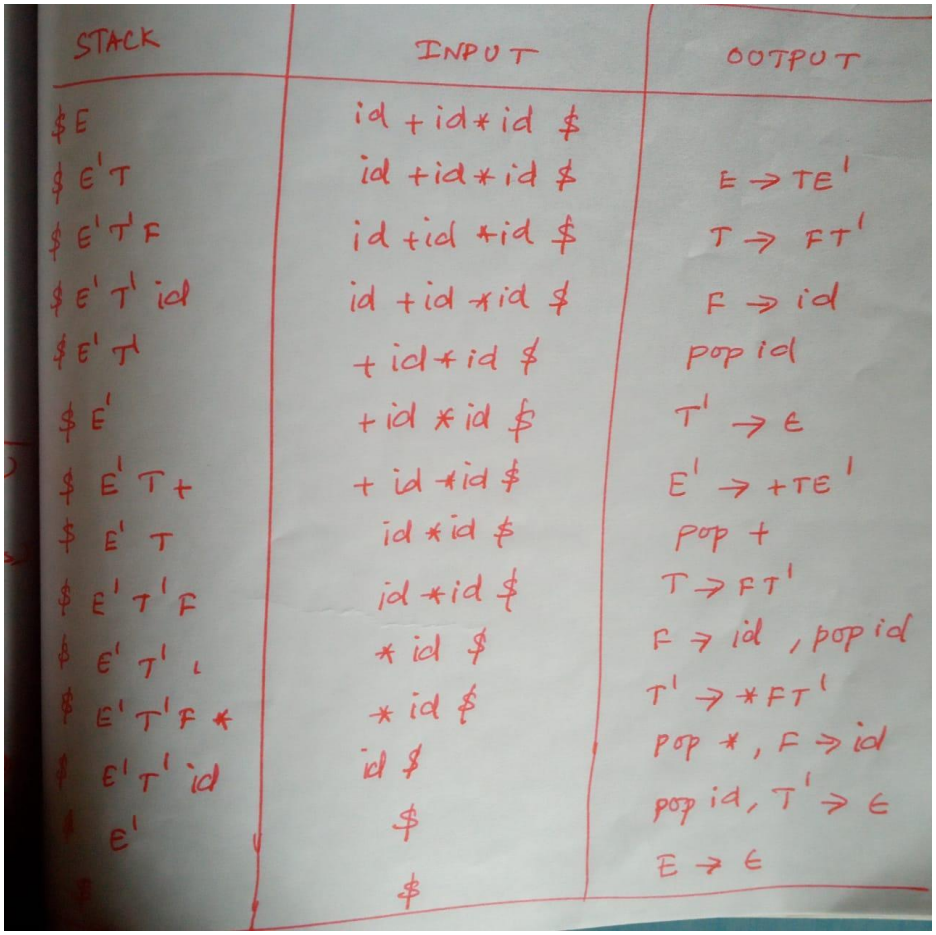
	FIRST	FOLLOW
S→ABCD	{a,b,c}	{$}
A→a| ε	{a, ε}	{b,c}
B→b| ε	{b, ε}	{c}
C→c	{c}	{d,$}
D→d| ε	{d, ε}	{S}

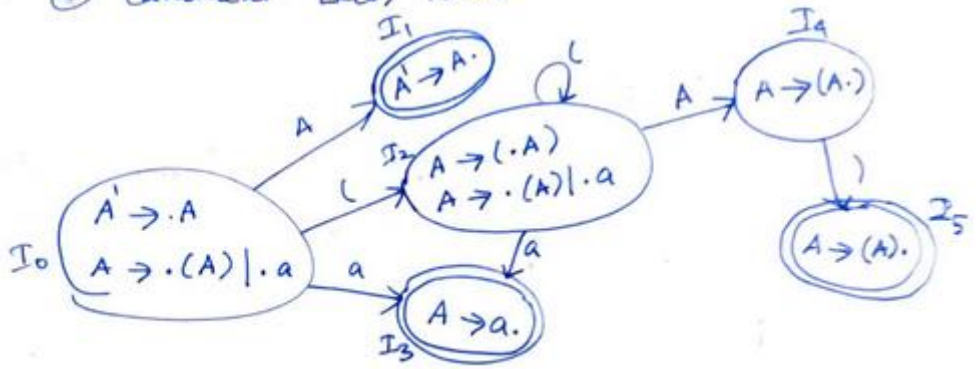
Part-B (2x13+14=40)

6.	a.	Convert the ε-NFA to DFA for the given regular expression (a|b)*abb	13	CO2	Und
		or			
	b.	Convert the given regular expression "ba(a+b)*ab" to DFA (Direct Method)	5	CO2	App



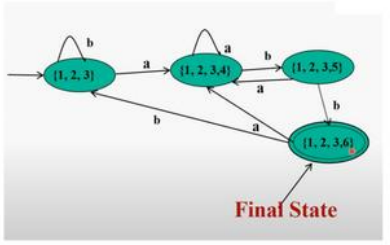
					
			9	CO2	Und
7.	a.	Check whether the given CFG is LL(1) or not. Check whether the string id+id*id is accepted by this CFG E→TE' E'→+TE'| ε T→FT' T'→*FT'| ε F→id|(E)	13	CO3	App

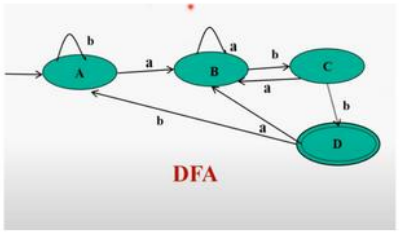
		LL(1). parsing Table. ① Example 1. (LL(1)) id + id * id $   			
		or			
	b.	Construct the SLR parsing table for the following grammar. check whether the string (a) is accepted or not A → (A)|a	13	CO3	Und

		① Augumented Grammar $A' \rightarrow A$ $A \rightarrow (A) \mid a$ ② Canonical LR(0) items  SLR Table Final Items – Reduce operation (Follow of ~~(First of LHS)~~ LHS) $A \rightarrow (A) \mid a$, FIRST {(, a} | FOLLOW {), $} ITEM / ACTION / GOTO a | (|) | $ | A I_0: S_3 | S_2 | | | 1 I_1: | | | Accept | I_2: S_3 | S_2 | | | 4 I_3: | | r_2 | r_2 | I_4: | | S_5 | | I_5: | | r_1 | r_1 |			
8.	a.	Convert the Regular Expression (a|b)*abb to DFA using Direct Method	14	CO2	Ana

DFA Construction

	Node	*followpos*
a	1	1,2,3
b	2	1,2,3
a	3	4
b	4	5
b	5	6
#	6	-





or

b. Construct the canonical parsing table for the grammar given below. the check whether the string "cdcd " is accepted or not. — 14 — CO3 — Eva

S->CC
C->cC
C->d

Step 3.
CLR Parse Table

	ACTION			GOTO	
	a	b	$	A	S
I_0	S_3	S_4		2	1
I_1			Accept		
I_2	S_6	S_7		5	
I_3	S_3	S_4		9	
I_4	r_3	r_3			
I_5			r_1		
I_6	S_6	S_7		8	
I_7			r_3		
I_8			r_2		
I_9	r_2	r_2			

Reg.No

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -II

Academic Year 2022-2023(ODD)

Fifth Semester

Computer Science and Engineering

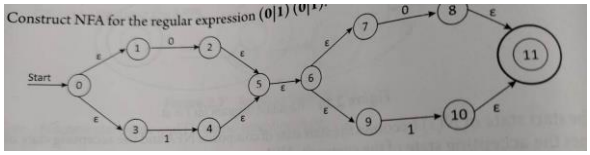
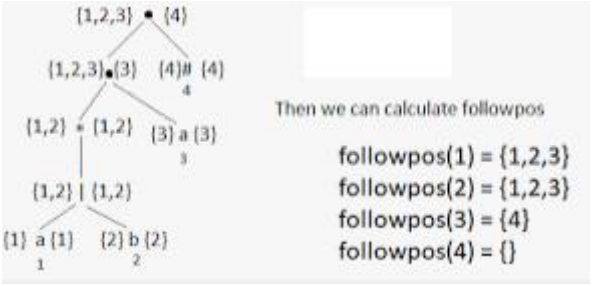
19CSB301 – Automata Theory and Compiler Design

Answer Key

B

Time: 1.5 Hours **Maximum Marks: 50**

Part-A (5 x 2 =10)

		CO	Blooms
1.	Draw the ε-NFA for the regular expression (0|1)(0|1) 	CO 2	Rem
2.	Construct the Syntax tree for (a|b)*a and find the first pos and last pos of each node 	CO 2	App
3.	Eliminate immediate left recursion for the following Grammar E -> E+T T->T*F F->(E)|id Eliminating the immediate left recursion from the productions for E and then for T, we obtain E→ TE' E'→ + TE' | ε T→ FT' T'→ * FT' | ε F→ (E) | id.	CO 3	App

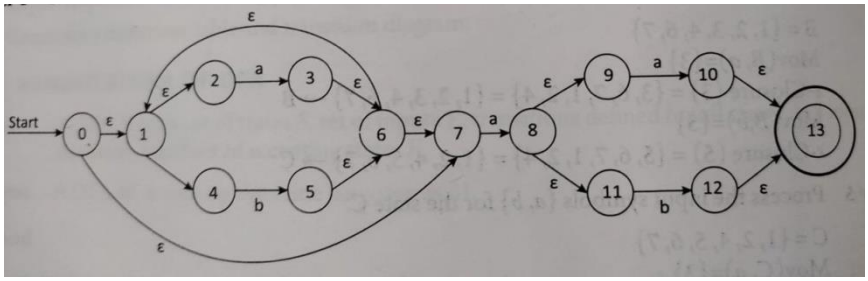
4. Perform left factoring for the grammar. CO 3 APP
S → iEtS / iEtSeS / a

5. Mention the two rules for Type Checking CO 3 Rem

- Type Synthesis
  - – Builds the type of an expression from the types of its subexpressions
  - – Requires names to be declared before usage
- Type inference
  - – determines the type of a construct from the way it is used

Part-B (2x13+14=40)

6. a. Construct the ε-NFA to DFA for the given regular expression (a|b)*a(a|b) using Thompsons Construction 13 CO2



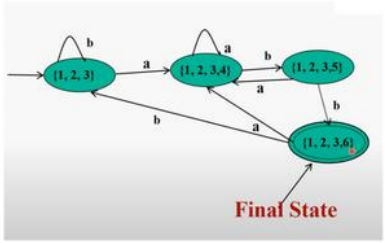
States	Input a	Input b
A	B	C
B	D	E
C	B	C
D	D	E
E	B	C

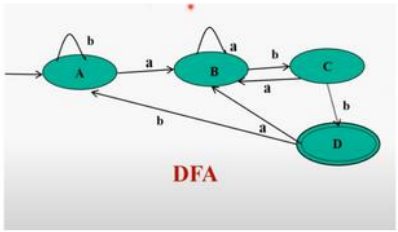
or

b. Convert the Regular Expression (a|b)*abb to DFA (Direct Method) 13 CO2

DFA Construction

	Node	*followpos*
a	1	1,2,3
b	2	1,2,3
a	3	4
b	4	5
b	5	6
#	6	-





7. a. Construct the predictive parsing table for the following grammar and hence check whether the string (a,a) is accepted or not. 13 CO3

S -> (L)|a

L-> L,S|S

Remove left recursion $L \rightarrow S\,L'$

$L' \rightarrow ,\,S\,L' \,/\, \varepsilon$

The grammar

$S \rightarrow (L)$

$S \rightarrow a$

$L \rightarrow S\,L'$

$L' \rightarrow ,\,S\,L' \,/\, \varepsilon$

Find FIRST for all the non-terminals.

Non-terminals = $\{S, L\}$

Terminals = $\{,(,), a\}$

FIRST(S) = $\{(, a\}$

FIRST(L) = $\{$FIRST$(S)\}$ = $\{(, a\}$

FIRST(L') = $\{,, \varepsilon\}$

Find FOLLOW for all the non-terminals.

FOLLOW(S) = $\{S$ followed by non-terminal L', $\$\}$

= $\{$FIRST(L') = $\{\{,, \varepsilon\}$, eliminate ε, find FOLLOW$(L)\}$

= $\{,,), \$\}$

FOLLOW(L)= $\{)\}$

FOLLOW(L')= $\{L'$ does not followed by terminal and non-terminal$\}$

= $\{$FOLLOW$(L)\}$

= $\{)\}$

	(	)	a	,	$
S	$S \rightarrow (L)$		$S \rightarrow a$		
L	$L \rightarrow S\,L'$		$L \rightarrow S\,L'$		
L'		$L' \rightarrow \varepsilon$		$L' \rightarrow ,SL'$	

The above string is accepted

or

b. Construct the CLR parsing table for the following grammar. check whether the string (a) is accepted or not. 13 CO3

CFG A → (A)|a

Step 1:Write the augumented Grammar

Step 2: Compute Closure function

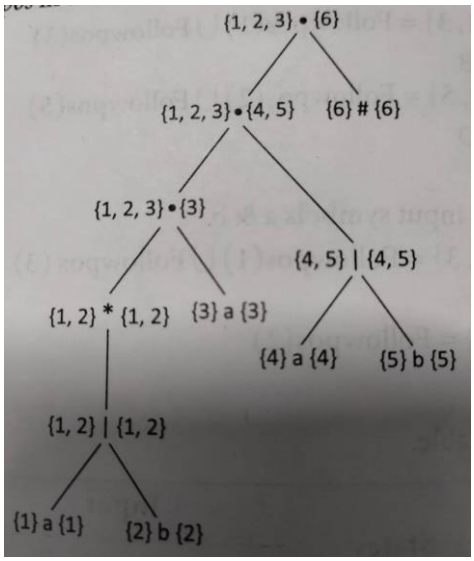
Step3: Find LR(0) items

Step 4:Find Goto function

Step 5:CLR Parsing Table construction

Step 6:Input string Parsing

8. a. Construct the DFA from the given Regular Expression (a/b)*a (a/b) using Direct Method 14 CO2



Nodes	followpos
1	{1, 2, 3}
2	{1, 2, 3}
3	{4, 5}
4	{6}
5	{6}
6	—

or

b. Construct the SLR parsing table for the following grammar and check whether the string "adad" is accepted or not. 14 CO3

S->CC

C->aC

C->d

Step 1 :Find the augumented grammar G

Step2:Find LR(0) Items

Step 3:Constuct Parsing Table

Step 4:Parse the string

States	Action			goto	
	c	*d*	$	S	C
0	*s*3	*s*4		1	2
1			acc		
2	*s*3	*s*4			5
3	*s*3	*s*4			6
4	*r*3	*r*3	*r*3		
5	*r*1	*r*1	*r*1		
6	*r*2	*r*2	*r*2		

Stack	Input string	Action
0	*cdcd* $	Shift 3
0*c*3	*dcd* $	Shift 4
0*c*3*d*4	*cd*$	Reduce *r*3 C → d
0*c*3C6	*cd*$	Reduce *r*2 C → cC
0C2	*cd*$	Shift 3
0C2*c*3	*d*$	Shift 4
0C2*c*3*d*4	$	Reduce *r*3 C → d
0C2*c*3C6	$	Reduce *r*1 S → CC
0C2*c*3C6	$	Reduce *r*2 C → cC
0C2C5	$	Reduce *r*1 S → CC
0S1	$	**Accept**

Und-Understanding Rem-Remembering App-Applying
Ana-Analysis Cre-Creating Eva-Evaluating

Prepared By **Verified By** **HoD**