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

SNS College of Technology, Coimbatore-35
(Autonomous)
B.E/B.Tech- Internal Assessment -II
Academic Year 2023-2024(ODD)

A

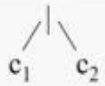
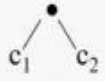
Fifth Semester

Computer Science and Engineering
19CSB301 – Automata Theory and Compiler Design

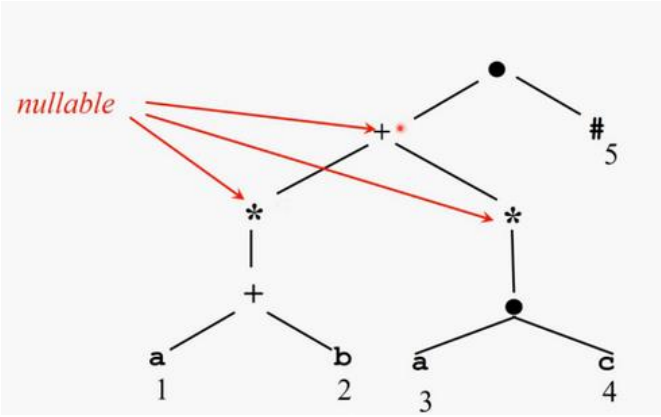
Time: 1.5 Hours **Maximum Marks: 50**

Part-A (5 x 2 =10 Marks)

		CO	Blooms
1.	Tabulate the rules for calculating firstpos and lastpos for "." and "*" Operator	CO2	Rem

	$nullable(c_1)$ **or** $nullable(c_2)$	$firstpos(c_1) \cup firstpos(c_2)$	$lastpos(c_1) \cup lastpos(c_2)$
	$nullable(c_1)$ **and** $nullable(c_2)$	**if** ($nullable(c_1)$) $firstpos(c_1) \cup firstpos(c_2)$ **else** $firstpos(c_1)$	**if** ($nullable(c_2)$) $lastpos(c_1) \cup lastpos(c_2)$ **else** $lastpos(c_2)$

2.	Construct the Syntax tree for ((a+b)*+(a.c)*) and mark the nullable nodes	CO2	App



3.	Define parser and list the types of parser	CO3	Rem

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.

4.	Find the first and follow of the CFG given below:	CO3	App

S→ Bb | Cd
B→aB | ε
C→cC | ε

	FIRST	FOLLOW

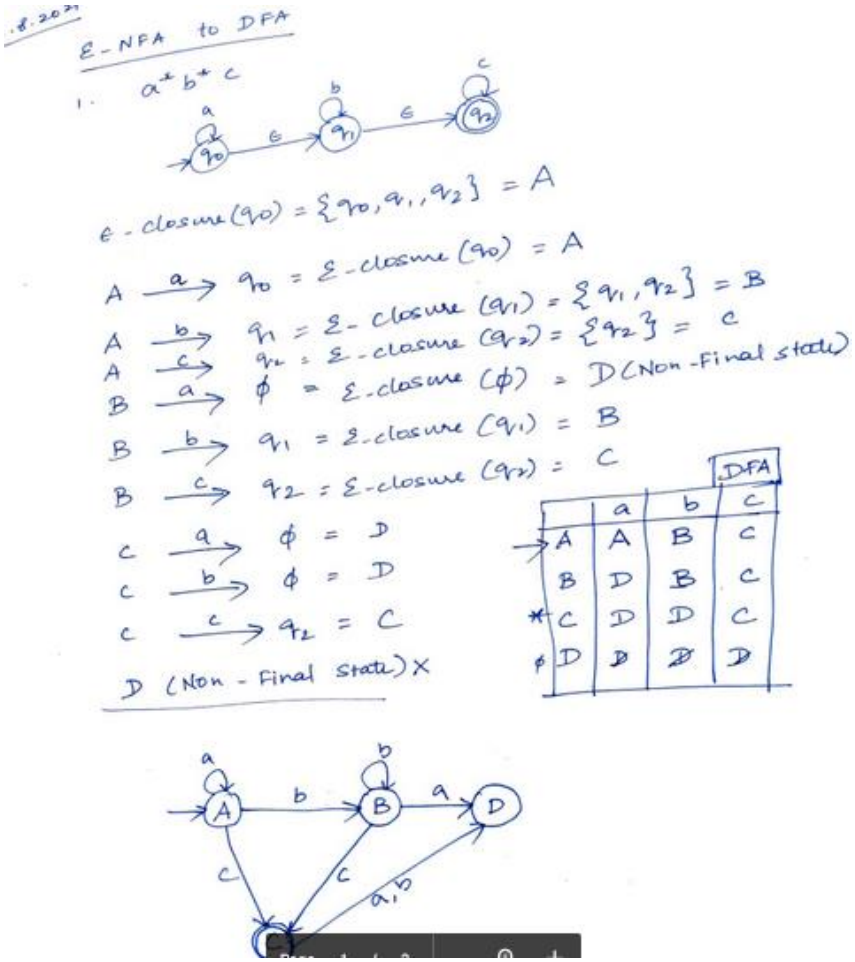
S→ Bb | Cd	{a,b,c,d}	{$}
B→aB | ε	{a, ε }	{b}
C→cC | ε	{a, ε}	{d}

5. Define Type checking CO3 Rem

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

Part-B((2x13)+14=40 Marks)

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

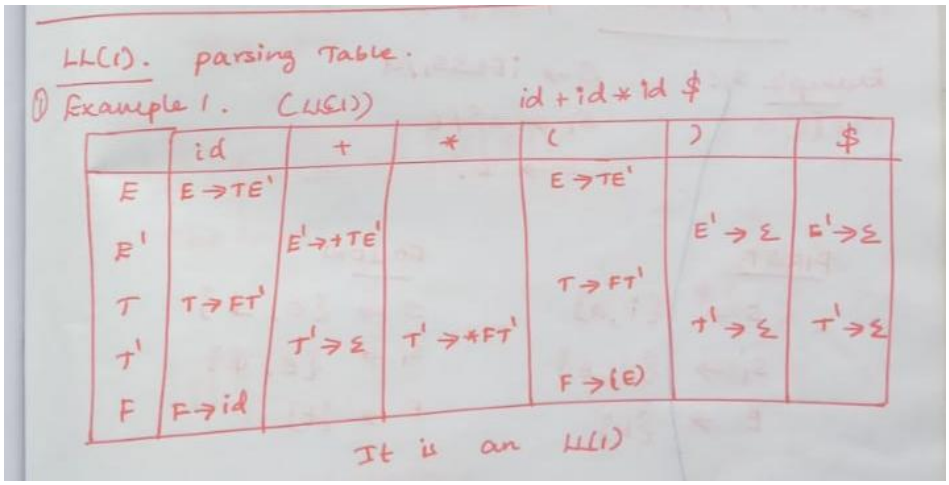


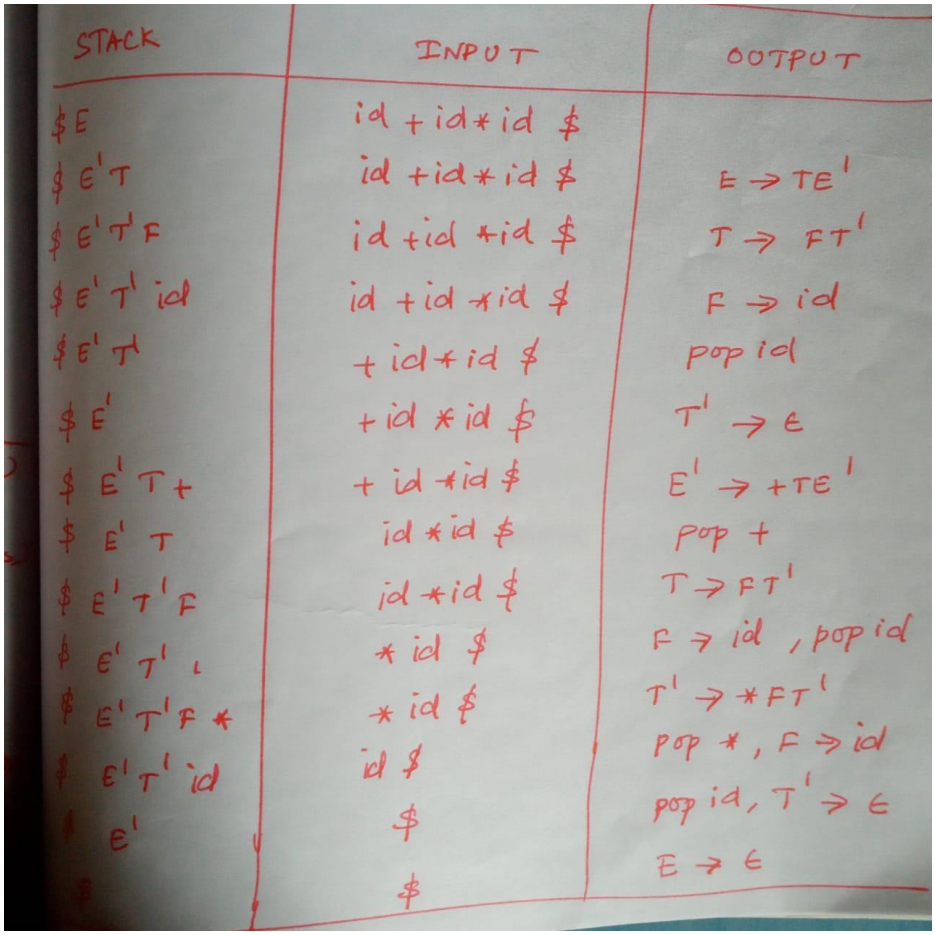
(or)

b. Construct DFA from the given regular expression 13 CO2 App "(a+b)*+(a+c)*"using Direct Method

7. a. Check whether the given CFG is LL(1) or not. Check whether the string id+id*id is accepted by this CFG 13 CO3 Ana

E→TE'
E'→+TE'| ε
T→FT'
T'→*FT'| ε
F→id|(E)





STACK	INPUT	OUTPUT
$E	id + id * id $	
$ E' T	id + id * id $	E → TE'
$ E' T' F	id + id * id $	T → FT'
$ E' T' id	id + id * id $	F → id
$ E' T'	+ id * id $	pop id
$ E'	+ id * id $	T' → ϵ
$ E' T +	+ id * id $	E' → +TE'
$ E' T	id * id $	pop +
$ E' T' F	id * id $	T → FT'
$ E' T'	* id $	F → id , pop id
$ E' T' F *	* id $	T' → *FT'
$ E' T' id	id $	pop * , F → id
$ E'	$	pop id, T' → ϵ
$	$	E → ϵ

(or)

b. Construct the canonical parsing table for the grammar given below. The check whether the string "cdcd" is accepted or not. 13 CO3 App

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

Step 3:
CLR Parse Table

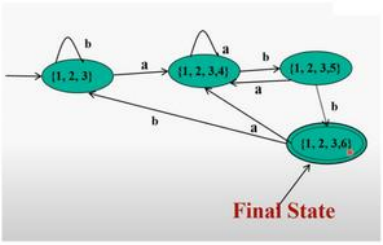
	ACTION			GOTO	
	a	b	$	A	S
I_0	S_3	S_4		2	1
I_1			Accept		
I_2	S_6	S_7		5	
I_3	S_3	S_4		9	
I_4	r_3	r_3			
I_5			r_1		
I_6	S_6	S_7		8	
I_7			r_3		
I_8			r_2		
I_9	r_2	r_2			

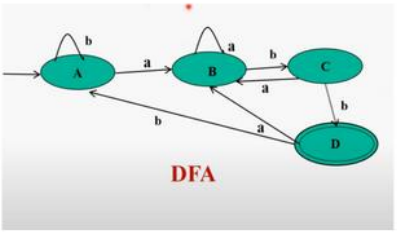
8. a. Construct the DFA from the given Regular Expression (a|b)*abb 14 CO2 App

using Direct Method

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 SLR parsing table for the following grammar. Check whether the string (a) is accepted or not. A → (A)|a — 14 CO3 Ana