Reg. No. : [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Question Paper Code:R5203

B.E./B.Tech. DEGREE EXAMINATION, NOV 2025

Fifth Semester

Computer Science and Engineering

R21UCS503 – THEORY OF COMPUTATION

(Common to CSE (AI & ML) & CI Engineering branches)
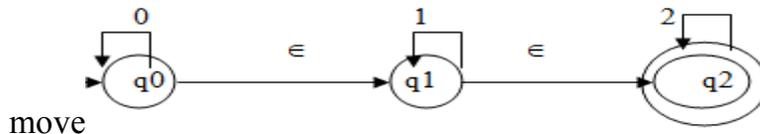
(Regulations R2021)

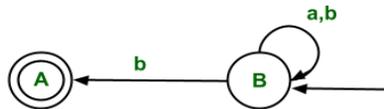Duration: Three hours   Maximum: 100 Marks

Answer ALL Questions

PART A - (10 x 2 = 20 Marks)

1. Obtain ε-closure of q0,q1,q2 in the following NFA with ε-   CO2-App

move

2. Construct DFA with the input symbols {0,1} in which it accepts all string   CO2-App
    ending with 00.

3. Convert the following Finite Automata to Regular Grammar   CO2-App

4. Convert the following Regular grammar into finite automata   CO2-App

    S->0A|1A,A->0A|1A|+B|-B,B->0B|1B|0|1

5. Construct CFG for the language having any number of a's over the input   CO2-App
    symbol {a}.

6. Simplify the following CFG by removing unit Production   CO2-App

    S->A|1,A->01|10

7. Define the formal definition of Linear Bounded automata.   CO1- U

8. Write down the formal definition of Push Down Automata (PDA).   CO1- U

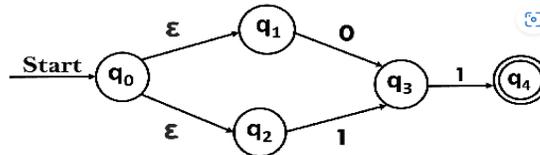9. Give the formal definition of Turing Machine.   CO1- U

10. Differentiate Recursive and Recursively Enumerable Language.   CO1- U

PART – B (5 x 16= 80 Marks)

11. (a) Convert the following NFA to its equivalent DFA   CO2-App   (16)

	a	b
->q1	{q1,q2}	{q2}
*q2	{q3}	{q2}
q3	{q3}	{q3,q4}
*q4	ϕ	ϕ

Or

(b) Convert the following NFA to its equivalent DFA   CO2-App   (16)
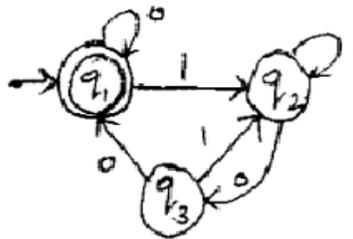

12. (a) Construct the ε-NFA for the given regular expression using   CO2-App   (8)
Thompson's construction $(a+b)^*.ab(a+b)^*$.

Construct   NFA   for   the   following   regular   CO2-App   (8)
expression $(a(aa)^*b+ab^*a)^*$

Or

(b)   Construct Regular Expression for the following DFA.   CO2-App   (16)

13. (a) Simplify the following grammar and find its equivalent in CNF.   CO2-App   (16)
S-> bA | aB ,
A-> bAA | aS | a,
B->aBB | bS | b

Or

(b) Find a Grammar in Chomsky Normal Form equivalent to the   CO2-App   (16)
given Grammar.
S-> aAbB
A ->aA | a
B->bB|b

R5203

14. (a) Construct a PDA accepting by empty stack for the language   CO2-App   (10)
$L=\{a^n b^{2n} \mid n>1)\}$.

     Convert the following CFG to Push Down Automata.   CO2-App   (6)

       S->aDB

       A->aD

       B->b

       D->a

       C>a

Or

(b) Convert the PDA $M=(\{q0,q1\},\{0,1\},\{X,Z0\}, \delta, q0,Z0,\Phi)$ into   CO2-App   (16)
     CFG Grammar. Where δ is defined as

         a. $\delta(q0,0,Z0) = (q1,XZ0)$

         b. $\delta(q0,0,X)=(q1,XX)$

         c. $\delta(q0,1,X) = (q1, \varepsilon)$

         d. $\delta(q1,1,X) = (q1, \varepsilon)$

         e. $\delta(q1, \varepsilon, X) = (q1, \varepsilon)$

         f. $\delta(q1, \varepsilon, Z0) = (q1, \varepsilon)$

15. (a) Describe in detail about programming Techniques for Turing   CO2-App   (16)
     Machines with examples.

Or

(b) Construct Turing Machine for the Language $L=\{a^n b^n c^n\}$ where   CO2-App   (16)
     n>=1.

R5203