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

Question Paper Code: U5203

B.E./B.Tech. DEGREE EXAMINATION, APRIL 2024

Fifth Semester

Computer Science Engineering

21UCS503 THEORY OF COMPUTAION

(Regulation 2021)

Duration: Three hours | Maximum: 100 Marks

Answer ALL Questions

PART A - (10 x 2 = 20 Marks)

1.	Design a DFA to accept strings of a's and b's having even no of a's also check whether the string **w1=abaa** is accepted or not**.**	CO2-App
2.	Obtain a NFA to accept the Language L with input symbol {0, 1} in which it accepts all string ending with 01.	CO2-App
3.	State Pumping Lemma for Regular language and its advantages.	CO1- U
4.	Construct a Regular expression which accepts {0, 1} the set of all binary strings Containing the third symbol from the left end is '1'.	CO2-App
5.	Simplify the following CFG by removing unit Production. S->A|1, A->01|10	CO2-App
6.	List out the applications of Context Free Grammar.	CO1- U
7.	Define Linear Bounded automata	CO1- U
8.	Check whether the given language L= $\{a^n b^n \mid n>=1\}$ is CFL or not.	CO1- U
9.	Construct a Turing Machine which accepts the string W= aba over the input symbol {a,b}	CO2-App
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

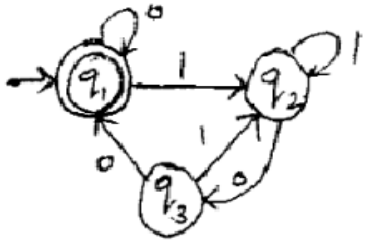
Or

(b) Consider the following ε-NFA. Compute the ε-Closure of each state and find its equivalent DFA. CO2-App (16)

	ε	A	b	c
->p	ϕ	{p}	{q}	{r}
q	{p}	{q}	{r}	ϕ
*r	{q}	{r}	ϕ	{p}

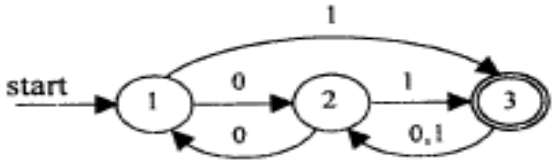
12. (a) (i) Construct an ε-NFA to accept the language indicated by the following regular expression ((01+001)*0*)*. CO2-App (8)

(ii) Construct Regular Expression for the following DFA. CO2-App (8)



Or

(b) (i) Find the regular expression for the following deterministic finite automata using State Elimination Method. CO2-App (8)



(ii) Construct an NFA to accept the language indicated by the following regular expression (0+1)*(00+11) CO2-App (8)

13. (a) (i) Convert the following CFG to Greibach Normal Form CO2-App (10)

S->CA|BB
B->b|SB
C->b
A->a

(ii) Simplify the Following Context Free Grammar. CO2-App (6)

S->AB|CA
B->BC|AB
A->a
C->aB|b

Or

(b) (i) Convert the following CFG to GNF CO2-App (10)

S->AA|0
A->SS|1

(ii) Consider the following productions CO2-App (6)

S->0B|1A
A->0|0S|1AA
B->1|1S|0BB

For the string W=00110101, find a leftmost derivation

14. (a) Convert the PDA M=({q0,q1},{0,1},{X,Z0},δ, q0,Z0,Φ) into Grammar. Where δ is defined as CO2-App (16)

a. δ(q0,0,Z0) = (q1,XZ0)
b. δ(q0,0,X)=(q1,XX)
c. δ(q0,1,X) = (q1, ε)
d. δ(q1,1,X) = (q1, ε)
e. δ(q1, ε, X) = (q1, ε)
f. δ(q1, ε, Z0) = (q1, ε)

Or

(b) (i) Construct a PDA for the given grammar and check the validation of W1=**aa*a0** and W2= **(a0+a).** CO2-App (16)

E-> I | E+E | E*E |(E)
I -> a | Ia | 0 | I0.

(ii) Construct a PDA for the language L={$a^m b^n c^m$ | m,n >=0}

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

(ii) Explain how Turing Machine is coded.

Or

(b) Write short notes on CO2-App (16)

(i) Two way infinite tape TM**.**

(ii) Multiple Tracks Turing Machine.