USN : 2GI20IS011 Course Code : 18CS54 / 16CS52 / 18IS54

Fifth Semester B.E. Semester End Examination, JANUARY_MARCH_2023

FORMAL LANGUAGES AND AUTOMATA THEORY

Time: 3 hrs. Max. Marks :100

Instructions :1. Answer any FIVE full Question selecting at least ONE Question from Each Unit.

MODULE 1

	L	CO	PO	M
1a. Elaborate the formal definition of DFA. Define transition diagram and transition table.	[1]	[1]	[1]	[6]
1b. Design a DFA to accept all the strings with no more than 3 a's continuously. Each string in this language consists of a's and b's.	[3]	[1]	[2]	[7]
1c. Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA) using subset construction.	[3]	[1]	[2]	[7]

δ	a	b
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	Φ	$\{q_2\}$
*q_2	Φ	Φ

OR

2a. Convert the following Є-NFA to its equivalent DFA. [3] [1] [1] [8]

δ	a	b
→ A	B	Φ
B	Φ	C
*C	D	E
*D	D	E
*E	D	E

2b. Indicate the ε-closure for all states of the following. [2] [1] [1] [4]

δ	ε	a	b	c
→p	ϕ	{p}	{q}	{r}
q	{p}	{q}	{r}	ϕ
*r	{q}	{r}	ϕ	{p}

2c. Design an NFA that accepts the language L={wbb | w ϵ Σ(a,b)}. Convert the resultant NFA to DFA . [3] [1] [1] [8]

MODULE 2

3a. Define Regular expression. Obtain a Regular Expression for the following

1. L = $\{a^n b^m \mid n \geq 1, m \geq 1 \text{ and } nm <= 3\}$
2. L = {vuv : v,u ∈ (a, b)* and | v | = 2
3. Set of all strings that do not end with "01" over $\sum = \{0,1\}$

[3] [2] [3] [12]

3b. Define Regular Expression. Apply state elimination method to obtain the Regular Expression

δ	0	1
*q0	q0	q1
*q1	q2	q1
q2	q2	q2

[3] [2] [3] [8]

OR

4a.

Define Distinguishable and Indistinguishable states .Apply Table Filling Algorithm to obtain a minimum state Automata

δ	0	1
→A	B	C
B	C	E
*C	D	C
D	C	F
*E	B	E

[3] [2] [3] [12]

4b. Let R be a Regular Expression. Then Prove that there exists a finite automaton M = (Q, â, d, q0, A) which accepts L(R)..

[2] [2] [2] [8]

MODULE 3

5a. Define Context Free Grammar. Obtain a context free grammar to generate a language L = $\{a^n b^m \mid n \neq m\}$

[3] [3] [2] [6]

5b. Consider the context free grammar with productions.

S →aB | bA
A→aS | bAA | a
B→bS | aBB | b

Write leftmost derivation for the string aaabbabbba using the grammar

[3] [3] [1] [6]

5c. Eliminate unit productions from the grammar

S → AB
A → a
B → C | b
C → D
D → E | bC
E → d | Ab

[2] [3] [1] [8]

OR

6a. Show that the following grammar is ambiguous for the string aabbab

S→ aB|bA
A→ aS|bAA|a
B→ bS|aBB|b

[3] [3] [2] [8]

6b. Define sentential form with an example.

[2] [3] [1] [4]

6c. Eliminate Useless symbols in the grammar.

S → aA | bB
A → aA | a
B → bB
D → ab | Ea
E → aC | d

[3] [3] [2] [8]

MODULE 4

7a. Define PushDown Automata . Design a PDA to accept the following Language by final state L= { na(w) = nb(w) , where n >= 1}

[3] [4] [3] [10]

7b. Define Turing machine. With a neat diagram explain the working model of Turing machine.

[2] [4] [2] [5]

7c. Design a Turing Machine to accept the language containing strings of 0's and 1's ending with 011.

[3] [4] [3] [5]

OR

8a. Design a Turing Machine to accept the following language and show the sequence of moves made by the Turing machine for the string "aaabbb" L={$a^n b^n$, where n>=1}

[3] [4] [3] [10]

8b. Design a PDA to accept the following language L={ $a^n b^n c^m$, where n ,m >=1 }

[3] [4] [3] [10]

MODULE 5

9a. Explain the structure of Lex program with an example.

[2] [5] [1] [7]

9b. Design a lex program to count the number of words.

[3] [5] [1] [7]

9c. Explain yacc parser with an example.

[2] [5] [1] [6]

OR

10a. Summarize lexer parser communication

[2] [5] [1] [5]

10b. Define regular expression. Explain characters that form a regular expression

[2] [5] [1] [8]

10c. Develop a yacc program to recognize a valid arithmetic expression that uses operators +, -, * and /.

[3] [5] [1] [7]