_ ,

==========

Department of Computer Science and Engineering

FLAT UNIT-WISE QUESTION BANK

UNIT-1

1. What is a state and write about few types of states?
2. What is a string? Write about concatenation of two strings?
3. Explain the design of a finite state machine with an example?
4. Explain the advantages of Finite State Machine?
5. Explain the classification of FA?
6. How can you represent FA Mathematically?

UNIT-2

1. Explain about alphabets and strings?
2. Explain the operations on strings?
3. Explain the operations on languages?
4. Explain the operations of Regular Language?
5. Explain the operations of CFG?
6. What is Kleene Closure and Positive Closure?
7. Write about the Mathematical representation of Finite State Machine FSM?

UNIT-3

1. What is a state diagram?
2. Write the design strategy for NFA?
3. Write about indistinguishable and distinguishable states with illustration?
4. What are Automata? Explain classification of Automata?
5. What is a state transition table?
6. What are the components of Finite state Automata? Give examples of Finite state machine?

7. Explain the disadvantages of Finite State Machine?
8. Design a language recognizer which consists of any number of 0's followed by number of 1's followed by number of 2's.
9. Design a finite state machine to find out the residues of 3.
10. State and prove the theorem for equivalence of NFA and DFA.
11. Design NFA to recognize the set of strings such as lab, calb, dabl over an alphabet {a, b, c, d, l} and convert each NFA to equivalent DFA
12. Describe the formal notation for NFA with epsilon closure and the uses of epsilon closure
13. For regular expression (0|1)*011, draw the NFA with ϵ-closures and convert it into NFA.

UNIT-4

1. Explain the closure properties of regular expressions.
2. How to find out equivalence of two DFA's. Explain with an example.
3. What is regular grammar? How to convert left linear grammar into right linear grammars
4. Convert the given regular expression l(l|d)* l over an alphabet {l, d} into NFA. Use extended transitions.
5. State and prove the minimization of DFA with an example.
6. Explain the procedure for the conversion of DFA into regular expression like (1|0)*110(1|0)* over an alphabet {0,1}.
7. Explain the Chomsky hierarchy of languages
8. Differentiate right linear and left linear grammars with an example.
9. Reduce the following DFA where q1 is the start state and q6 is the final state.

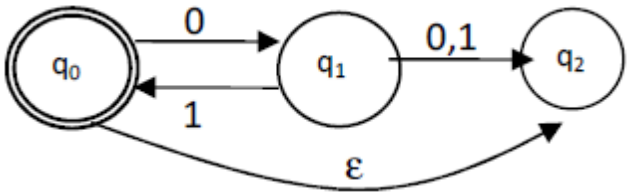
δ	0	1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_6	q_7
q_4	q_4	q_5
q_5	q_6	q_7
q_6	q_4	q_5
q_7	q_6	q_7

10. Construct a regular expression corresponding to the DFA represented by the below transition table. q1 is both the initial state and final state

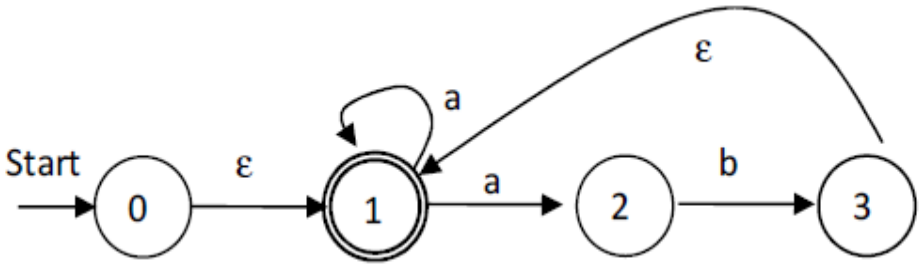
δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_2

UNIT-5

1. What is NFA? Explain the transitions of NFA?
2. Construct an NFA that accepts the set of all strings over {0,1} that start with 0 or 1 and end with 10 or 01.
3. Construct a DFA equivalent to the NFA given below



4. Construct a DFA accepting the language { W ∈ {a,b}* | W has neither aa nor bb as substring}
5. Convert the following NFA-ε to NFA



6. State and prove Arden's theorem.
7. Discuss the properties of Regular Expressions and Regular Languages.
8. Construct a NFA equivalent to the regular expression (10+11)*00.
9. Check wither the following time DFA's are equal or not

	0	1
q_1	q_1	q_2
q_2	q_3	q_1
q_3	q_2	q_3
	0	1
q_4	q_4	q_5
q_5	q_5	q_4
q_6	q_7	q_6
q_7	q_6	q_4

UNIT-6

1. Design a Mealy machine to add two binary numbers of the form x1x2…xk,y1y2…yk?
2. Prove that S -> aSbS | bSaS | e is ambiguous.
3. Design a Turing Machine to accept the language L = {W WR | WÎ(a+b)*}
4. Differentiate Turing Machines and Real Machines?
5. Design a Turing Machine to compute Max(n1, n2)?
6. Explain about Universal Turing Machine?
7. What is Halting Problem of Turing Machine? Is it decidable or not? Explain?
8. What are P and NP class of Languages? What is NP Complete and give examples?
9. Design a Turing Machine "Parity Counter" that outputs 0 or 1, depending on Whether the number of 1's in the input sequence is even or odd respectively.