**THEORY OF COMPUTATION**

**(Computer Science and Engineering)**

**Assignment Questions**

**UNIT - 1**

1. Define deterministic finite automata? Construct DFA for accepts strings over {0, 1} which

contains even number of 0’s and 1’s. Show steps for acceptance: 01010011

and 00010111.

1. Design DFA for accepting strings w such that na(w) mod 3=0 and nb(w) mod 3=2. Verify it

with input string : ababbabb

1. Design DFA that produces residues of modulo 3 of input strings over {0,1} treated as binary

numbers.

1. Construct deterministic finite automata for L, language over {0,1} which accepts all strings

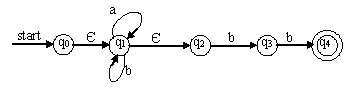
that contain 1 as a third symbol from right-end.

1. Define non-deterministic finite automata? Construct and describe NFA for accepting strings, over input alphabet {a, b}, of the form containing :exactly two a’s, at least two a’s, and at most two a’s

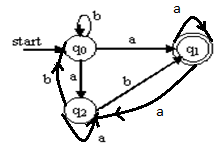
1. Compare and contrast deterministic finite automata and non- deterministic finite automata in detail and present in tabular form.
2. Construct NFA for accepting languages L1= {2nd symbol from right-end is a} and L2 ={2nd symbol from left-end is a } over {a, b}. and describe.
3. Design DFA that produces residues of modulo 4 of input strings over alphabet {0,1,2} treated as base-3 numbers.
4. Design DFA which accepts all strings over ∑ = {0,1} divisible by 6 treated as base-2 numbers.

**UNIT – 2**

1. What is the significance of NFA with ε-transitions? Eliminate ε-transitions from the following ε- NFA, and obtain its equivalent DFA.

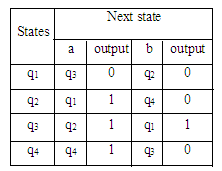


1. Construct NFA for accepting L = {w| w is a string which contains “a” as 3rd symbol from RHS} and convert it ito equivalent DFA?
2. Construct DFA from the following NFA given in transition diagram.

Write acceptance paths for the string: bbabaaaab from the obtained DFA. 

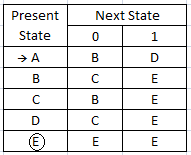
1. Construct a NFA with ε-transitions equivalence to the regular expression R = (0+1)\*(00+11)(0+1)\*, with proper explanation.
2. Differentiate the Mealy and Moore Machines.? Construct moore machine for the mealy

machine given in the following transition table.



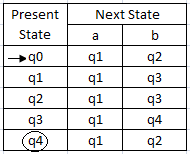
1. Define the Mealy and Moore Machine. Design mealy machine that produces as output the residues of modulo 5 of input strings over ∑={0,1,2} treated as ternary(base -3) numbers.
2. Construct Mealy and Moore machines that takes binary inputs and produce 2’s compliment that input as 0/1. Assume that the input is read from LSB to MSB, and end carry is discarded.
3. Define Mealy and Moore Machine and differentiate them. Design mealy machine that produces as output the residues of modulo 4 of input strings over ∑={0,1,2} treated as ternary(base -3) numbers.

1. Minimize the DFA given in following transition table and its draw equivalent minimum DFA transition diagram.



1. Minimize the DFA given in following transition table and its draw equivalent minimum DFA transition diagram.

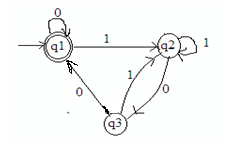
.



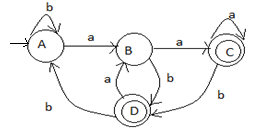
**UNIT – 3**

1. State the Arden’s theorem? Prove it with proper explanation?
2. Define a regular expression and regular set? What are identity rules useful for simplification of regular expressions?
3. Explain the pumping lemma of regular sets? Show that the language L={ anbn | n>=0 } is not regular?

1. Explain the concept of pumping lemma of regular languages.? Show that L={ 0p | p is a prime number } is not regular.
2. What are general steps for construct regular expression from finite automata? Explain? Construct regular expression for the FA given in the following transition diagram.



1. Construct Non-deterministic FA and then deterministic finite automata for the regular expression r = (a + b)\*abb.
2. Construct ε-NFA for the regular expression r = (0+1(0+1)\*+11)\*
3. Construct regular expression for the FA given in the following transition diagram.



**UNIT- 4**

1. What is a regular grammar, left-linear and right-linear grammar. Convert the following regular expression r = (0+1)(0+1(00+11)\*(0+1) into right-linear grammar and left-linear grammar production.

2. Define a regular grammar, left linear and right linear grammar. Convert the regular expression r = (ab(ab+b))(a+b)\*abb into right linear grammar.

3. Design the right-linear and left-linear grammars from the finite automata which accepts the strings over { a, b } that contain equal number a’s and b’s.

4. Design the right-linear and left-linear grammars from the finite automata which accepts the strings w over { a, b } that contain na(w) mod 3 =0 and nb(w) mod 2 =0

5.. Explain about CFG normalization steps? Convert the following Context Free Grammar into Chomsky Normal Form.

S→AaB | aaB | aA | A

A→ε | C | aB

B→bbA | ε | bA

6. Remove all useless, ε (null) and unit productions from the following Context-free grammar and obtain equivalent Chomsky normal form of productions.

S→ AaA| CA| BaB

A→ aaBa| CDA | aa |DC

B→bB | bAB |bb| aS

C→Ca |bc |D

D→ bD |A

7. Explain the grammar simplification process? Convert the following CFG into CNF. S→A0B | 00B | 0A | A

A→ε | C | 0B

B→11A | ε | 1A

8. What is a GNF? Convert the following Context Free Grammar into Chomsky Normal Form. S→AaB | aaB | aA | A

A→ε | C | aB

B→bbA | ε | bA

9. Explain with example, substitution and left factoring methods useful in conversion of CFG

into GNF. Convert the following CFG into Greibach Normal Form.

A1→A2A2 | aA2

A2→A1A1 | b | aA1

10. Write and explain pre-processing steps for converting CFG into GNF. Convert the following

Context Free Grammar into Greibach Normal Form.

S→AA | a A→SS | b | aA.

11. What is a context free grammar? What is ambiguous grammar? Show that the following grammar is ambiguous grammar. E→E+E | E\*E | E-E| (E) | a. where E is the start symbol of the grammar. Write its equivalent unambiguous grammar.

**UNIT – 5**

1. What is a Pushdown Automata? Explain in detail with its structure? Design a PDA acceptance by reaching final state and verify with the suitable input sting for the given CFL L={apbqcm | p,q,m≥1and p+m=q}.

2. Describe the steps for converting CFG productions into PDA transitions. Construct a PDA for accepting following CFG productions.

S→aAB | aA

A→bSA | b | aS

B→aB | bA | a

3. a) Define pushdown automata and explain its behavior with model structure.

b) Design NPDA for accepting CFG L={wwR | w is a string of (0+1)\*} R: Reverse.

verify for acceptance of input 101101.

4. a) Define pushdown automata and explain its behavior with model structure.

b) Design PDA for accepting CFG L={wcwR | w is a string of (0+1)\*} R: Reverse.

verify for acceptance of input 1101c1011.

5. a) Define pushdown automata and explain its behavior with model structure

b) Design PDA acceptance by reaching empty stack for the L={anbm+ncm | m,n≥1} and verify.

6. a) Define pushdown automata and explain its behavior with model structure

b) Design PDA acceptance by reaching empty stack for the L={a3bncn | n≥1}

7. a) Define pushdown automata? What is a PDA acceptance by empty stack?

b) Design PDA acceptance by reaching empty stack for the accepting L={w | w is a string of odd length palindrome} and w is over {0,1}

8. a) What is a Non-deterministic pushdown automata?

b) Design PDA acceptance by reaching empty stack for the accepting L={w | w is a string of evenlength palindrome} and w is over {a,b}

**UNIT – 6**

**Turing Machines (TM) – TMs, REL, Types of TMs,**

**Computability Theory - LBA, CSG, PCP.**

1. Explain about the Turing Machine? Construct Turing Machine that computes addition of two integer numbers.

2. Describe about Turing machine? Design Turing Machine to accept all strings

L={w | w belongs over ∑={ 0,1 }}, where w contains equal number of 0’s and 1’s, that is n0(w)=n1(w) .

3. a) Define Turing Machine?

b) Design a Turing Machine to recognize the language of strings w that contains equal

number of a’s and b’s over (a+b)\*.

4. Explain about Turing Machine? Construct a Turing Machine to recognize the language.

L = {wwR | R: Reverse, w=(a+b)\*}

5. Describe about a Turing Machine and explain its structure. Design a Turing Machine for accepting

L ={ 0n1n2n | n >=1}? Explain?.

6. Design a Turing machine for the language L = {wcw | w belongs to (a+b)+ }.

7. Explain about an instantaneous descriptor of TM? Costruct a Turing Machine that accepts all strings those consisting of equal number of a’s and b’s, also give instantaneous description of input string: bbabaa.

8. Design and explain a Turing Machine to recognize the language.

L = {wcwR | R: Reverse, w=(a+b)\*}

9. Design a Turing Machine to recognize the language and explain?

L = {wcwR | R: Reverse, w=(0+1)\*}

10. What is an instantaneous descriptor of TM? Construct a Turing Machine that accept all the strings consisting of equal number of 0’s and 1’s,

11. a) Define Turing Machine and explain its structure.

b) Discuss about computable functions.? Design a Turing machine that computes

subtraction function, f, where, f(x, y) = x-y, where x >= y.

12. a) Write about computable functions by TM? Design Turing machine for f(n)=(n mod 3).

b) Define TM accept all strings consisting of equal number of 0’s and 1’s.?