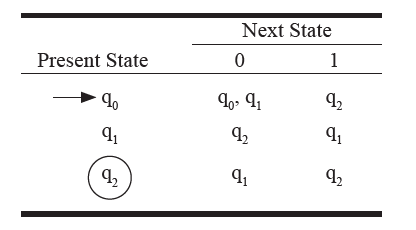
**UNIT – 1**

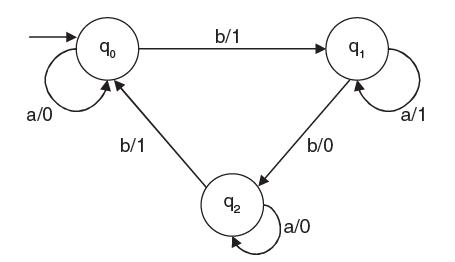
1. a) Define Automata and explain its Applications and characteristics.

b) Compare NFA and DFA

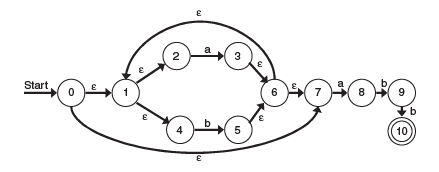
1. Construct a DFA from the given NFA.



1. Convert the following Mealy machine to an equivalent Moore machine



1. Convert the following NFA with epsilon move to an equivalent DFA.



5. a) Design a Moore machine which counts the occurrence of substring aba in a given input string.

b) Design a Mealy machine which determines the residue mod-3 for each binary string treated as a binary integer.

1. Compare and contrast Moore machine and Moore machine

7. Describe the following:

i) Alphabet, String, Language, Empty String.

ii) NFA.

iii) Transition Diagram.

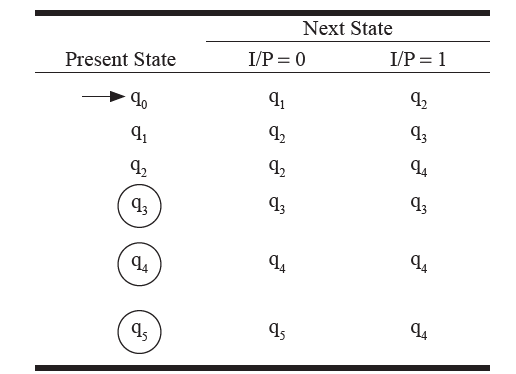
iv) NFA with e (Epsilon) moves

8. Obtain the DFA that accepts/recognizes the language

*L*(*M*) = {*w* | *w*Î {*a*, *b*, *c*}\* and *w* contains the pat tern *abac*}

9. Construct finite automata that accept a string w, where w is binary number divisible by 3.

10. Construct a minimum state automaton from the transitional table given below.



**UNIT – 2**

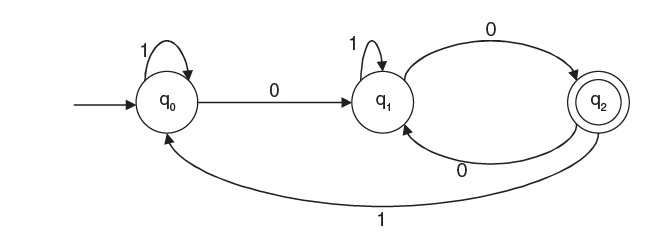
1. Construct an FA equivalent to the RE: L= ab + (aa + bb)(a + b)\* b.
2. Convert the following regular grammar into FA:

S -> aA/bB/a/b

A -> aS/bB/b

B -> aA/bS.

1. Construct an RE from the given FA by the algebraic method using Arden’s Theorem.



1. Show that L = {anbnabn + 1} is not regular.
2. a) Define Regular Expression? Explain about the Properties of Regular Expressions?

b) Construct a DFA for the Regular expression (0+1)\* (00+11) (0+1)\* ?

c)Show that L= {a2n/n<0 } is regular.

6. Construct a CFG for the following

a) Set of string of 0 and 1 where consecutive 0 can occur but no consecutive 1 can occur.

b) Set of all (positive and negative) odd integers.

c) Set of all (positive and negative) even integers.

7. Construct a CFG for the language L = {WCWR | W∈ (a, b)\*}

**UNIT – 3**

1, Remove the null production from the following grammar

**a)** S aAB

A bB

B ∈

**b)** S aA

B b

A Bb

B ->∈

2. Prove that the following grammar is ambiguous.

S -> a/abSb/aAb

A -> bS/aAAb

1. Remove the left recursion from the following grammar.

E ->E + T | T

T -> T\*F | F

F ->id | (E)

1. Check whether the following grammar is ambiguous.

S -> iCtS/iCtSeS/a

C -> b

1. Simplify the following CFG.

S ->AaB/aaB

A -> D

B -> bbA/∈

D ->E

E -> F

F ->aS

1. Construct the string aaabbabbba from the grammar

S aB/bA

A a/aS/bAA

B b/bS/aBB

By using a) Leftmost derivation b) Rightmost derivation.