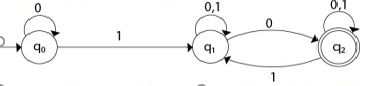
# UNIT-I

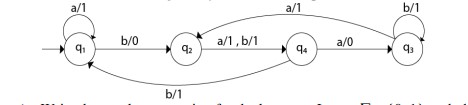
1. List and explain the classifications of Finite Automata. Discuss the applications of it.
2. List the various operations on languages in detail and relate with transition diagrams?
3. Explain the formal definition of an NFA with a suitable example.
4. i) Draw a DFA which accepts strings ending with 11 where the input is {0,1}

ii) Draw a DFA which accepts strings ending with 01 where the input is {0, 1}

1. Demonstrate the mathematical definition of DFA. Design DFA which accepts even number of a’ s and even number of b’s where the input is a, b.
2. Explain the procedure for constructing minimum state DFA with an example.
3. Convert the given NFA to equivalent DFA.

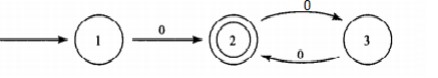


1. Construct a Moore machine that determines whether an input string contains an even or odd number of 1's. The machine should give 1 as output if an even number of 1's are in the string and 0 otherwise.
2. Compare and contrast the features of NFA with DFA. What is the importance of -€ transitions.
3. Convert the following Mealy machine into equivalent Moore machine.



# UNIT-II

1. List and explain the closure properties of Regular grammar.
2. Compute the regular expression for the following machine.

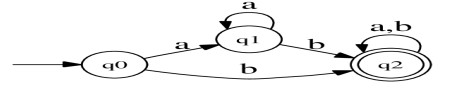


1. Write the regular expression for the language L over ∑ = {0, 1} such that all the strings do not contain the substring 01.
2. Draw the DFA for the following Regular Expressions

i) (01)\* 1(0+1)\*

ii) (ab)\*+ (a+b)\*

1. Explain the Pumping Lemma for regular sets. Show that L={a p | p is a prime} is not regular.
2. Draw the NFA for this regular expression (0/1)\*011 with є-closures and convert it into NFA.
3. Illustrate the Chomsky hierarchy with a neat sketch.
4. What is regular expression? Draw the regular expression for the language over {0, 1}such that set of all strings that contain exactly three 1’s
5. Write the regular expression for the language L over ∑ = {0, 1} such that all the stringsshould have at least one 0 and at least one 1.
6. Derive the regular expression for the following DFA



**UNIT-III**

1. Define Context Free Grammar. State and Explain the closure properties of CFG
2. Discus various steps in signification of context free grammar
3. Design the CFG for the expressions {a n b n where n>=1 }
4. Simplify the following grammar with the following productions

**S ->Aa/B/cA**

**B -> A/bb/E**

**A ->bc/B**

1. Derive the left most and the rightmost derivations for the CFG **S****aAS|a, A****SbA|SS|ba**with the input string aabbaa.
2. Eliminate unit productions and epsilon production from the grammar **S****Aa/B, B****A|bb, A****a|bc|B**
3. Convert the given grammar into CNF.

**S→ASB | aB**

**A→B|S**

**B→b|ε**

1. Convert the grammar into Greibach Normal Form.

**S****CA|BB**

**B****b|SB**

**C****a**

**A****b**

1. Prove that the given Language L={anbm where n>1,m>n} is not regular
2. Design the CFG for the expressions {WCWR , where W belongs to (a+b)+ and WR is the reverse of the string}

**UNIT-IV**

1. Define Push Down Automata. Explain the basic structure of PDA with a neat graphical representation.
2. Describe the components of Push Down Automata and Mention the applications of PDA.
3. Design a PDA for accepting a language {anb2n | n>=1}.
4. Develop a PDA that accepts the strings of the form anb2n where n>1
5. Construct the PDA for the given grammar S--> AA |a, ASA|b
6. Construct a PDA for the following grammar

S->AA/a

A->SA/b

1. Discuss various steps for converting CFG to PDA with your own example.
2. Is a push-down automaton with two stacks equivalent to a turning machine? Justify your answer with proper explanation.
3. Construct a PDA from the following CFG.

**G = ({S, X}, {a, b}, P, S) where the productions are given below.**

**S → XS |**  **,**

**A → aXb | Ab | ab**

1. Discuss the notation and applications of two stack push down automata

**UNIT-V**

1. Explain the Types of Turing machine.
2. Define the TM with formal notations. Explain the concept of Universal Turing Machine.
3. Design Turing machine and its transition diagram to accept the language L = {a n b n | n >=1}
4. Design a Turing Machine to accept the language L={a n b n c nd n/n>=1}
5. Explain the concepts NP-Hard and NP-complete with examples.

6. Construct the Turing machine that computes subtraction, where the fist operand

length is more than the second operand. X is a symbol that separates the two

1. operands.
2. Discuss the decidable and undecidable problems with examples.
3. Design TM which accepts strings ending with 111 where the input is taken from {0,1}
4. List the elements of TM’s and give the block diagram of TM.
5. Construct a TM that computes a function f(m, n) = m+ni.e, addition of two numbers.