Question Bank

Name of Subject: Automata Theory
Subject Code: PC501IT

UNIT I

Automata: Introduction to Finite Automata, Central Concepts of Automata Theory.
Finite Automata: An informal picture of Finite Automata, Deterministic Finite Automata, non-deterministic Finite Automata, An Application, Finite Automata with Epsilon Transitions

Short answer questions

1. Differentiate the + closure and * closure.
2. Elaborate alphabet, string, powers of an alphabet and concatenation of strings.
3. Elaborate language and Grammar and give an example.
4. Explain transition table and transition graph?
5. Give the DFA accepting the language over the alphabet 0, 1 that has the set of all strings beginning with 101.
6. Give the DFA accepting the language over the alphabet 0,1 that have the set of all strings that either begins or end(or both) with 01.
7. Define NFA,DFA, ε-NFA
8. Difference between DFA and NFA.
9. Write the notations of DFA.
10. is it true that the language accepted by any NFA is different from the regular language? Justify your Answer.
11. Construct a finite automaton for the regular expression (0+1)*
12. Illustrate if L be a set accepted by an NFA then there exists a DFA that accepts L.

ESSAY questions

1. Obtain DFAs to accept strings of a's and b's having exactly one a.
2. Obtain a DFA to accept strings of a's and b's having even number of a's and b's.
3. Give Applications of Finite Automata.
4. Define DFA, NFA & Language?
5. Obtain a DFA to accept strings of a's and b's starting with the string ab.
6. Draw a DFA to accept string of 0's and 1's ending with the string 011.
7. Write DFA to accept strings of 0's, 1's & 2's beginning with a 0 followed by odd number of 1's and ending with a 2.
8. Design a DFA to accept string of 0's & 1's when interpreted as binary numbers would be multiple of 3.
9. Find closure of each state and give the set of all strings of length 3 or less accepted by automaton
10. Obtain a DFA to accept strings of a's and b's having a sub string aa.

δ		a	b
□ p	{r	{q}	{p,r
q	} I	{p}	} I
*r	{p,q}	{r}	{p}

11.

Convert above automaton to a DFA

12.Convert following NFA to DFA using subset construction method**.**

δN	0	1
□ p	{p,r}	{q}
q	{r,s}	{p}
*r	{p,s}	{r}
*s	{q,r}	I

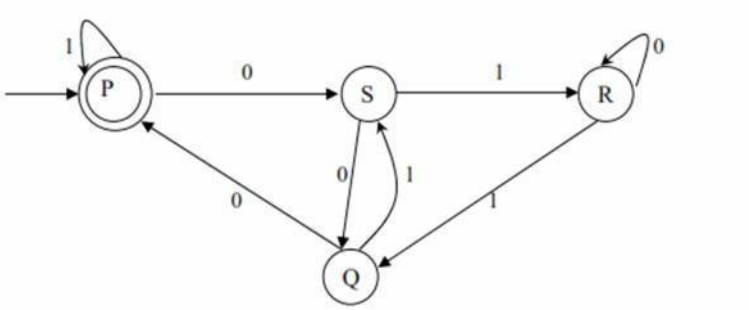
UNIT II

Regular Expression And languages: Regular Expressions, Finite Automata and Regular Expression, Applications of Regular Expressions, Algebraic Laws for Regular Expression. Properties of Regular Languages: Proving Languages not to be Regular, Closure Properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

1. Explain Regular Expression.
2. Explain some operators of Regular Expressions
3. State pumping lemma for regular languages
4. Classify some applications of the pumping lemma.
5. Elaborate Epsilon –Closures.
6. State minimization of DFA.
7. Illustrate a regular expression for the set of all the strings
8. List out applications of regular expressions

1. Write Regular expression for the following L = { $a^n b^m$: m, n are even} L = { a^n, b^m : m>=2, n>=2}.

2. Convert the following DFA to Regular Expression



1. Prove pumping lemma?
2. Prove that L={w|w is a palindrome on {a,b}*} is not regular. i.e., L={aabaa, aba, abbbba,…}
3. Prove that L={ all strings of 1's whose length is prime} is not regular. i.e., L={1^2,1^3,1^5 ,1^7 ,1^{11}, }

4. What is the language accepted by the following FA.

5. Write short note on Applications of Regular Expressions
6. Show that following languages are not regular

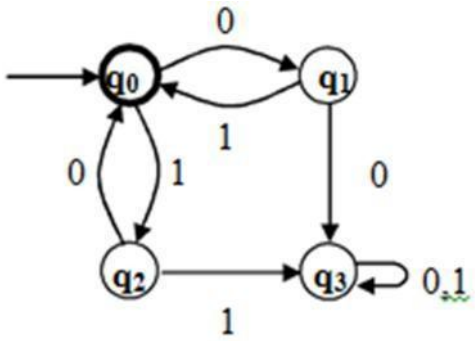
7. L={$a^n b^m$ | n, m t0 and n<m }
   L={$a^n b^m$ | n, m t0 and n>m }
   L={$a^n b^m c^m d^n$ | n, m t1 }
   L={a^n | n is a perfect square }
   L={a^n | n is a perfect cube }

8. Apply pumping lemma to following languages and understand why we cannot complete proof

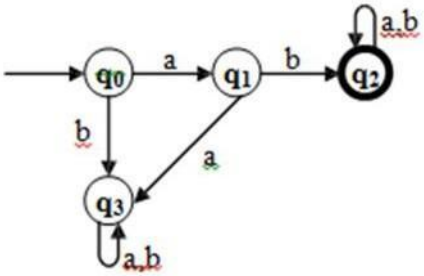
L={anaba| n t0 } L={anbm | n, mt0 }

9. Obtain a DFA to accept strings of a's and b's starting with the string ab **(**

10. Obtain a regular expression for the FA shown below**:**



11. Solve:



12. Explain Closure properties with an example.

UNIT III

Context Free Grammars and Languages: Context-Free Grammars, Parse Trees, Applications,Ambiguity in Grammars and Language Properties of Context Free Languages: Normal Forms for Context-Free Grammars, Pumping Lemma, Closure Properties, Decision Properties of CFL's.

1. What is the purpose of normalization? Construct the CNF and GNF for the following grammar and explain the steps.

   S→aAa | bBb |€

   A→C|a

B→C | b

C→CDE | €

D→A | B | ab

2.Constuct a CFG for the regular expression (011+1)(01)
3.Construct a CFG over {a,b} generating a language consisting of equal number of a's and b's.
4.Is the grammar below ambiguous S→ SS | (S) | S(S)S | E?
5.Convert the following grammar into an equivalent one with no unit productions and no useless symbols
S→ABA A→aAA|aBC|bB B→ A|bB|Cb C→CC|Cc
6.Consider the following grammar G with productions
S→ ABC | BaB
A→aA|BaC|aaa
B→bBb|a
C→CA|AC
Give a CFG with no useless variables that generates the same language.

7.Write down the context free grammar for the language L = { anbn |n>=1}

8.Consider the following grammar for list structures:
S→a|^|(T)
T→T,S|S
Find left most derivation, rightmost derivation and parse tree for (((a,a),^(a)),a)
8.Is the following grammar is ambiguous? Justify your answer.
1. E→ E+E |E*E | id
2. E→ E+E|E*E|(E)|a
9.Find the context free languages for the following grammars.
1. S→aSbS|bSaS| ε
2. S→aSb|ab
3. S→aSb|aAb, A→bAa, A→ba
10.Find CFG for the following languages.
1. L = { ai bj ak | j > i + k }
2. L = { ai bj ck | i = j or j = k }
11.Expand Chomsky Classification of language taking suitable example of each classification.
12. Differentiate between leftmost and rightmost derivations.
13.Discuss Normal forms-Chomsky and Greibach Normal forms with example.
14.Explain Derivation Tree, Expression Tree and Ambiguity with Example

UNIT IV

Push down Automata: Definition, Language of PDA, Equivalence of PDA's and; CFG's, Deterministic Pushdown Automata.

1.Discuss Normal forms-Chomsky and Greibach Normal forms with example.
2.Discuss about PDA acceptance (1) From empty Stack to final state. (2) From Final state to Empty Stack
3.What is PDA? What are its closure properties? Draw a PDA that accepts {0n1n |n>=0}
4.Construct PDA for the language L = {wwR| W in (a+b)*)
5. Explain the definition of a non-deterministic push down automata (ndpa). Construct pda A accepting L={wcwT | w ∈ {a,b}*} by final state.
6.Design PDA for odd number of palindromes

7.Give transition table for deterministic PDA recognizing the following language. { ai bj ck | i, j, k ≥ 0 and j = i or j = k }
8.Explain in Brief: (i) GNF. (ii)Chomsky Normal Form(CNF).
9. Construct a PDA which recognizes all strings that contain equal number of 0's and 1's.

10..PDA is more powerful than a finite automaton. Justify this statement.

11.Explain the informal introduction and formal definition of PDA.
12.Define Instantaneous description (ID) in PDA.
13.Explain about the graphical notation of PDA.
14.Write the process for convert PDA into an equivalent CFG.
15.Define push down automata? Explain acceptance of PDA with empty stack.
16.Explain Deterministic Push down Automata with example?

UNIT V

Turning Machines: Problems that Computer Cannot Solve, The Turning Machine,
Programming Techniques for Turning Machines, Extensions to the Turning Machines, Restricted Turning Machines, Turning Machine and Computers. Undecidable Problems about Turning Machines, Post's Correspondence Problem, Other Undecidable Problems.

1 Briefly explain the different types of Turing machines.
2. Explain how a TM can be used to determine the given number is prime or not.
3. Prove that Halting problem is undecidable.
4. What are the various models of Turing Machine?
5. With an example explain the universal Turing machine
6.Difference between Finite Automata and Turing Machines.
Explain in Brief: (i) Halting Problem.
7.Define Turing Machine. Describe its capabilities. Also write short notes on Universal Turing Machine.

1.Explain Model of Turing Machine in detail
2.What are the differences between a Finite automata and a Turing machine?
3. Design a Turing Machine for L= {a n b n c n}
4.Design a TM to accept the language L={0n1n | n>=1}
5.Construct a Turing machine which multiplies two unary numbers.
6.Construct a Turing machine that recognizes the language L={anbn, n>1}. Show an ID for the string 'aabb' with tape symbols.
7.Explain conversion of regular Expression to TM with example.
8.Explain the various types of Turing machine.
9.Explain Universal turing machine
10.Explain in detail about variations of the TM?
11.Construct a Turing machine that recognizes the language anbncn.
12. Define PCP. Verify whether the following lists have a PCP solution.