**Unit I: Basic Concepts and Formal Language theory**

1. Prove by mathematical induction:
2. 20 +2 1 +2 2 +···+2n =2 n+1 −1 for all integers n ≥0.
3. For all n>=1,  = 
4. Define:  Finite Automaton (FA)

Transition diagram,

String,

Symbol,

Alphabet,

Regular language,

Regular Expression,

1. What are the applications of automata theory?
2. Define FA and NDFA
3. What is Є­closure of any state say q1
4. Differentiate between NFA and DFA.
5. Differentiate  L\*  and L+
6. For given relations R1 : {(1, 2), (2,3), (2,4)} and R2: {(1 ,2), (2,3), (2,4), Find R\*.
7. Prove that “Every NFA has its equivalent FA”
8. For given relations R1= { (a,b), (b,c), (b,d) } and R2= { (a,b), (b,c), (b,d) } find R\* and R+.
9. What is the relationship among RE, DFA, NFA and FA with Є –transition

11. Write regular expressions for the following languages over the alphabet Σ = {0,1}:

(a) All strings that do not end with 00.

є + 0 + 1 + (0 +1)∗(01 + 10 + 11)

(b) All strings that contain an even number of 1’s.

0∗(10∗10∗)∗

(c) All strings which do not contain the substring 10.

0∗1∗

12. Is it true, the language accepted by Є-NFA is different from the regular language? Justify your answer.

13.Define NFA with Є- transition. Is the NFA’s with Є -transition are more powerful than the NFA’s without Є -transition?

**Problems on FA**

14. Draw FA for any string without an odd number of consecutive O's AFTER an odd number of consecutive 1 's over the alphabet {0,1}

15. Draw FA for any string without more than 2 consecutive O's over the alphabet {0,1}

All strings that don’t contain the substring 101

16.All strings that contain exactly 3 zeros

17. Design a FSM to accept those strings made up of {0,1} such that machine will reach to the final state if it start with ‘00’ and end with 11.

18. Design a FA which accept odd number of ‘x’ and any number of ‘y’ .

19. Design FA to check whether a given unary number is divisible by 4.

20. Design FA to check whether a given binary number is divisible by three.

21. Design a FA that reads strings made up of symbols in the word “CHARIOT” and recognize those strings that contain the substring “HAIT” .

23.Construct a DFA accepting all strings w, over {0, 1} such that the number of '1's in w is multiples of 3.

24. Is NFA more Powerful than DFA? Justify.

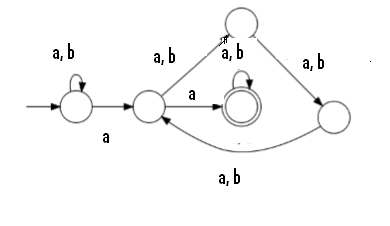
22. Construct FSM to accept a regular language L over input {a,b}such that every string consist at least one occurrence of substring “0110” .

23. Draw FA for any string without more than 2 consecutive a's over the alphabet {a,b}

Consider the following non-deterministic ﬁnite automaton (NFA) over the alphabet Σ = {a,b}.

Give a one-sentence description of the language recognized by the NFA. Write a regular

expression for this language.



24. Convert following NFA to its equivalent DFA

|  |  |  |  |
| --- | --- | --- | --- |
| states | a | B | Є |
| ->q0 | q2 | - | q1 |
| \*q1 | q0 | - | - |
| q2 | q1 | {q2, q1} | - |

26.Design NFA for L = {*x* | *x* ε {a, b}\* and *x* is any string that begins in “*abb*” or “*a*”}

27. Design NFA for L = {*x* | *x* ε {0, 1}\* and *x* is a string that contains at least one ‘1’ in each consecutive block of four symbols}

28. Design n-modulo-5 such that ‘n’ is a ternary number. i.e. *n* ε {0, 1, 2}\*

For Σ = { a, b}, set of all strings with no consecutive ‘a’s and ‘b’s

L= { 1(2n+1) | n ≥ 0}

**29.Design NFA for the following problems:**

1. all string in {a, b}+ with either two consecutive a’s or two consecutive b’s.



30. For the finite state machine M given in the following table, test whether the strings 101111, 11111 are accepted by M. (assume q3 is final state) 6M



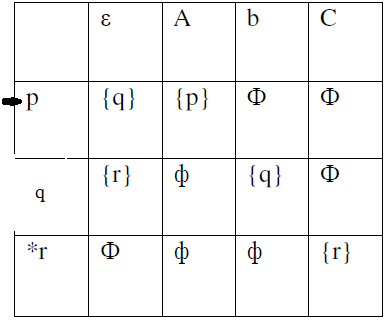
32. Construct a DFA equivalent to the NFA.

M=({p,q,r},{0,1}, δ,p,{q,s})

Where δ is defined in the following table. 6M



33. Consider the following ε–NFA. Compute the ε–closure of each state and find it’s equivalent DFA.(by direct method ) 8M



34. Consider the following ε–NFA. Compute the ε–closure of each state and

find it’s equivalent DFA. 8M

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ε | a | b | c |
| ->p | ε | {p} | {q} | ε |
| q | {p} | {q} | {r} | ε |
| \*r | {q} | {r} | ε | {p} |

35. Prove that following language is not regular [6]

L= {ww | w ∈(0∪1)+}

36. Design a FSM for divisibility by 3 testers from decimal numbers from 0 to 9. [4]

37. Define Following terms [4]

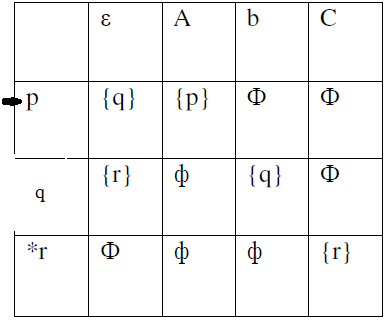
* 1. Symbol
  2. String
  3. L\*
  4. L+

38. Construct a DFA equivalent to the NFA. [6]

M=({p,q,r},{0,1}, δ,p,{q,s}) Where δ is defined in the following table. 

39.Consider the following ε–NFA. Compute the ε–closure of each state and find it’s

equivalent DFA using direct method [6]



40. Give the mealy and moore machine for the following processes “ For input from

(0+1)\*, if input ends in 101, output ‘X’, if input ends in 110 output ‘Y’, otherwise ‘Z’. [6]

41. Design Moore and Mealy machine for finding 2’s complement of a number [6]

1. Construct a DFA for the following [4]
   1. All strings that contain exactly 4 zeros.
   2. All strings that don’t contain the substring 110.

**OR**

1. Draw an NFA with ‘Є’ moves for string that have any number of zeroes, followed

by any numbers of one’s, followed by any number of two’s. Convert it to a NFA

without ‘Є’. [6]

1. Define Following Terms [4]
   1. Finite Automata
   2. Transition Diagram
   3. Moore Machine
   4. Mealy Machine

**OR**

1. Give Moore & Mealy machine which gives an output 1 if the input string ends in

**bab**” [4]

1. Design a DFA of all strings over alphabet {0,1} such that the third symbol from

right end is ‘1’. [6]

**OR**

1. Design a DFA that reads string made up of letters in the word “CHARIOT” and

recognizes these strings that contain the word “CAT” as substring. [6]

1. Convert following NFA to DFA. [4]

------------------------------------------------------------------

q δ(q,0) δ(q,1)

------------------------------------------------------------------

q0 {q0} {q0,q1}

q1 {q2} {q2}

q2 {q3} {q3}

q3 Ф Ф

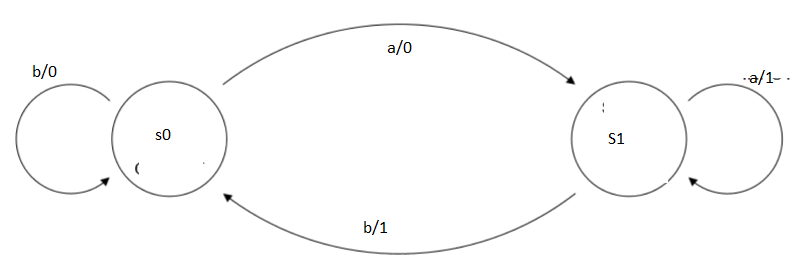
------------------------------------------------------------------

**Unit II- Deterministic and Non deterministic Finite Automata**

1. State and explain limitations and applications of FA.
2. Compare Moore and Mealy machines.
3. Design Moore and Mealy machines to find 2’s complement of given binary numbers
4. Convert given Mealy machine to its equivalent Moore machine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **i/p=0** | | **i/p=1** | |
|  | **Sates** | **output** | **Sates** | **Output** |
| ->q1 | q3 | 0 | q2 | 0 |
| q2 | q1 | 1 | q4 | 0 |
| q3 | q2 | 1 | q1 | 1 |
| q4 | q4 | 1 | q3 | 0 |

1. Convert following Mealy to Moore machine



1. Convert following NFA- ε to DFA

0 1 2 3

ε **^ ^**

ε **^ ^**

ε **^ ^**

1. Construct Mealy Machine that accepts strings ending with “00” and”11” , convert same to Moore machine. (6 M)

Q. Describe the language accepted by following Regular expressions and justify with examples

1. 1.(1+0)\*.10
2. (a\*.b.a\*.b.a\*)\*
3. 0\*1+1\*0

Q. Write regular expressions for each of the following languages:

i. For L: { a, b}, set of all strings with no consecutive a's and 'b's

ii. For L : { 0, 1}, set of all strings in which every 0 is immediately followed by at least two 1's.

iii. L={1(2n+1) | n>=1}

**FA to RE Conversion (Arden’s Theorem)**

1. Construct a Regular expression corresponding to the state diagram given in the following figure

.

1

**0 1**

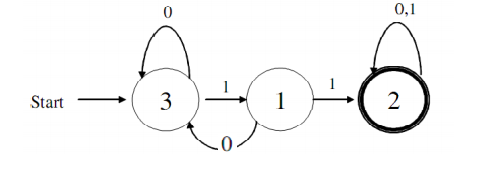
**1 0**

1. Construct a Regular expression corresponding to the state diagram given in

the following figure.

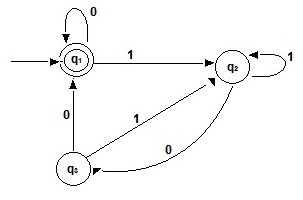


1. Find the regular expression for the following DFA’s. 6M



1. Find the regular expression for the set of strings recognized by the given

FA . Use Arden’s theorem.



1. Design the finite automata then equivalent regular expression using Arden’s theorem that accepts the set of all strings over the alphabet {a, b} with an equal number of a’s and b’s such that each prefix has at most one more a than b’s and at most one more b than a’s

***RE to DFA/NFA conversion***

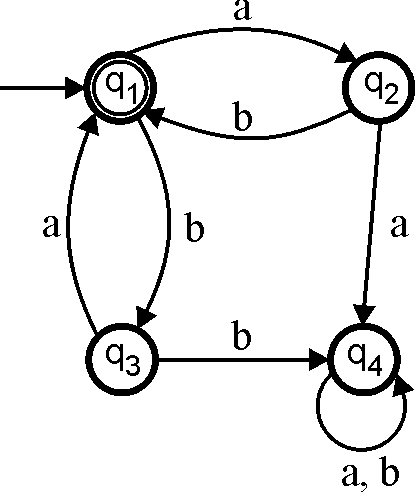
1. Construct an NFA equivalent to the following Regular Expressions ((0+1)(0+1)(0+1))\*
2. 10+(0+11)0\*1
3. 0\*1+10
4. (0+1)\* (00+11)(0+1)\*
5. (ab|ba)\* aa (ab|ba)\*
6. 01[((10)\*+111)\*+0]\*1
7. (1+10+110)\*0

**Problems on Pumping Lemma**

1. Using pumping lemma for regular sets, prove that the language

L = { am bn | m >n} is not regular.

1. Prove by pumping lemma that L= {anbn | n>=1 } is not regular language
2. Prove that, L={wwR | w is in (0 |1)\* and wR is reverse of w}, is not regular.
3. Prove by pumping lemma that L= {aib2i | i>=1 } is not regular language
4. Design Regular Expression for following statements :
   1. ∑={a,b} set of all string no consecutive ‘a’ and ‘b’
   2. ∑={0,1} set of all strings in which every zero immediately followed by at least two 1’s.
5. For given Finite Automata, find regular expression using Ardens’ Theorem [3]



1. Construct a NFA with ‘Є’ for following RE: [3]

(0+1)\*(011+01010)(0+1)\*

21. Construct a FA following RE: [3]

1(1+10)\*+10(0+01)\*

1. Show that {0i12i | i>0} is not regular using pumping lemma. [4]
2. Show that L={ap| p is a prime} is not regular using pumping lemma. [4]
3. Prove that, RE(LHS) = RE (RHS)
4. **= (11+111**
5. **= ^ + a ( a + b b**
6. (1+00\*1)+(1+00\*1) (0+10\*1)\* (0+10\*1)=0\*1(0+10\*1)\*
7. (a\*ab+ba)\*a\* = (a+ab+ba)\*
8. ε +1\*(011)\*(1\*(011)\*)\* = (1+011)\*
9. **1 1 = 1+0(0+1011**
10. Find regular expression for following Languages
11. L over ∑={a,b,c} , such that every string will have at least one ‘a’ followed by at least one ‘b’ followed by any number of ‘c’
12. L over ∑={0,1} , such that every string will have at least three consecutive 1’s.
13. L over ∑={0,1} , such that every string will begin and end with either 00 or 11..
14. L = {*x* | *x* ε {a, b}\* and *x* is any string that begins in “*abb*” or “*a*”}
15. L = {*x* | *x* ε {0, 1}\* and *x* is a string that contains at least one ‘*0*’ in each consecutive block of five symbols}
16. Design n-modulo-5 such that ‘n’ is a ternary number. i.e. *n* ε {0, 1, 2}\*
17. For Σ = { a, b}, set of all strings with no consecutive ‘a’s and ‘b’s
18. L= { 1(2n+1) | n ≥ 0}
19. Prove the formula [4]

(1+00\* 1)+ (1+00\* 1) (0+10\*1)\* (0+10\*1) =0\*1(0+10\*1)\*

1. Find regular expression corresponding to each of the following subset of {0, 1}\*

[6]

* 1. The language of all strings not containing the substring 111
  2. The language of all strings containing both 101 and 010 as substring.

**OR**

1. Construct a NFA with ‘Є’ for following RE: [6]
   1. (0+1)\*(011+01010)(0+1)\*
   2. (0+1)(01)\*(011)\*
2. Show that {0i1i | i>=1} is not regular using pumping lemma. [4]
3. Show that L={ap| p is a prime} is not regular using pumping lemma. [4]

. **Unit 3: Grammar`**

Q. Define ambiguous grammar with example. (2)

Q. S→aB|bA

A→a|aS|Baa

B→b|bS|Abb

Is ab, baba, abbbaa in L(G) ? (3)

Q. Write short note on Chomsky Hierarchy. (4)

Q. Define CFG. (1)

Q. Show that following grammar is ambiguous. (4)

S→a | abSb | aAb

Q. Define recursive and recursively enumerable languages with example. (4)

A→bs | aAAb

Q. Construct reduced grammar equivalent to grammar (5)

S→aAa

A→Sb | bcc| DaA

Questions: Design Context Free Grammar (CFG) for the following languages

1. ( 011 + 1 )\* . ( 01 )\*
2. L= { an.b2n | n ≥ 1}
3. L = { x |x Є { (, ) }\* with strings having well-formed parentheses (WFP)}
4. L = { x |x Є {a, b}\* with strings of even length and center two symbols the same }
5. L = { ambn | m, n>= 0 and m ≠ n }
6. L = { ambn | m, n>= 0 and m < n }
7. L = { aibjck| i = j + k }
8. L = { 0i1i+k 0k| i, k >=0}
9. L = { 0a1b 2c| |a-c| = b}
10. L = { 0i1j 0i+j| i, j >=0}
11. L = { 0i1j0k| j> (i + k) }

Q. Reduce the following grammar to CNF (6)

S→1A|0B

A→1AA|0S|0

B→0BB|1S|1

Q. Check whether the given grammar is CNF?

S→bA | aB

A→bAA | aS | a

B→aBB |bS | b

**UNIT 4 and 5**

Q**.** Design TM that creates a copy of its i/p string to the right of the i/p but with a blank separating the copy from the original.

Q. Design TM to do right shift operation on a binary number.

Q. Design TM to perform ***n mod 2*** operation where n is a unary no.

Q. Design TM which accepts the words from{ T,O,S,E,Y}\* and replaces occurrence of “TOY” by “TOS” .

Q. Construct a PDA accepting { anbman|m, n >= 1}.Construct the corresponding context free grammar accepting the same set.

Q Construct a PDA detecting palindrome.

Q. Consider the language L={ x {a,b}\* | na(x) >nb(x) }

Design a CFG for above language and convert it into PDA.

Q. Show that L={ap | p is a prime no. } is a not context free language.

Q Draw PDA for ODD length palindrome over alphabet {a,b} and draw the computation tree showing all possible sequences of moves for the two input strings “aba” and “aabab”.

Q. Distinguish between Turing Machine and Post machine.

Q. Design post machine over alphabet {a,b} for the strings to check equal no. of a’s over b’s.

Q. Design post machine over alphabet {0,1} for the L={ 0n12n} for n>0.

Q. Design post machine over alphabet {a,b,c} for L={ aibjck |k=i+j}.

**UNIT 6**

**Unit VI – Tractable & Intractable Problems**

1. What do you mean by Polynomial type problems?
2. Describe any problem in detail that is solvable in polynomial time.
3. Explain Kruskals Algorithm and find its time complexity.
4. What are non deterministic polynomial time algorithms?
5. Explain the problem classes NP Hard and NP Complete.
6. Explain relations between P, NP, NP complete and NP hard problems.
7. What is Satisfiability (SAT) problem?
8. What is Travelling Salesman Problem? Explain its time complexity.
9. Explain normal forms for Boolean Expressions with example.
10. Explain Node Cover problem with example.
11. Explain problem of independent set with example.
12. Explain Hamiltonian circuit problem with example.
13. Explain conversion of Boolean expression to CNF with example.
14. What do you mean by time complexity of any algorithm?
15. What do you mean by NP complete problems?
16. What are deterministic and non deterministic algorithms?
17. What are tractable and intractable problems?
18. Is the following formula satisfiable?

(x  y)  (x  y)  (x  y)  (x  y)

1. Explain the following terms with example
   1. Computational complexity
   2. P – class problems
   3. NP – class problems
2. Justify that the SAT Problem is NP-complete.
3. Explain in detail, the polynomial -time reduction approach for proving

that a problem is NP- Complete.

1. Explain Tractable and In-tractable Problem.
2. Justify whether the Traveling Salesman Problem is a class P or class NP problem.