**MAEER’s MIT Arts, Commerce & Science College, Alandi(D), Pune.**

**Question Bank**

**Class: - T.Y.B.Sc. (Computer Science)**

**Semester- I**

**Paper- II**

**Subject: - Theoretical Computer Science & Compiler Construction**

**Chapter 1) Preliminaries**

**1 mark questions**

1. Write a power set of A where A={1,2,3}
2. What is the value of n, where |ε|=n?
3. Let A= {1,2,3} and B={2,4,3} then write the Cartesian product of A and B.
4. What is the value of n, where n=| ε |.
5. Define the term language.
6. List any two properties of sets.
7. Define the term empty string with example.
8. Define suffix and list all proper suffixes of the string “abcd”.
9. What do you mean by diagonalization in the context of infinite sets?
10. Let A= {1,2,3}, B= {a,b} write A\*B and power of A.
11. Define countable infinite set. Give example.
12. What is the suffix and prefix of a string x=abcd?
13. Let A={0,1,2} B={1,3} Find (A∩B)\* and (A-B)\*
14. Let A={a,b} B={b,c} Find (A∩B)\* and (A-B)\*
15. Let X= {a,b,c} and Y= {e,f,g} then write the Cartesian product of X and Y.
16. Let A= {a,b,c} and B={d,e,f} then write the Cartesian product of A and B.

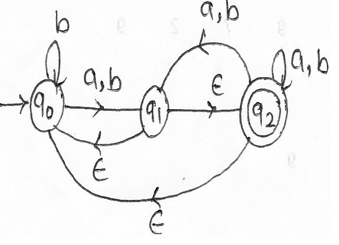
**Chapter 2) Finite Automata**

**1 mark questions**

1. Write a mapping of δ in case of NFA and DFA both.
2. Construct NFA for language L where L={a(a+b)\*b}
3. Write 5 tuples of DFA and NFA.
4. Construct NFA without ε for language L where L=(0+1)\*01.
5. Write the condition for string acceptance in FA.
6. Construct NFA for language L where L=01\*(01)\*+1.
7. State equivalence theorem of NFA and DFA.
8. Define ε -closure (q).
9. Construct NFA with ε - transitions for ab\*+ba\*.
10. State any two disadvantages of FA.
11. Explain transition function of DFA and NFA.
12. FA have only 1 start state. State true or false.
13. FA have more than one final states. Justify true or false.

**5 Mark Questions**

14. Construct DFA equivalent to given NFA.



15. Construct Mealy machine over {0,1} which toggles it’s input.

16. Construct DFA for language over {0,1} having two consecutive 1’s

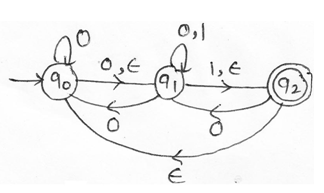
and three consecutive 0’s.

17. Find the minimum state DFA equivalent to the following DFA..

M=({q0, q1, q2, q3, q4, q5},{a,b}, δ,q0, {q3,q5 })

|  |  |  |
| --- | --- | --- |
| δ | a | b |
| q0 | q2 | q4 |
| q1 | q0 | q5 |
| q2 | q2 | q4 |
| q3 | q0 | q4 |
| q4 | q1 | q3 |
| q5 | q0 | q5 |

18. Construct DFA equivalent to given NFA.



19. Construct Moore machine to generator 1’s complements of number.

20. Construct DFA for accepting string over {a,b,c} such that if it starts

with ‘a’ then it contains substring ‘abc’ in it and if it starts with ‘b’

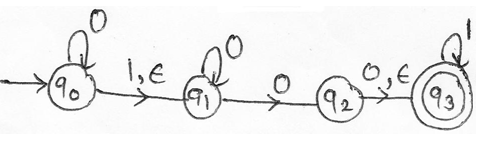
then it ends with ‘c’.  
21. Construct Moore machine to take input from {a,b} which outputs

even or odd according to number of a’s encountered is even or odd.

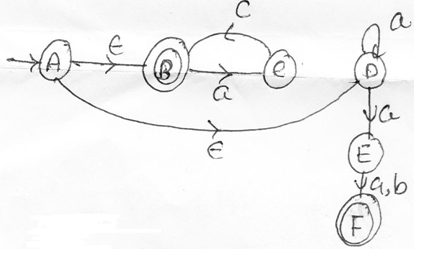
22. Construct DFA for accepting string over {a,b,c} such that if it starts

with ‘a’ and not having substring ‘bac’ in it.

23. Construct DFA equivalent to given NFA.



24. Construct DFA equivalent to given NFA.



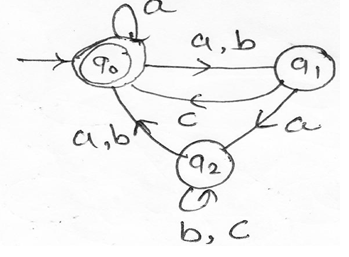
25. Construct Mealy machine to convert each occurrence of substring

101 by 100 over alphabet {0,1}

26. Construct DFA for language L over {p,q,r} which accepts all stirngs

ending with ‘pq’ and not having substring ‘rpq’ in it.

27. Construct DFA equivalent to given NFA.



28. Construct Mealy machine to input from (a+b)\* such that if input has

substring ‘aba’ the machine outputs “A”, if it has substring “aab”, the

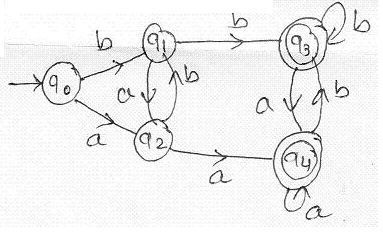
machine output ‘B’,otherwise outputs ‘C’.

29. Construct a DFA for all the following language L.

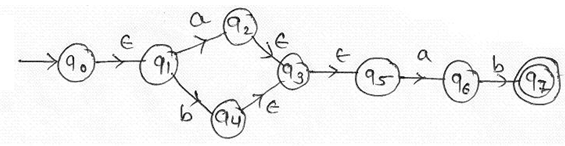
Where L= {set of all strings over {a,b,c} with minimum length 3 and

second and second last symbol is same.

30. Find minimum state FA for the following DFA.



31. Construct DFA equivalent to given NFA.



32. Design Moore machine to give output A, if the string ends in ‘pqr’,

‘B’ if the string ends in ‘prp’, otherwise output will be ‘C’.

33. Construct DFA for language over {0,1} to accept string which is

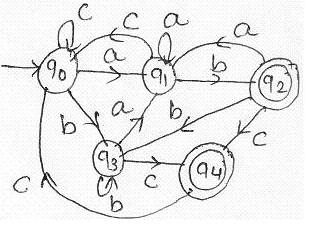
treated as binary no. and whose decimal equivalent is decimal by 3.

34. Design Moore machine for binary input sequence such that if it has a

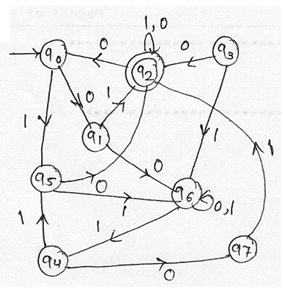
substring “010” the machine outputs “#”, if it has substring “110”,

then outputs “\*” else output “$”.

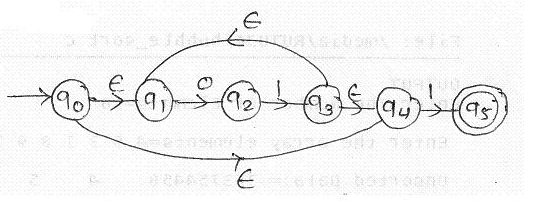
35. Find minimum state FA for the following DFA.



36. Find minimum state FA for the following DFA.

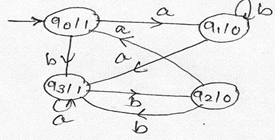


37. Construct DFA equivalent to given NFA.



38. Construct equivalent Mealy machine of the following Moore

machine.



40. Find minimum state FA for the following DFA.

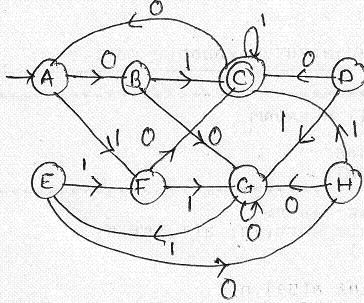
M=({q0, q1, q2, q3, q4},{a,b,c}, δ,q0, {q1, q2,q4 })

|  |  |  |  |
| --- | --- | --- | --- |
| δ | a | b | C |
| q0 | q1 | q3 | q3 |
| q1 | q1 | q2 | q4 |
| q2 | q1 | q4 | q3 |
| q3 | q3 | q3 | q3 |
| q4 | q1 | q4 | q4 |

41. Define equivalence theorem of Moore and Mealy machine with suitable example.

42. Find minimum state FA for the following DFA.

M=({A,B,C,D,E,F,G,H},{0,1}, δ,A, {C})



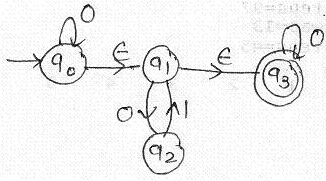
43. Construct a DFA for the language accepting string starting with ‘b’ and not having ‘aba’ as a substring over {a,b}

44. Construct a DFA for language L ={a+(a+b)\*}

45. Construct DFA to accept binary number whose decimal equivalent is

divisible by 5.

46. Construct DFA equivalent to given NFA with ε-move.



47 . Find the minimum state DFA equivalent to the following DFA..

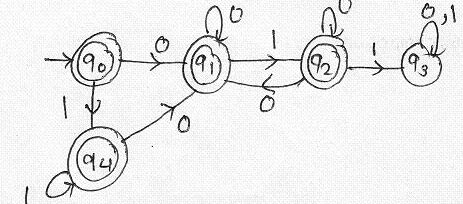
M=({q0, q1, q2, q3, q4, q5, q6},{0,1}, δ,q0, q5 )

|  |  |  |
| --- | --- | --- |
| δ | 0 | 1 |
| q0 | q1 | q2 |
| q1 | q3 | q4 |
| q2 | q5 | q6 |
| q3 | q3 | q4 |
| q4 | q5 | q6 |
| q5 | q3 | q4 |
| q6 | q5 | q6 |

48. Construct DFA for a language over {0,1,2} which start with ‘00’ ends

with ‘22’ and having substring ‘11’ in it.

49. Find minimum state FA for the following DFA.



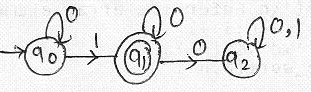
**Chapter 3) Regular Languages**

**1 Mark questions**

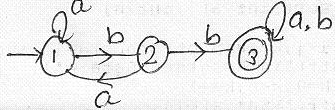
1. Write smallest possible strings accepted by the following regular expression 10 + (0+11)0\*1.
2. Write smallest possible strings accepted by the following regular expression (ab+ba\*)\*b.
3. If ε is regular expression then it denotes set {ε} justify.
4. Write smallest possible strings accepted by the following regular expression a(a+b)\*ab.
5. Describe the language L= {an bn |n>=1}.
6. Write smallest possible strings accepted by the following regular expression 01 + (0+1)01\*.
7. Describe in English the set of accepted by following regular expression a(a+b)\*b + b(a+b)\*a
8. Write a regular expression representing following set.

L=set of all strings of a’s and b’s of length 10 exactly.

1. Describe in English language the set accepted by following FA.



1. Describe the language for regular expression (011)\*0.
2. Write a regular expression for language of all string that begins and ends with 00 or 11.
3. Describe the language for following FA.



1. Let L1= (a+b)\*, L2= (0+1)\* write L1 U L2.
2. Let L1= (a\*b\*+b\*a\*), L2=(a+b)\* write L1 ∩ L2.
3. Find the language L of the following (a\*b\*)\*a.
4. Define a language over ∑ = {0,1} containing all possible combination of 0’s and 1’s but not having two consecutive 0’s.
5. Explain kleene closure and positive closure with example.
6. Find the languages for the (a+b)\*
7. Define the language containing all strings of a’s and b’s having at least one combination of double letters using regular expression.
8. Construct NFA with ε-transition for the regular expression 01\*+1.
9. Give any two identities of regular expression.
10. If L1 contains equal no of a’s and b’s over {a,b} and L2 ={a\*b\*}

Find L1 ∩ L2.

1. State pumping lemma of regular set and also state its application.
2. Give two kinds of operation that can be carried out on regular languages.
3. State pumping lemma of regular set.
4. Is (a\*+b\*)\* = (a+b)\* true? Justify.
5. Construct NFA with ε-transition for a\*+b.
6. Every regular languages is CFL, State True or False.
7. Write smallest possible strings generated by regular expression

a (a+b)b\*.

1. What will be the smallest possible strings generated by regular

expression = (ab\*)\*

**5 Mark Questions**

1. Construct FA for the following regular expression

a(a+b)\*b+b(a+b)\*a.

1. Construct FA for the following regular expression

ab(a+b)\*+ba(a+b)\*.

1. Construct FA for the following regular expression

((a+b)\*+abb)\*.

1. Check whether following language is regular. Justify your answer.

a. L= {0p | p is perfect No.}

b. Find language for the following ab\* + ab\*

1. Construct FA for (01+10)\* +0(01\*)\*.
2. Prove L = {an bn an | n>=0} is not regular.
3. Show that L = {x xR | x ε {a,b}\*} is not regular.
4. Show that regular sets are closed under complementation with an

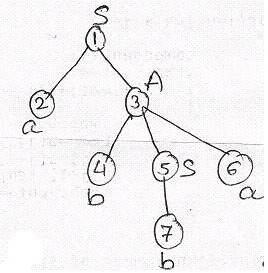
example.

1. Show that regular sets are closed under union and concatenation with an example.
2. Construct NFA for regular expression (1(01)\*)+(0(01\*1).
3. Draw FA equivalent to regular expression a(a+b)\*b+b(b+a)\*a.

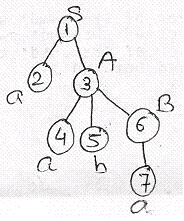
**Chapter 4) Context Free Grammar and Languages.**

**1 Mark Question**

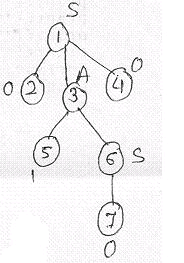
1. Define nullable symbol.
2. Define useless symbol.
3. What yield the derivation tree given below?



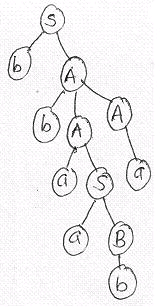
1. Define unit production.
2. Every CFL is regular language. State True or False.
3. Define ambiguous grammar.
4. Define Left linear grammar.
5. What yield the derivation tree given below?



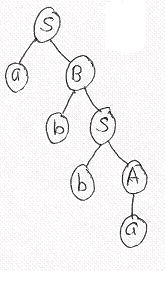
1. Define inherently ambiguous context free language.
2. Define phrase structure grammar.
3. State any one lemma used in the procedure for constructing GNF.
4. What yield the derivation tree given below?



1. State any one lemma used in the procedure for constructing GNF.
2. Define Right linear grammar with example.
3. What yield the derivation tree given below?



1. What are the types of grammar in Chomsky hierarchy?
2. Write the step for eliminating ε-production in CFG.
3. Mension various methods of grammar simplification.
4. What yield the derivation tree given below?



1. State the lemma 2 for converting a CFG to GNF.
2. Define unrestricted grammar.
3. Define Context Sensitive Grammar.

**5 Mark Questions**

1. Convert the following grammar into GNF

S🡪AB

A🡪BS | b

B🡪SA | a

1. Convert the following grammar into CNF

S🡪 aAbB | BbS

B🡪 aAbA | Aab | B

A🡪 aB | aBb |a

1. Rewrite the following grammar after removing ε-production

S🡪 XaX | Bx

X🡪 XaX | abX | ε

1. Construct CFG for a language over {a,b,c} which accepts equal

number of a’s and b’s.

1. Convert the following grammar into GNF

S🡪AB

A🡪SB | a

B🡪AB | b

1. Rewrite the following grammar after removing useless symbol

S🡪AB | CA

B🡪 BC | AB

A🡪 a

C🡪 aB | b

D🡪 SS | d

1. Construct CFG which accepts set of palindromes over {0,1}\*.
2. Convert the following grammar into CNF

S🡪 AaB | a

B🡪 Ba | b

A🡪 SBb | bA

1. Rewrite the following grammar after removing ε-production

S🡪 aS | AB

A🡪 a | ε

B🡪 b | ε

D🡪 b

1. Convert the following grammar into CNF

S🡪 ABC

B🡪 Bb | bb

A🡪 a | b

C🡪 aC | CC | ba

1. Construct CFG for a language L over {a,b} which generates strings where number of a’s are twice to that of number of b’s.
2. Convert the following grammar into GNF

S🡪ABC

A🡪 a | b

B🡪Bb | aa

C🡪 aC | cC | ba

1. Convert the following grammar into CNF

P🡪 aQRb | QRP

Q🡪 Rb | PR

R🡪 a | QRS

S🡪 b

1. Remove unit production from following grammar

S🡪 A | bb

A🡪 B | b

B🡪 S | a

1. Convert the following grammar into GNF

S🡪aAS | a

A🡪 SbA | SS | bA

1. Construct CFG for L = {0x1y2z | y > x+z}
2. Convert the following grammar into CNF

X🡪 aXbYcZ

Y🡪 XYZ | b

Z🡪 a | b | ε

1. Remove unit production from B-production grammar

S🡪 A | bb

A🡪 B | b

B🡪 A | bb | a

1. Convert the following grammar into CNF

S🡪 ABA

A🡪 aA | ε

B🡪 bB | ε

1. Define the following with suitable examples
   * 1. Leftmost derivation
     2. Rightmost derivation
     3. Ambiguous Grammar
2. Convert the following grammar into GNF

S🡪 RR | a

R🡪 SS | b

1. Convert the following grammar into GNF

S🡪 AA | 0

A🡪 SS | 1

1. Rewrite the grammar after removing ε–production.

S 🡪 aSa | bSb | ε

A 🡪 aBb | bBa

B 🡪 aB | bB | ε

1. Construct CFG for a language L where

L = {an bn cm dm | m,n>=1}

1. Convert the following grammar into GNF

A1🡪 A2A3

A2🡪 A3A1 | a

A3🡪 A1A2 | b

1. Convert the following grammar into CNF

S🡪0Q | 1P | P

P🡪 0 | 0S | 1PP

Q🡪 1 | 1S | 0QQ

1. Convert the following grammar into CNF

S🡪 aSd | aAd

A🡪 bAc | bc

**Chapter 5) Pushdown Automata**

**1 Mark Questions**

1. Write formal definition of DPDA.
2. Write formal definition of NDPDA.
3. The class of language accepted by DPDA and NPDA is same justify.
4. Give diagrammatic representation of PDA.
5. Define ID for PDA.
6. State formal definition of PDA.
7. Differentiate between DPDA and NPDA.
8. DPDA and NPDA are not equivalent. Justify.
9. State two methods for defining language accepted by PDA.
10. State any two difference between PDA and FA.
11. How δ function is mapped in PDA?
12. Write tuples of PDA.
13. Class of CFG and PDA is same. Justify True/ False.

5 Mark Questions

1. Construct PDA for L = {ax by cz | x=2y+z,y,z>=1}.
2. Construct PDA for L = {an b2n ck | n>=1,k>=0}.
3. Construct PDA for L = {am bn cn+2 dm | n>=1,m>=1}.
4. Construct PDA for following CFG

S🡪 aAb | As

A🡪 Bb | a

B🡪 Sa | b

1. Construct PDA for L = {an bn +1 | n>=1}.
2. Construct PDA for L = {an b2n +1 | n>=1}.
3. Construct PDA for L = {0m 1n 2k | m,n,k>=1,m=n+k}.
4. State the equivalence theorem of PDA and CFL. Also construct a

PDA for a language odd palindrome.

1. Show that DPDA and NPDA are not equivalent with an appropriate example.
2. Construct PDA over {0,1} to accept a string which does not contain the substring “00”.
3. Construct PDA for L = {am bn cm+n  | n>=1,m>=1}.
4. Convert CFG to PDA.

S🡪 aB | bA

A🡪 a | aS | bAA

B🡪 b | bS | aBB

**Chapter 6) Turing Machine**

**1 Mark Questions**

1. Define TM.
2. Give diagrammatic representation of TM.
3. What is difference between TM and FA.
4. Every language accepting by TM is regular. State true or false.
5. Give the snapshot of TM and obtain instantaneous description.
6. Explain move of TM.
7. Define: ID of a turing machine.
8. Define language accepted by a turing machine.

**5 Mark Questions**

1. Construct TM accepting following language

L = {am bn | m>n and n>0}

1. Construct TM accepting following language

L = {am bn | n>=m and m>=1}

1. Construct TM accepting following language

L = {0i 1 2i+2 | i>=0}

1. Design a TM to recognize well- formedness of parenthesis (only

round open close brackets).

1. Construct TM accepting following language

L = {02n 2n | n>=1}

1. Design a TM that recognizes the well- formedness of parenthesis

over {(,)}

1. Construct TM accepting following language

L = {0n 1n 0 n | n>=1}.