

30 *Years*
Previous Solved Papers

GATE 2025

Computer Science & Information Technology



- ✓ Fully solved with explanations
- ✓ Analysis of previous papers
- ✓ Topicwise presentation
- ✓ Thoroughly revised & updated





MADE EASY Publications Pvt. Ltd.

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016 | **Ph.:** 9021300500, 8860378007

Email : infomep@madeeasy.in | **Web :** www.madeeasypublications.org



GATE - 2025

Computer Science & IT

Topicwise Previous GATE Solved Papers (1995-2024)

Editions

1 st Edition	: 2007
2 nd Edition	: 2008
3 rd Edition	: 2009
4 th Edition	: 2010
5 th Edition	: 2011
6 th Edition	: 2012
7 th Edition	: 2013
8 th Edition	: 2014
9 th Edition	: 2015
10 th Edition	: 2016
11 th Edition	: 2017
12 th Edition	: 2018
13 th Edition	: 2019
14 th Edition	: 2020
15 th Edition	: 2021
16 th Edition	: 2022
17 th Edition	: 2023
18th Edition	: 2024

© Copyright, by **MADE EASY Publications Pvt. Ltd.**

All rights are reserved. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photo-copying, recording or otherwise), without the prior written permission of the above mentioned publisher of this book.

MADE EASY Publications Pvt. Ltd. has taken due care in collecting the data and providing the solutions, before publishing this book. In spite of this, if any inaccuracy or printing error occurs then **MADE EASY Publications Pvt. Ltd.** owes no responsibility. We will be grateful if you could point out any such error. Your suggestions will be appreciated.

Preface

Over the period of time the GATE examination has become more challenging due to increasing number of candidates. Though every candidate has ability to succeed but competitive environment, in-depth knowledge, quality guidance and good source of study is required to achieve high level goals.



B. Singh (Ex. IES)

The new edition of **GATE 2025 Solved Papers : Computer Science & Information Technology** has been fully revised, updated and edited. The whole book has been divided into topicwise sections.

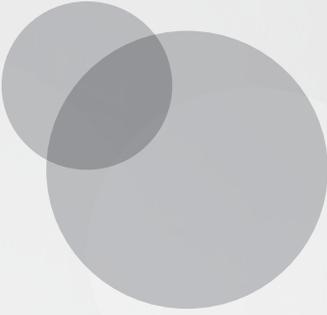
At the beginning of each subject, analysis of previous papers are given to improve the understanding of subject.

I have true desire to serve student community by way of providing good source of study and quality guidance. I hope this book will be proved an important tool to succeed in GATE examination. Any suggestions from the readers for the improvement of this book are most welcome.

B. Singh (Ex. IES)

Chairman and Managing Director

MADE EASY Group



GATE-2025

Computer Science & IT

CONTENTS

1.	Discrete and Engineering Mathematics.....	1-124
2.	Theory of Computation.....	125-188
3.	Digital Logic.....	189-245
4.	Computer Organization and Architecture.....	246-311
5.	Programming and Data Structures.....	312-414
6.	Algorithms.....	415-484
7.	Compiler Design.....	485-521
8.	Operating System.....	522-594
9.	Databases.....	595-663
10.	Computer Networks.....	664-721
11.	General Aptitude.....	722-770

Theory of Computation

UNIT

II

CONTENTS

1. Finite Automata : Regular Languages 127
2. Push Down Automata: CFL & DCFL 158
3. Turing Machine: RE, REC and Undecidability 177

Theory of Computation

Syllabus

Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context-free languages, pumping lemma. Turing machines and undecidability.

Analysis of Previous GATE Papers

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
1995	1	2	5
1996	3	2	7
1997	1	3	7
1998	5	3	11
1999	3	1	5
2000	2	1	4
2001	3	2	7
2002	1	6	18
2003	3	6	15
2004	1	4	9
2005	–	7	14
2006	2	5	12
2007	2	5	12
2008	3	5	13
2009	4	3	10
2010	1	3	7
2011	3	3	9
2012	4	1	6
2013	2	3	8

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2014 Set-1	2	2	6
2014 Set-2	2	2	6
2014 Set-3	2	2	6
2015 Set-1	1	2	5
2015 Set-2	1	3	7
2015 Set-3	1	1	3
2016 Set-1	3	3	9
2016 Set-2	3	4	11
2017 Set-1	2	4	10
2017 Set-2	3	3	9
2018	2	3	8
2019	2	3	8
2020	3	3	9
2021 Set-1	2	3	8
2021 Set-2	3	4	11
2022	2	3	8
2023	3	3	9
2024 Set-1	–	3	6
2024 Set-2	1	3	7

- 1.1** Which of the following definitions below generates the same language as L

Where $L = \{x^n y^n \mid n \geq 1\}$

- (i) $E \rightarrow xEy \mid xy$
 (ii) $xy \mid (x^+xyy^+)$
 (iii) x^+y^+
 (a) (i) only (b) (i) and (ii)
 (c) (ii) and (iii) (d) (ii) only

[1995 : 2 M]

- 1.2** A finite state machine with the follows state table has a single input X and a single output Z.

present state	next state, z	next state, z
	x=1	x=0
A	D,0	B,0
B	B,1	C,1
C	B,0	D,1
D	B,1	C,0

It the initial state is unknown, then the shortest input sequence to reach the final state C is

- (a) 01 (b) 10
 (c) 101 (d) 110 [1995 : 2 M]

- 1.3** Which two of the following four regular expressions are equivalent?

- (i) $(00)^* (\epsilon + 0)$ (ii) $(00)^*$
 (iii) 0^* (iv) $0(00)^*$
 (a) (i) and (ii) (b) (ii) and (iii)
 (c) (i) and (iii) (d) (iii) and (iv)

[1996 : 1 M]

- 1.4** Let $L \subseteq \Sigma^*$ where $\Sigma = \{a, b\}$, which of the following is true?

- (a) $L = \{x \mid x \text{ has an equal number of a's and b's}\}$ is regular
 (b) $L = \{a^n b^n \mid n \geq 1\}$ is regular
 (c) $L = \{x \mid x \text{ has more a's than b's}\}$ is regular
 (d) $L = \{a^m b^n \mid m \geq n \geq 1\}$ is regular

[1996 : 1 M]

- 1.5** Given $\Sigma = \{a, b\}$, which one of the following sets is not countable?

- (a) Set of all strings over Σ
 (b) Set of all languages over Σ
 (c) Set of all regular languages over Σ
 (d) Set of all languages over Σ accepted by Turing Machines [1997 : 1 M]

- 1.6** Which one of the following regular expressions over $\{0,1\}$ denotes the set of all strings not containing 100 as a substring?

- (a) $0^*(1+0)^*$ (b) 0^*1010^*
 (c) $0^*1^*01^*$ (d) $0^*(10+1)^*$

[1997 : 2 M]

- 1.7** If the regular set A is represented by $A = (01+1)^*$ and the regular set 'B' is represented by $B = ((01)^*1^*)^*$, which of the following is true?

- (a) $A \subset B$
 (b) $B \subset A$
 (c) A and B are incomparable
 (d) $A = B$

[1998 : 1 M]

- 1.8** Which of the following set can be recognized by a Deterministic Finite state Automaton?

- (a) The numbers 1, 2, 4, 8, 2^n , written in binary
 (b) The numbers 1, 2, 4, 2^n , written in unary
 (c) The set of binary string in which the number of zeros is the same as the number of ones.
 (d) The set $\{1, 101, 11011, 1110111, \dots\}$

[1998 : 1 M]

- 1.9** The string 1101 does not belong to the set represented by

- (a) $110^*(0+1)$
 (b) $1(0+1)^*101$
 (c) $(10)^*(01)^*(00+11)^*$
 (d) $(00+(11)^*0)^*$

[1998 : 1 M]

- 1.10** How many substrings of different lengths (non-zero) can be formed from a character string of length n?

- (a) n (b) n^2
 (c) 2^n (d) $\frac{n(n+1)}{2}$

[1998 : 1 M]

- 1.11** Let L be the set of all binary strings whose last two symbols are the same. The number of states in the minimum state deterministic finite state automaton accepting L is

- (a) 2 (b) 5
 (c) 8 (d) 3

[1998 : 2 M]

- 1.12** Consider the regular expression $(0 + 1)(0 + 1) \dots$ n times. The minimum state finite automation that recognizes the language represented by this regular expression contains
- (a) n states
 (b) $n + 1$ states
 (c) $n + 2$ states
 (d) None of these

[1999 : 1 M]

- 1.13** Let S and T be language over $\Sigma = \{a, b\}$ represented by the regular expressions $(a + b^*)^*$ and $(a + b)^*$, respectively. Which of the following is true?
- (a) $S \subset T$
 (b) $T \subset S$
 (c) $S = T$
 (d) $S \cap T = \phi$

[2000 : 1 M]

- 1.14** Let L denotes the language generated by the grammar $S \rightarrow 0S0 \mid 00$. Which of the following is true?
- (a) $L = 0^+$
 (b) L is regular but not 0^+
 (c) L is context free but not regular
 (d) L is not context free

[2000 : 1 M]

- 1.15** What can be said about a regular language L over $\{a\}$ whose minimal finite state automation has two states?
- (a) Can be $\{a^n \mid n \text{ is odd}\}$
 (b) Can be $\{a^n \mid n \text{ is even}\}$
 (c) Can be $\{a^n \mid n \geq 0\}$
 (d) Either L can be $\{a^n \mid n \text{ is odd}\}$ or L can be $\{a^n \mid n \text{ is even}\}$

[2000 : 2 M]

- 1.16** Consider the following statements:
- $S_1 : \{0^{2n} \mid n \geq 1\}$ is a regular language
 $S_2 : \{0^m 1^n 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1\}$ is a regular language
- Which of the following is true about S_1 and S_2 ?
- (a) Only S_1 is correct
 (b) Only S_2 is correct
 (c) Both S_1 and S_2 are correct
 (d) None of S_1 and S_2 is correct

[2001 : 1 M]

- 1.17** Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least
- (a) N^2
 (b) 2^N
 (c) $2N$
 (d) $N!$

[2001 : 1 M]

- 1.18** Consider a DFA over $\Sigma = \{a, b\}$ accepting all strings which have number of a 's divisible by 6 and number of b 's divisible by 8. What is the minimum number of states that the DFA will have?

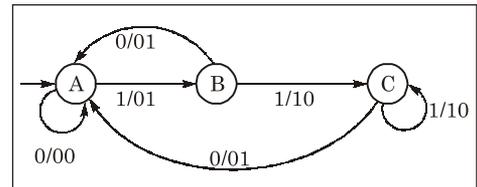
- (a) 8
 (b) 14
 (c) 15
 (d) 48

[2001 : 2 M]

- 1.19** Consider the following languages:
- $L_1 = \{ww \mid w \in \{a, b\}^*\}$
 $L_2 = \{ww^R \mid w \in \{a, b\}^*, w^R \text{ is the reverse of } w\}$
 $L_3 = \{0^{2i} \mid i \text{ is an integer}\}$
 $L_4 = \{0^{i^2} \mid i \text{ is an integer}\}$
- Which of the languages are regular?
- (a) Only L_1 and L_2
 (b) Only L_2, L_3 and L_4
 (c) Only L_3 and L_4
 (d) Only L_3

[2001 : 2 M]

- 1.20** The finite state machine described by the following state diagram with A as starting state, where an arc label is x/y and x stands for 1-bit input and y stands for 2-bit output



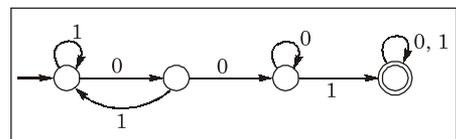
- (a) Outputs the sum of the present and the previous bits of the input
 (b) Outputs 01 whenever the input sequence contains 11
 (c) Output 00 whenever the input sequence contains 10
 (d) None of the above

[2002 : 2 M]

- 1.21** The smallest finite automation which accepts the language $L = \{x \mid \text{length of } x \text{ is divisible by } 3\}$ has
- (a) 2 states
 (b) 3 states
 (c) 4 states
 (d) 5 states

[2002 : 2 M]

- 1.22** Consider the following deterministic finite state automaton M .

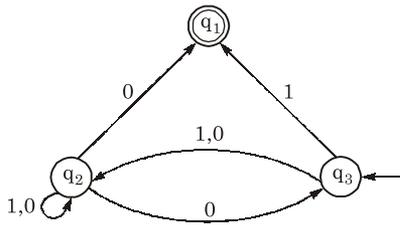


Let S denote the set of seven bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is

- (a) 1
 (b) 5
 (c) 7
 (d) 8

[2003 : 2 M]

1.23 Consider the NFA M shown below:



Let the language accepted by M be L. Let L_1 be the language accepted by the NFA M_1 , obtained by changing the accepting state of M to a non-accepting state and by changing the non-accepting state of M to accepting states. Which of the following statements is true?

- (a) $L_1 = \{0, 1\}^* - L$
- (b) $L_1 = \{0, 1\}^*$
- (c) $L_1 \subseteq L$
- (d) $L_1 = L$ [2003 : 2 M]

1.24 Which one of the following regular expressions is NOT equivalent to the regular expression $(a + b + c)^*$?

- (a) $(a^* + b^* + c^*)^*$
- (b) $(a^*b^*c^*)^*$
- (c) $((ab)^* + c^*)^*$
- (d) $(a^*b^* + c^*)^*$

[2004 : 1 M]

1.25 Let $M = (K, \Sigma, \delta, s, F)$ be a finite state automaton, where

$K = \{A, B\}, \Sigma = \{a, b\}, s = A, F = \{B\},$

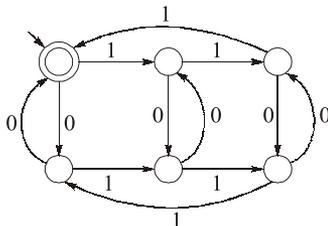
$\delta(A, a) = A, \delta(A, b) = B, \delta(B, a) = B$ and $\delta(B, b) = A$
 A grammar to generate the language accepted by M can be specified as $G = (V, \Sigma, R, S)$, Where $V = K \cup \Sigma$, and $S = A$

Which one of the following set of rules will make $L(G) = L(M)$?

- (a) $\{A \rightarrow aB, A \rightarrow bA, B \rightarrow bA, B \rightarrow aA, B \rightarrow \epsilon\}$
- (b) $\{A \rightarrow aA, A \rightarrow bB, B \rightarrow aB, B \rightarrow bA, B \rightarrow \epsilon\}$
- (c) $\{A \rightarrow bB, A \rightarrow aB, B \rightarrow aA, B \rightarrow bA, B \rightarrow \epsilon\}$
- (d) $\{A \rightarrow aA, A \rightarrow bA, B \rightarrow aB, B \rightarrow bA, A \rightarrow \epsilon\}$

[2004 : 2 M]

1.26 The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively



- (a) divisible by 3 & 2
- (b) odd and even
- (c) even and odd
- (d) divisible by 2 & 3

[2004 : 2 M]

1.27 Which of the following statements is TRUE about the regular expression 01^*0 ?

- (a) It represents a finite set of finite strings.
- (b) It represents an infinite set of finite strings.
- (c) It represents a finite set of infinite strings.
- (d) It represents an infinite set of infinite strings.

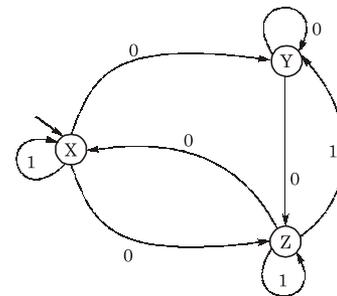
[2005 : 1 M]

1.28 The language $\{0^n 1^n 2^n \mid 1 \leq n \leq 10^6\}$ is

- (a) Regular
- (b) Context-free but not regular
- (c) Context-free but its complement is not context-free
- (d) Not context-free

[2005 : 1 M]

1.29 Consider the non-deterministic finite automaton (NFA) shown in the figure.



State X is the starting state of the automaton. Let the language accepted by the NFA with Y as the only accepting state be L_1 . Similarly, let the language accepted by the NFA with Z as the only accepting state be L_2 . Which of the following statements about L_1 and L_2 is TRUE?

- (a) $L_1 = L_2$
- (b) $L_1 \subset L_2$
- (c) $L_2 \subset L_1$
- (d) None of these

[2005 : 2 M]

1.30 Consider the regular grammar:

- $S \rightarrow Xa \mid Ya$
- $X \rightarrow Za$
- $Z \rightarrow Sa \mid \epsilon$
- $Y \rightarrow Wa$
- $W \rightarrow Sa$

where S is the starting symbol, the set of terminals is {a} and the set of non-terminals is {S, W, X, Y, Z}. We wish to construct a deterministic finite automaton (DFA) to recognize the same language. What is the minimum number of states required for the DFA?

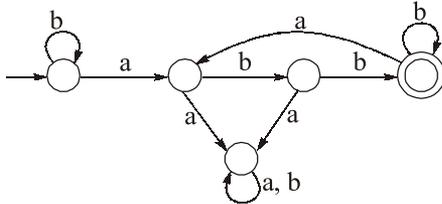
- (a) 2
- (b) 3
- (c) 4
- (d) 5

[2005 : 2 M]

1.31 A language L satisfies the Pumping Lemma for regular languages, and also the Pumping Lemma for context-free languages. Which of the following statements about L is TRUE?

- (a) L is necessarily a regular language.
- (b) L is necessarily a context-free language, but not necessarily a regular language.
- (c) L is necessarily a non-regular language.
- (d) None of the above [2005 : 2 M]

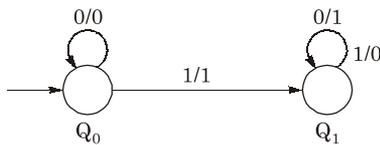
1.32 Consider the machine M :



The language recognized by M is

- (a) $\{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is followed by exactly two } b\text{'s}\}$
- (b) $\{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is followed by at least two } b\text{'s}\}$
- (c) $\{w \in \{a, b\}^* \mid w \text{ contains the substring 'abb'}\}$
- (d) $\{w \in \{a, b\}^* \mid w \text{ does not contain 'aa' as a substring}\}$ [2005 : 2 M]

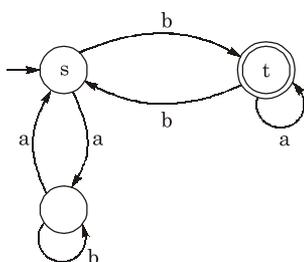
1.33 The following diagram represents a finite state machine which takes as input a binary number from the least significant bit



Which one of the following is TRUE?

- (a) It computes 1's complement of the input number
- (b) It computes 2's complement of the input number
- (c) It increments the input number
- (d) It decrements the input number [2005 : 2 M]

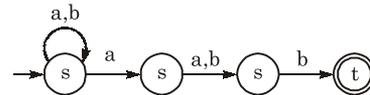
1.34 In the automaton below, s is the start state and t is the only final state.



Consider the strings $u = \text{abbaba}$, $v = \text{bab}$, and $w = \text{aabb}$. Which of the following statements is true?

- (a) The automaton accepts u and v but not w
- (b) The automaton accepts each of u , v , and w
- (c) The automaton rejects each of u , v , and w
- (d) The automaton accepts u but rejects v and w [2006 : 1 M]

1.35 Which regular expression best describes the language accepted by the non-deterministic automaton below?



- (a) $(a + b)^* a(a + b)b$
- (b) $(abb)^*$
- (c) $(a + b)^* a(a + b)^* b(a + b)^*$
- (d) $(a + b)^*$ [2006 : 1 M]

1.36 Consider the regular grammar below:

$$S \rightarrow bS \mid aA \mid \epsilon$$

$$A \rightarrow aS \mid bA$$

The Myhill-Nerode equivalence classes for the language generated by the grammar are

- (a) $\{w \in (a + b)^* \mid \#_a(w) \text{ is even}\}$ and $\{w \in (a + b)^* \mid \#_a(w) \text{ is odd}\}$
- (b) $\{w \in (a + b)^* \mid \#_b(w) \text{ is even}\}$ and $\{w \in (a + b)^* \mid \#_b(w) \text{ is odd}\}$
- (c) $\{w \in (a + b)^* \mid \#_a(w) = \#_b(w)\}$ and $\{w \in (a + b)^* \mid \#_a(w) \neq \#_b(w)\}$
- (d) $\{\epsilon\}, \{w a \mid w \in (a + b)^*\}$ and $\{wb \mid w \in (a + b)^*\}$ [2006 : 2 M]

1.37 Which of the following statements about regular languages is NOT true?

- (a) Every language has a regular superset
- (b) Every language has a regular subset
- (c) Every subset of a regular language is regular
- (d) Every subset of a finite language is regular [2006 : 2 M]

Directions for Q.1.38 to Q.1.39:

Let L be a regular language. Consider the constructions on L below:

- I. $\text{repeat}(L) = \{ww \mid w \in L\}$
- II. $\text{prefix}(L) = \{u \mid \exists v : uv \in L\}$
- III. $\text{suffix}(L) = \{v \mid \exists u : uv \in L\}$
- IV. $\text{half}(L) = \{u \mid v : |\exists v| = |u| \text{ and } uv \in L\}$

1.38 Which of the constructions could lead to a non-regular language?

- (a) Both I and IV
- (b) Only I
- (c) Only IV
- (d) Both II and III [2006 : 2 M]

1.39 Which choice of L is best suited to support your answer above?

- (a) $(a + b)^*$
- (b) $\{\epsilon, a, ab, bab\}$
- (c) $(ab)^*$
- (d) $\{a^n b^n \mid n \geq 0\}$

[2006 : 2 M]

1.40 If s is a string over $(0 + 1)^*$, then let $n_0(s)$ denote the number of 0's in s and $n_1(s)$ the number of 1's in s. Which one of the following languages is not regular?

- (a) $L = \{s \in (0 + 1)^* \mid n_0(s) \text{ is a 3-digit prime}\}$
- (b) $L = \{s \in (0 + 1)^* \mid \text{for every prefix } s' \text{ of } s, \mid n_0(s') - n_1(s') \mid \leq 2\}$
- (c) $L = \{s \in (0 + 1)^* \mid n_0(s) - n_1(s) \leq 4\}$
- (d) $L = \{s \in (0 + 1)^* \mid n_0(s) \bmod 7 = n_1(s) \bmod 5 = 0\}$

[2006 : 2 M]

1.41 Consider the regular language $L = (111 + 11111)^*$. The minimum number of states in any DFA accepting this languages is

- (a) 3
- (b) 5
- (c) 8
- (d) 9

[2006 : 2 M]

1.42 Which of the following is TRUE?

- (a) Every subset of a regular set is regular
- (b) Every finite subset of a non-regular set is regular
- (c) The union of two non-regular sets is not regular
- (d) Infinite union of finite sets is regular

[2007 : 1 M]

1.43 A minimum state deterministic finite automation accepting the language $L = \{w \mid w \in (0, 1)^*, \text{ number of 0s \& 1s in } w \text{ are divisible by 3 and 5, respectively}\}$ has

- (a) 15 states
- (b) 11 states
- (c) 10 states
- (d) 9 states

[2007 : 2 M]

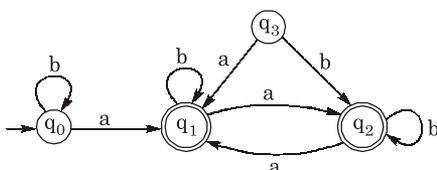
1.44 Which of the following languages is regular?

- (a) $\{ww^R \mid w \in \{0, 1\}^+\}$
- (b) $\{ww^R x \mid x, w \in (0, 1)^+\}$
- (c) $\{wxw^R \mid x, w \in (0, 1)^+\}$
- (d) $\{xww^R \mid x, w \in (0, 1)^+\}$

[2007 : 2 M]

Common Data for Q.1.45 & Q.1.46:

Consider the following Finite State Automation



1.45 The language accepted by this automaton is given by the regular expression

- (a) $b^* ab^* ab^* ab^*$
- (b) $(a + b)^*$
- (c) $b^* a (a + b)^*$
- (d) $b^* ab^* ab^*$

[2007 : 2 M]

1.46 The minimum state automaton equivalent to the above FSA has the following number of states

- (a) 1
- (b) 2
- (c) 3
- (d) 4

[2007 : 2 M]

1.47 The two grammars given below generate a language over the alphabet $\{x, y, z\}$

G1: $S \rightarrow x \mid z \mid xS \mid zS \mid yB$

$B \rightarrow y \mid z \mid yB \mid zB$

G2: $S \rightarrow y \mid z \mid yS \mid zS \mid xB$

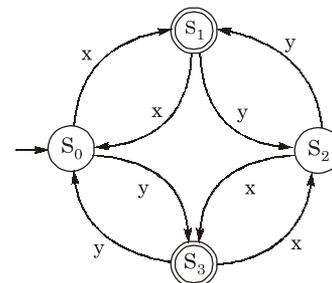
$B \rightarrow y \mid yS$

Which one of the following choices describes the properties satisfied by the strings in these languages?

- (a) G1 : No y appears before any x
G2 : Every x is followed by at least one y
- (b) G1 : No y appears before any x
G2 : No x appears before any y
- (c) G1 : No y appears after any x
G2 : Every x is followed by at least one y
- (d) G1 : No y appears after any x
G2 : Every y is followed by at least one x

[2007 : 2 M]

1.48 Consider the following DFA in which s_0 is the start state and s_1, s_3 are the final states.



What language does this DFA recognize?

- (a) All strings of x and y
- (b) All strings of x and y which have either even number of x and even number of y or odd number or x and odd number of y
- (c) All strings of x and y which have equal number of x and y
- (d) All strings of x and y with either even number of x and odd number of y or odd number of x and even number of y

[2007 : 2 M]

1.49 Consider the grammar given below:

$$\begin{aligned}
 S &\rightarrow xB \mid yA \\
 A &\rightarrow x \mid xS \mid yAA \\
 B &\rightarrow y \mid yS \mid yBB
 \end{aligned}$$

Consider the following strings.

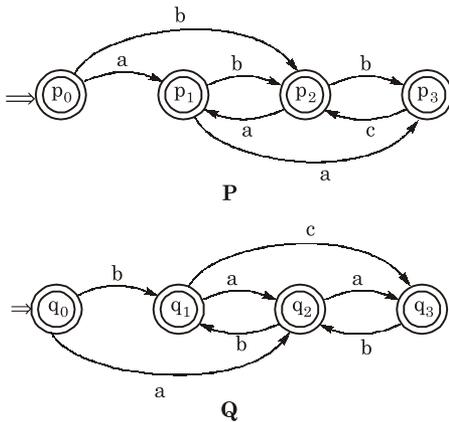
- (i) xxyyx (ii) xxyyxy
- (iii) xyxy (iv) yxxy
- (v) yxx (vi) yxx

Which of the above strings are generated by the grammar?

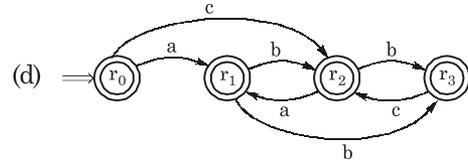
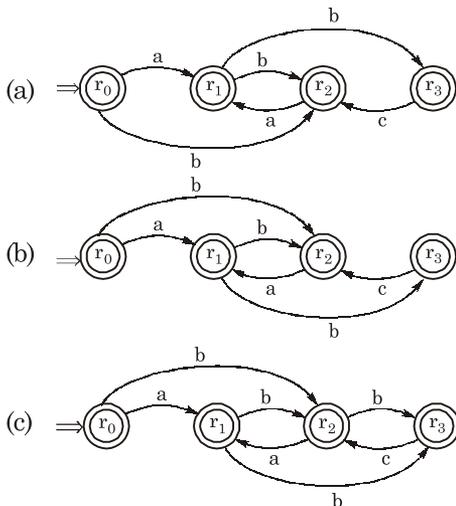
- (a) (i), (ii), and (iii)
- (b) (ii), (v), and (vi)
- (c) (iii) and (iv)
- (d) (i), (iii), and (iv)

[2007 : 2 M]

1.50 Consider the following finite automata P and Q over the alphabet {a, b, c}. The start states are indicated by a double arrow and final states are indicated by a double circle. Let the languages recognized by them be denoted by L(P) and L(Q) respectively.



The automaton which recognizes the language $L(P) \cap L(Q)$ is :



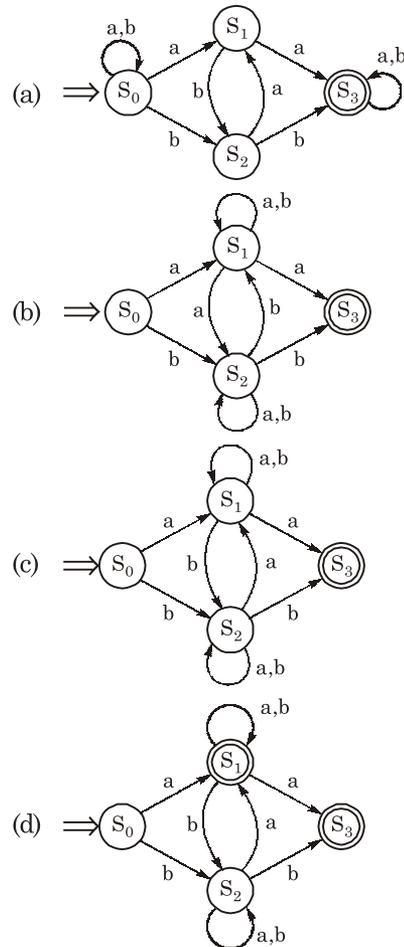
(e) None of the above [2007 : 2 M]

Common Data for Q.1.51 to Q.1.53:

Consider the regular expression:

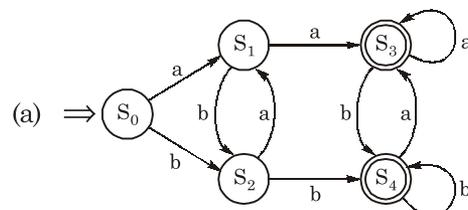
$$R = (a + b)^* (aa + bb) (a + b)^*$$

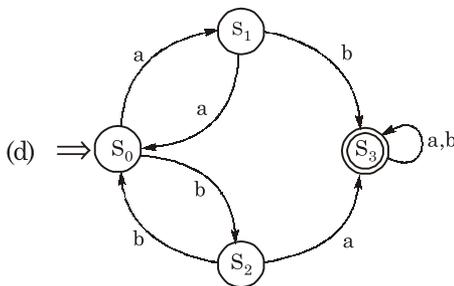
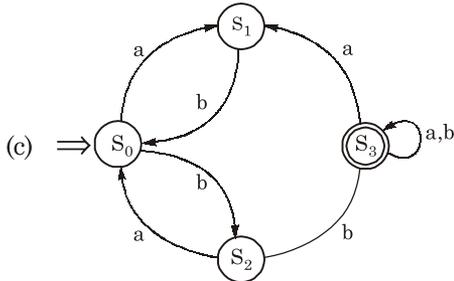
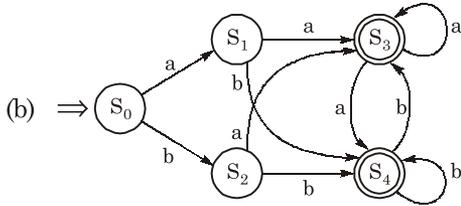
1.51 Which of the following non-deterministic finite automata recognizes the language defined by the regular expression R? Edges labeled λ denote transitions on the empty string.



[2007 : 2 M]

1.52 Which deterministic finite automaton accepts the language represented by the regular expression R?





[2007 : 2 M]

1.53 Which one of the regular expressions given below defines the same language as defined by the regular expression R ?

- (a) $(a(ba)^* + b(ab)^*)(a + b)^+$
- (b) $(a(ba)^* + b(ab)^*)^*(a + b)^*$
- (c) $(a(ba)^*(a + bb) + b(ab)^*(b + aa))(a + b)^*$
- (d) $(a(ba)^*(a + bb) + b(ab)^*(b + aa))(a + b)^+$

[2007 : 2 M]

1.54 Which of the following regular expressions describes the language over $\{0, 1\}$ consisting of strings that contain exactly two 1's?

- (a) $(0 + 1)^* 11(0 + 1)^*$
- (b) $0^* 110^*$
- (c) $0^* 10^* 10^*$
- (d) $(0 + 1)^* 1(0 + 1)^* 1(0 + 1)^*$

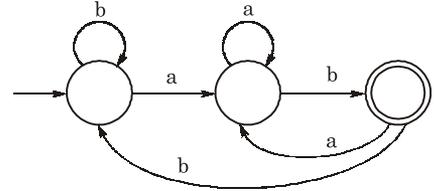
[2008 : 1 M]

1.55 Let N be an NFA with n states and let M be the minimized DFA with m states recognizing the same language. Which of the following is NECESSARILY true?

- (a) $m \leq 2^n$
- (b) $n \leq m$
- (c) M has one accept state
- (d) $m = 2^n$

[2008 : 1 M]

1.56 If the final states and non-final states in the DFA below are interchanged, then which of the following languages over the alphabet $\{a, b\}$ will be accepted by the new DFA?

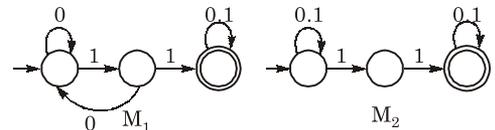


- (a) Set of all strings that do not end with ab
 - (b) Set of all strings that begin with either an a or ab
 - (c) Set of all strings that do not contain the substring ab ,
 - (d) The set described by the regular expression $b^*aa^*(ba)^*b^*$
- [2008 : 2 M]

1.57 Which of the following languages is (are) non-regular?

- $L_1 = \{0^m 1^n \mid 0 \leq m \leq n \leq 10000\}$
 - $L_2 = \{w \mid w \text{ reads the same forward and backward}\}$
 - $L_3 = \{w \in \{0, 1\}^* \mid w \text{ contains an even number of 0's and an even number of 1's}\}$
 - (a) L_2 and L_3 only
 - (b) L_1 and L_2 only
 - (c) L_3 only
 - (d) L_2 only
- [2008 : 2 M]

1.58 Consider the following two finite automata. M_1 accepts L_1 and M_2 accepts L_2 . Which one of the following is TRUE?



- (a) $L_1 = L_2$
- (b) $L_1 \cap L_2 = \phi$
- (c) $L_1 \cap \overline{L_2} = \phi$
- (d) $L_1 \cup L_2 \neq L_1$

[2008 : 2 M]

1.59 Given below are two finite state automata (\rightarrow indicates the start and F indicates a final state)

Y :		a	b
	$\rightarrow 1$	1	2
	2(F)	2	1

Z :		a	b
	$\rightarrow 1$	2	2
	2(F)	1	1

Which of the following represents the product automaton $Z \times Y$?

- (a)

	a	b
$\rightarrow P$	S	R
Q	R	S
R(F)	Q	P
S	P	Q
- (b)

	a	b
$\rightarrow P$	S	Q
Q	R	S
R(F)	Q	P
S	P	Q

(c)

	a	b
→ P	Q	S
Q	R	S
R(F)	Q	P
S	Q	P

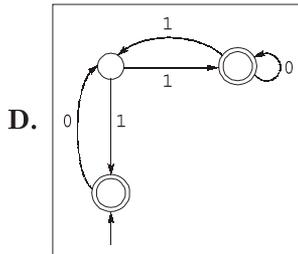
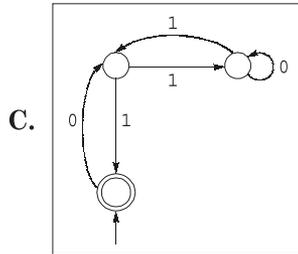
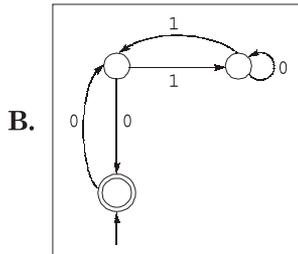
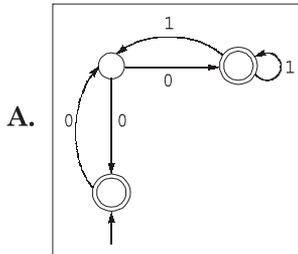
(d)

	a	b
→ P	S	Q
Q	S	R
R(F)	Q	P
S	Q	P

[2008 : 2 M]

1.60 Match List-I with List-II and select the correct answer using the codes given below the lists:

List-I:



List-II

- $\epsilon + 0(01^*1 + 00)^*01^*$
- $\epsilon + 0(10^*1 + 00)^*0$
- $\epsilon + 0(10^*1 + 10)^*1$
- $\epsilon + 0(10^*1 + 10)^*10^*$

Codes:

A B C D

- 2 1 3 4
- 1 3 2 4
- 1 2 3 4
- 3 2 1 4

[2008 : 2 M]

1.61 Which of the following are regular sets?

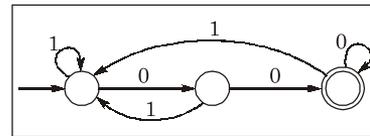
- $\{a^n b^{2m} \mid n \geq 0, m \geq 0\}$
- $\{a^n b^m \mid n = 2m\}$
- $\{a^n b^m \mid n \neq m\}$
- $\{xycy \mid x, y \in \{a, b\}^*\}$

- 1 and 4 only
 - 1 and 3 only
 - 1 only
 - 4 only
- [2008 : 2 M]

1.62 Which one of the following languages over the alphabet $\{0, 1\}$ is described by the regular expression: $(0 + 1)^*0(0 + 1)^*0(0 + 1)^*$

- The set of all strings containing the substring 00
 - The set of all strings containing at most two 0's
 - The set of all strings containing at least two 0's
 - The set of all strings that begin and end with either 0 or 1
- [2009 : 1 M]

1.63 The following DFA accepts the set of all strings over $\{0, 1\}$ that



- begin either with 0 or 1
 - end with 0
 - end with 00
 - contain the substring 00
- [2009 : 2 M]

1.64 Let $L = \{\omega \in (0 + 1)^* \mid \omega \text{ has even number of 1s}\}$, i.e., L is the set of all bit strings with even number of 1s. Which one of the regular expressions below represents L ?

- $(0^*10^*1)^*$
- $0^*(10^*10^*)^*$
- $0^*(10^*1)^*0^*$
- $0^*1(10^*1)^*10^*$

[2010 : 2 M]

1.65 Let ω be any string of length n in $\{0, 1\}^*$. Let L be the set of all substrings of ω . What is the minimum number of states in a non-deterministic finite automaton that accepts L ?

- $n - 1$
 - n
 - $n + 1$
 - 2^{n+1}
- [2010 : 2 M]

1.66 The lexical analysis for a modern computer language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

- Finite state automata
 - Deterministic pushdown automata
 - Non-deterministic pushdown automata
 - Turing machine
- [2011 : 1 M]

Answers Finite Automata : Regular Languages

1.1 (a)	1.2 (b)	1.3 (c)	1.4 (d)	1.5 (b)	1.6 (d)	1.7 (d)	1.8 (a)	1.9 (c, d)
1.10 (d)	1.11 (b)	1.12 (b)	1.13 (c)	1.14 (b)	1.15 (d)	1.16 (a)	1.17 (b)	1.18 (d)
1.19 (d)	1.20 (a)	1.21 (b)	1.22 (c)	1.23 (b)	1.24 (c)	1.25 (b)	1.26 (a)	1.27 (b)
1.28 (a)	1.29 (a)	1.30 (b)	1.31 (d)	1.32 (b)	1.33 (b)	1.34 (d)	1.35 (a)	1.36 (a)
1.37 (c)	1.38 (b)	1.39 (a)	1.40 (c)	1.41 (d)	1.42 (b)	1.43 (a)	1.44 (c)	1.45 (c)
1.46 (b)	1.47 (a)	1.48 (d)	1.49 (c)	1.50 (b)	1.51 (a)	1.52 (a)	1.53 (c)	1.54 (c)
1.55 (a)	1.56 (a)	1.57 (d)	1.58 (a)	1.59 (a)	1.60 (c)	1.61 (a)	1.62 (c)	1.63 (c)
1.64 (b)	1.65 (c)	1.66 (a)	1.67 (c)	1.68 (b)	1.69 (a)	1.70 (b)	1.71 (c)	1.72 (d)
1.73 (a)	1.74 (d)	1.75 (c)	1.76 (a)	1.77 (b)	1.78 (a)	1.79 (a)	1.80 (3)	1.81 (1)
1.82 (c)	1.83 (a)	1.84 (3)	1.85 (b)	1.86 (d)	1.87 (c)	1.88 (b)	1.89 (2)	1.90 (b)
1.91 (4)	1.92 (d)	1.93 (8)	1.94 (b)	1.95 (c)	1.96 (d)	1.97 (2)	1.98 (b)	1.99 (d)
1.100 (2)	1.101 (a)	1.102 (*)	1.103 (6)	1.104 (a)	1.105 (a)	1.106 (256)	1.107 (a, b, c)	1.108 (d)
1.109 (c)	1.110 (c)	1.111 (4)	1.112 (a, c)	1.113 (44)	1.114 (d)	1.115 (15)	1.116 (a)	

Explanations Finite Automata : Regular Languages

1.1 (a)

$$L = \{x^n y^n \mid n \geq 1\}$$

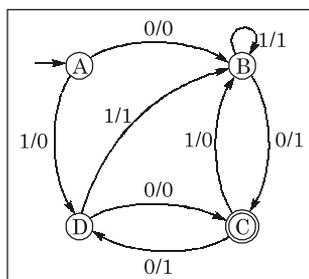
The language L produces all strings having equal number of x and y and y follows the x.

- (i) $E \rightarrow x E y \mid xy$: The language produced by above grammar contain all strings having equal no. of x & y and y follows x.
- (ii) $xy \mid x^+xy y^+$: The above can generate string xxxxyy which doesn't contain equal no. of x and y
- (iii) x^+y^+ : The above regular expression generates all strings in which y follows x and contain x and y greater than one. Above RE can generate string xxxy which doesn't contain equal no of x & y.

So only (i) generates the same language as L.

1.2 (b)

The state diagram which represents the given state table is



From state A to reach C, the shortest sequence is

$$A \xrightarrow{0} B \xrightarrow{0} C$$

$$A \xrightarrow{1} D \xrightarrow{0} C$$

From B to reach C

$$B \xrightarrow{0} C$$

$$B \xrightarrow{1} D \xrightarrow{0} C$$

From C to reach C

$$C \xrightarrow{0} D \xrightarrow{0} C$$

$$C \xrightarrow{1} B \xrightarrow{0} C$$

From D to reach C

$$D \xrightarrow{0} C$$

$$D \xrightarrow{1} B \xrightarrow{0} C$$

So 10 is the shortest input sequence to reach the final state C, whatever be the initial state.

1.3 (c)

- (i) $(00)^*(\epsilon + 0)$
 $\equiv (00)^* \epsilon + (00)^* 0 = 0^*$
 Even number of 0's and Odd number of 0's
 i.e., any number of 0's
 - (ii) $(00)^* \equiv$ Even number of 0's
 - (iii) $0^* \equiv$ any number of 0's
 - (iv) $0(00)^* \equiv$ Odd number of 0's
- So (i) and (iii) are same.

1.4 (d)

- (a) $L = \{x \mid x \text{ has an equal number of a's and b's}\}$ is context free language (since there is comparison between number of a's and number of b's).
- (b) $L = \{a^n b^n \mid n \geq 1\}$ is context free language (since there is comparison between number of a's and number of b's).
- (c) $L = \{x \mid x \text{ has more a's than b's}\}$ is context free language (since there is comparison between number of a's and number of b's).
- (d) $L = \{a^m b^n \mid m \geq 1, n \geq 1\}$ is regular (since m and n are independent and hence there is no comparison). The regular expression is aa^*bb^* .

1.5 (b)

- (a) The set of all strings over Σ is Σ^* which is countably infinite.
- (b) Set of all languages over Σ is 2^{Σ^*} . According to Cantor's theorem if S be an countably infinite set, then its power set 2^S is uncountable. So 2^{Σ^*} is uncountable because Σ^* is countably infinite.
- (c) Set of all regular languages over Σ is countably infinite.
- (d) Set of all languages over Σ accepted by Turing machine is the set of all RE languages which is countably infinite.

1.6 (d)

- (a) r.e. (regular expression) = $0^*(1 + 0)^*$ can generate string 100, which contains substring 100
- (b) r.e. (regular expression) = $0^* 1010^*$ can generate string 10100, which contain 100 as a substring. Also, this regular expression cannot generate ϵ which is in the given language.
- (c) r.e. (regular expression) = $0^*1^*01^*$ generates strings which doesn't contain 100 as substring. However, ϵ is the smallest string which doesn't contain 100 as substring but above RE can't generate ϵ .
- (d) r.e. (regular expression) = $0^*(10 + 1)^*$ generates all strings which doesn't contain 100 as substring.

1.7 (d)

So, $(a^*b^*)^* \equiv (a + b)^*$
 $B = ((01)^* 1^*)^*$

So,

$$\equiv (01 + 1)^* \equiv A$$

$$A = B$$

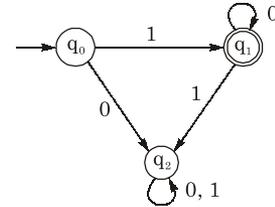
1.8 (a)

The numbers 1, 2, 4, 8, ..., 2^n ... are represented by 1, 10, 100, 1000, ...

The pattern in the regular expression is 1 followed by 0's.

The regular expression for above is 10^*

The DFA for above language is



So the numbers 1, 2, 4, 8, ..., 2^n ... written in binary can be recognized by a deterministic finite state automaton.

1.9 (c, d)

- (c) The string 1101 doesn't belong to set represented by $(10)^*(01)^*(00 + 11)^*$ because once 11 appears in string then 1 and 0 only appears in pairs.
- (d) $(00 + (11)^* 0)^*$ can generate only strings with even number of 1's and hence cannot generate 1101.

1.10 (d)

Number of substrings (of all lengths inclusive) that can be formed from a character strings of

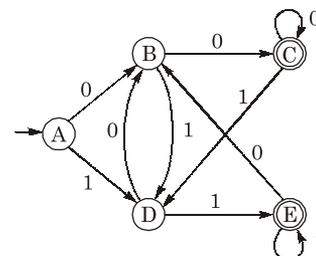
$$\text{length } n \text{ is } \frac{n(n+1)}{2} + 1.$$

Since we do not want to count the substring of zero length (i.e. null string), the number of

$$\text{substrings becomes } \frac{n(n+1)}{2}.$$

1.11 (b)

The number of states in the minimum state deterministic finite state automation accepting all binary strings whose last two symbols are the same is 5.



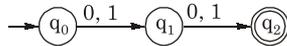
1.12 (b)

The minimum state finite automation that recognizes the language represented by regular expression $(0 + 1) (0 + 1) \dots n$ times is $n + 1$.

This language contains strings with exactly length n .

$(n + 1)$ states are required to count length upto n . No trap state is required since we are making minimal FA, not minimal DFA.

For example, for $n = 2$ the design is shown below.



1.13 (c)

According to rules of regular expressions

$$(r_1 + r_2)^* \equiv (r_1 + r_2^*)^*$$

Therefore $(a + b^*)^* \equiv (a + b)^*$

So $S = T$

1.14 (b)

The language generated by the grammar

$$S \rightarrow 0S0 \mid 00 \text{ is}$$

$$L = \{0^2, 0^4, 0^6, 0^8, \dots\}$$

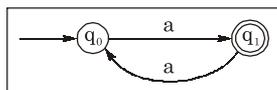
$$= \{0^{2n+2} \mid n \geq 0\}$$

$$\Rightarrow = \{0^{2n} \mid n \geq 1\} = 00(00)^*$$

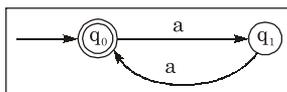
So above language is regular but not 0^+ .

1.15 (d)

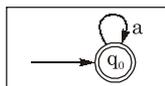
$$L = \{a^n \mid n \text{ is odd}\}$$



$$L = \{a^n \mid n \text{ is even}\}$$



$$L = \{a^n \mid n \geq 0\}$$



So for two states minimal finite state automation, L can be $\{a^n \mid n \text{ is odd}\}$ or L can be $\{a^n \mid n \text{ is even}\}$

1.16 (a)

$$L_1 = \{0^{2n} \mid n \geq 1\}$$

L_1 produces language having even number of 0's which is regular language.

Regular expression for S_1 is $00(00)^*$

$L_2 = \{0^m 1^n 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1\}$ is context free language not regular. (m occurs in two places, so there is comparison of count).

So, S_1 is correct but S_2 is not correct.

1.17 (b)

For an arbitrary NFA with N states, the maximum number of states in an equivalent minimized DFA is 2^N .

1.18 (d)

A DFA over $\Sigma = \{a, b\}$ accepting all strings which have no. of a's divisible by 6 and number of b's divisible by 8 is a grid machine (product automata) having $6 \times 8 = 48$ states.

1.19 (d)

$$L_1 = \{ww \mid w \in \{a, b\}^*\}$$

is context sensitive language (CSL) (since there is infinite string matching in straight order).

$$L_2 = \{ww^R \mid w \in (a, b)^*, w^R$$

is the reverse of $w\}$

is context free language (since there is infinite string matching in reverse order).

$$L_3 = \{0^{2i} \mid i \text{ is an integer}\} = (00)^*$$

is regular language which contains all strings having even number of 0's.

$$L_4 = \{0^{i^2} \mid i \text{ is an integer}\}$$

is context sensitive language (CSL) (since the power is infinite and non linear).

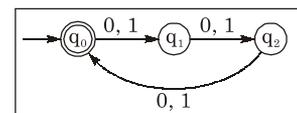
1.20 (a)

The state diagram represents the FSM which outputs the sum of the present and previous bits of the input.

State A represents previous bit is a 0 and B and C represents previous bit is a 1.

1.21 (b)

The minimal finite automation with 3 states which accepts the language $L = \{x \mid \text{length of } x \text{ is divisible by } 3\}$ is as follows:



1.22 (c)

The given bit pattern can be represented as:

1 — 1 — 1

The four blanks can be filled in $2^4 = 16$ ways. Therefore there are 16 such strings in this pattern. Not all of these are accepted by the machine. The strings and its acceptance is given below:

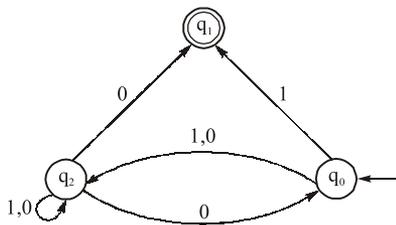
					accepted		
1	0	0	1	0	0	1	✓
1	0	0	1	0	1	1	✓
1	0	0	1	1	0	1	✓
1	0	0	1	1	1	1	✓
1	0	1	1	0	0	1	✓
1	1	0	1	0	0	1	✓
1	1	1	1	0	0	1	✓

Only these seven strings given above are accepted. The other strings (9 of them) in this pattern are rejected, since they don't reach the final state.

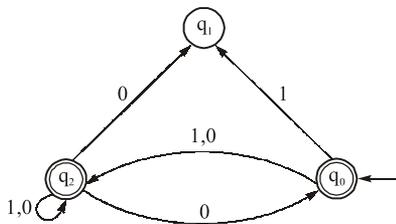
∴ Correct answer is (c).

1.23 (b)

The given machine M is



Now the complementary machine \bar{M} is



In the case of dfa, $L(\bar{M}) = \overline{L(M)}$ but in the case of nfa this is not true. In fact $L(\bar{M})$ and $L(M)$ have no connection.

∴ To find $L_1 = L(\bar{M})$ we have to look at \bar{M} and directly find its language.

$$L_1 = L(\bar{M}) = \epsilon + (0 + 1)(0 + 1)^* + \dots = (0 + 1)^* + \dots = (0 + 1)^*$$

1.24 (c)

In choices (a), (b) and (d), inside the parenthesis we can generate "a", "b" and "c" separately and hence all three are same as $(a + b + c)^*$.

In choice (c) the strings "a" and "b" cannot be generated separately since "ab" is always together.

So, choice (c) is not same as $(a + b + c)^*$.

1.25 (b)

$$\delta(A, a) = A \equiv A \rightarrow aA$$

$$\delta(A, b) = B \equiv A \rightarrow bB$$

$$\delta(B, a) = B \equiv B \rightarrow aB$$

$$\delta(B, b) = A \equiv B \rightarrow bA$$

Since B is final state, so we need to put $B \rightarrow \epsilon$. So the correct grammar is choice (b) which is $\{A \rightarrow aA, A \rightarrow bB, B \rightarrow aB, B \rightarrow bA, B \rightarrow \epsilon\}$.

1.26 (a)

The given finite state machine accepts any string $w \in (0, 1)^*$ in which the number of 1s is multiple of 3 and the number of 0s is multiple of 2.

1.27 (b)

Given regular expression is infinite set (because of $*$) of finite strings. A regular expression cannot generate any infinite string (since string is always finite in length by definition).

1.28 (a)

Given language is finite. Hence it is regular language.

1.29 (a)

Writing Y and Z in terms of incoming arrows (Arden's method), we get

$$Y = X0 + Y0 + Z1$$

$$Z = X0 + Z1 + Y0$$

Clearly, $Y = Z$

1.30 (b)

The given grammar after substitution of X and Y becomes

$$S \rightarrow Zaa \mid Waa$$

$$Z \rightarrow Sa \mid \epsilon$$

$$W \rightarrow Sa$$

Which after substituting Z and W is equivalent to $S \rightarrow Saaa \mid aa \mid Saaa$.

Which is equivalent to $S \rightarrow Saaa \mid aa$.

So, $L(G) = (aaa)^*aa$

So the language generated by the grammar is the set of strings with a 's such that number of $a \pmod 3$ is 2. So the number of states required should be 3 to maintain the count of number of a 's $\pmod 3$.