Fully Solved

2650 MCQs

Useful For

GATE and PSUs

Computer Science and Information Technology

MADE EASY
Publications

MADE EASY Publications

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016
E-mail: infomep@madeeasy.in
Contact: 011-45124660, 0-8860378007
Visit us at: www.madeeasypublications.org

2650 Multiple Choice Questions for GATE and PSUs : Computer Science and Information Technology

First Edition: 2018

Published by: MADE EASY Publications, New Delhi-110016

PREFACE

B. Singh (Ex. IES)

It gives me great happiness to introduce the **First Edition** on Computer Science and Information Technology containing nearly 2650 MCQs which focuses in-depth understanding of subjects at basic and advanced level which has been segregated topicwise to disseminate all kind of exposure to students in terms of quick learning and deep apt. The topicwise segregation has been done to align with contemporary competitive examination pattern. Attempt has been made to bring out all kind of probable competitive questions for the aspirants preparing for GATE and PSUs. The content of this book ensures threshold level of learning and wide range of practice questions which is very much essential to boost the exam time confidence level and ultimately to succeed in all prestigious engineer's examinations. It has been ensured from MADE EASY team to have broad coverage of subjects at chapter level.

While preparing this book utmost care has been taken to cover all the chapters and variety of concepts which may be asked in the exams. The solutions and answers provided are upto the closest possible accuracy. The full efforts have been made by MADE EASY Team to provide error free solutions and explanations.

I have true desire to serve student community by way of providing good sources of study and quality guidance. I hope, this book will be proved an important tool to succeed in competitive examinations. 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

CONTENTS

■■■■

CHAPTER 1

Basic Level

Theory of Computation

1. Finite Automata: Regular Languages

Q.1 Which of the following is false?

(a) The languages accepted by FA's are regular languages.

(b) Every DFA is an NFA.

(c) There are some NFA's for which no DFA can be constructed.

(d) If L is accepted by an NFA with $\in$ transition then L is accepted by an NFA without $\in$ transition.

Q.2 Let $r_1 = ab^*c^*$ and $r_2 = (a^*b+c)^*$ and $r_3 = (a+b+c)^*$. Then which of the following is true

(a) w = 'ac' belongs to $L(r_2)$ and $L(r_3)$ but not $L(r_1)$

(b) w = 'ac' belongs to $L(r_3)$ only

(c) w = 'ac' belongs to $L(r_1)$, $L(r_2)$ and $L(r_3)$

(d) w = 'ac' belongs to $L(r_1)$ and $L(r_3)$ but not $L(r_2)$

Q.3 Let $\Sigma = \{a, b\}$, $r_1 = a(a+b)^*$ and $r_2 = b(a+b)^*$. Which of the following is true?

(a) $L(r_1) = L(r_2) = \Sigma^*$

(b) $L(r_1) \cap L(r_2) = \{\in\}$

(c) $L(r_1) \cup L(r_2) = \Sigma^*$

(d) $L(r_1) \cup L(r_2) \cup \{\in\} = \Sigma^*$

Q.4 Which of the following statements are true?

(i) $abcd \in L((b^*a^*)^*(d^*c^*)^*)$

(ii) $abcd \in L((d^*c^*b^*a^*)^*)$

(iii) $abcd \in L((a^*b^*a^*c^*d^*)^*)$

(a) (i) and (iii) only (b) (ii) and (iii) only

(c) (i) and (ii) only (d) All of these

Q.5 Which of the following are regular languages?

(i) The language $\{w \mid w \in \{a, b\}^*$, w has an odd number of b's$\}$.

(ii) The language $\{w \mid w \in \{a, b\}^*$, w has an even number of b's$\}$.

(iii) The language $\{w \mid w \in \{a, b\}^*$, w has an even number of b's and odd number of a's$\}$.

(a) (i) and (ii) only (b) (i) only

(c) (ii) only (d) All of these

Q.6 Which of the following regular expression corresponds to the language of all strings over the alphabet $\{a, b\}$ that contains exactly two a's

(i) aa

(ii) ab^*a

(iii) b^*ab^*a

(a) (i) and (ii) only (b) (ii) and (iii) only

(c) (i) and (iii) only (d) None of these

Q.7 Which of the following regular expression corresponds to the language of all strings over the alphabet $\{a, b\}$ that do not end with ab?

(a) $(a+b)^*(aa+ba+bb)$

(b) $(a+b)^*(aa+ba+bb)+a+b+\in$

(c) b^*ab^*a

(d) b^*aab^*

Q.8 What is regular expression corresponding to the language of strings of even lengths over the alphabet of $\{a, b\}$?

(a) $(aa+bb+ba+ab)^*$

(b) $(aa+bb)^*$

(c) $(ab+bb+ba)^*$

(d) $a^*b^*a^*b^*$

Q.9 How many minimum number of states will be there in the DFA accepting all strings (over the alphabet $\{a,b\}$) that do not contain two consecutive a's?

(a) 2 (b) 3

(c) 4 (d) 5

Q.10 How many minimum number of states are required in the DFA (over the alphabet $\{a, b\}$) accepting all the strings with the number of a's divisible by 4 and number of b's divisible by 5?

(a) 20 (b) 9

(c) 7 (d) 15

Q.11 Which of the following definitions below generates the same language as L where $L = \{x^n\ y^n | n >= 1\}$?

(i) $E \rightarrow xEy | xy$

(ii) $xy | (x^+\ y^+)$

(iii) $x^+\ y^+$

(a) (i) only (b) (i) and (ii) only

(c) (ii) and (iii) (d) (i) and (iii) only

Q.12 Let $X = \{0, 1\}$, $L = X^*$ and $R = \{0^n 1^n | n > 0\}$ then the language $L \cup R$ and R respectively

(a) Regular, Regular

(b) None regular, Regular

(c) Regular, Not regular

(d) Not regular, Not regular

Q.13 How many states does the DFA constructed for The set of all strings ending with "00", have?

(a) 2 (b) 3

(c) 4 (d) 5

Q.14 Which of the following identities are correct?

(a) $rs^* = rss^*$ (b) $(r^*s^*) = (r + s)^*$

(c) $(r + s)^* = r^* + s^*$ (d) $(r^*s^*)^* = (r + s)^*$

Q.15 Let L_1 and L_2 are regular sets defined over alphabet Σ^*. Mark the false statement

(a) $L_1 \cup L_2$ is regular (b) $L_1 \cap L_2$ is not regular

(c) $\Sigma^* - L_1$ is regular (d) L_1^* is regular

Q.16 Consider $L_1 = \{0^n\ 1^n | n \geq 1\}$, $L_2 = \{0^n\ c1^n | n \geq 1\}$

(i) L_1 and L_2 are accepted by non-deterministic PDA.

(ii) L_1 and L_2 are accepted by deterministic PDA.

(iii) Only L_2 is accepted by deterministic PDA.

Which of the following statements are correct?

(a) Only (i) (b) (i) and (ii)

(c) (i) and (iii) (d) All (i), (ii), (iii)

Q.17 Suppose $r_1 = \in$, $r_2 = 0^*\ 1^*$, which of the following statement is true about r_1 and r_2?

(a) r_1 is not regular expression, while r_2 is a regular

(b) r_1 and r_2 both are regular expression

(c) r_1 is regular expression but r_2 not

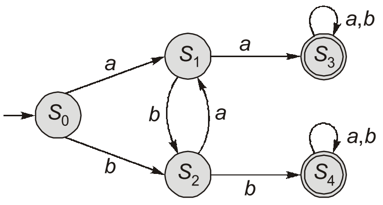
(d) Neither r_1 nor r_2 are regular expressions

Q.18 The transition function of DFA from one state to another on a given input symbol w is a function $Q \times \Sigma^*$ to

(a) 2^Q (b) Q

(c) Q' (d) Q^2

Q.19 Consider the machine M shown below:



$L(M) = ?$

(a) $L(M)$ = {words starting with aa or bb}

(b) $L(M)$ = {words ending with aa or bb}

(c) $L(M)$ = {words containing aa or bb as a subword}

(d) None of these

Q.20 Let $L = \{w | w \text{ has } 3k + 1\ b\text{'s}\ \forall k \geq 0\}$, construct a minimized finite automata D accepting L. How many states are there in D?

(a) 4

(b) 3

(c) 2

(d) The language is not regular

Q.21 Which of the following is undecidable?

(a) Equivalence of regular languages

(b) Equivalence of context free languages

(c) Finiteness check on context free language

(d) Emptiness of regular languages

Q.22 Let, $L_1 = \{a^n\ b^n\ c^n | n \geq 0\}$

$L_2 = \{a^{2n}\ b^{2n}\ c^{2n} | n \geq 0\}$

$L_3 = \{a^{2n}\ b^{2n}\ c^n | n \geq 0\}$

(a) $L_1 \subseteq L_2$ and $L_3 \subseteq L_2$

(b) $L_2 \subseteq L_1$ and $L_2 \subseteq L_3$

(c) $L_2 \subseteq L_1$ but $L_2 \not\subset L_3$

(d) $L_1 \subseteq L_2$ and $L_2 \subseteq L_3$

Q.23 Suppose $L_1 = \{10, 1\}$ and $L_2 = \{011, 11\}$. How many distinct elements are there in $L = L_1L_2$.

(a) 4 (b) 3

(c) 2 (d) None of these

Q.24 Which of the following regular expression does not represent strings beginning with atleast one 0 and ends with at least one 1?

(a) 0^*1^* (b) $00^*(0 + 1)^*1$

(c) $0(0 + 1)^*1$ (d) None of these

Q.25 Strings generated by $(1 + 01)^*$ does not contain the substring,

(a) 10 (b) 11

(c) 01 (d) 00

Q.26 The following CFG:

$$S \to aS \mid bS \mid a \mid b$$

is equivalent to the regular expression

1. $(a^* + b)^*$ **2.** $(a + b)^+$

3. $(a + b)\,(a + b)^*$ **4.** $(a + b)^*\,(a + b)$

(a) 2 and 3 only (b) 2, 3 and 4

(c) All of the above (d) 3 and 4 only

Q.27 The Moore machine has six tuples

$$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

Which of the following is true?

(a) δ is the output function

(b) δ is the transition function $\Sigma \to Q$

(c) λ is the transition function $\Sigma \times Q \to Q$

(d) λ is the output function mapping $Q \to \Delta$

Q.28 For the previous question, δ is the transition function from

(a) $Q \to R$ (b) $\delta \times \Sigma \to d$

(c) $\Sigma \times Q \to Q$ (d) None of these

Q.29 The regular expression for "Binary numbers that are multiples of two" is

(a) $(0 + 1)^*.1$ (b) $(0 + 1)^*.0$

(c) $(1 + 0)^*.1$ (d) $(1 + 0)^*.00$

Q.30 The regular expression for "strings of *a*'s and *b*'s containing two consecutive *a*'s" is

(a) $(a + b)^*\,ab(a + b)^*$

(b) $(a+b)^*\,bb(a + b)^*$

(c) $(a + b)^*\,aa(a + b)^*$

(d) $(a + b)^*\,ba(a + b)^*$

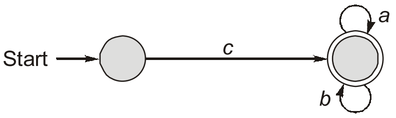
Q.31 The regular expression that describe the language generated by the grammar:

$$G = (\{T,Z\},\{a,b\}\,Z,\{Z \to aZ \mid \in \mid, Z \to bT,\ T \to aZ\})$$

(a) ab^*a^* (b) a^*ba^*b

(c) $ab^*a\,a$ (d) $(a + ba)^*$

Q.32 Which of the string is accepted by given NDFA?



(a) $c.(a + b)^*$ (b) $c.a^*b^*$

(c) $c.(ab)^*$ (d) None of these

Q.33 The regular expression $(a \mid b)\,(a \mid b)$ denotes the set

(a) $\{a, b, ab, aa\}$ (b) $\{a, b, ba, bb\}$

(c) $\{a, b\}$ (d) $\{aa, ab, ba, bb\}$

Q.34 Let *a* and *b* the regular expressions then $(a^* + b^*)^*$ is not equivalent to

(a) $(b^* + a^*)^*$ (b) $(a + b)^*$

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

Q.35 Consider the following regular expression:

$$R = (ab \mid abb)^*\,bbab$$

Which of the following strings is NOT in the set denoted by *R*?

(a) *abababb* (b) *ababbabbbab*

(c) *bbab* (d) *abbabbbab*

Q.36 The following GFG:

$S \to aS \mid bS \mid a \mid b$ and $S \to aS \mid bS \mid a \mid b \mid \in$ is equivalent to regular expressions

(a) $(a + b)$ and $\in + a + b$ respectively

(b) $(a + b)\,(a + b)^*$ and $(a + b)^*$ respectively

(c) $(a + b)\,(a + b)$ and $(\in + a + b)\,(\in + a + b)$ respectively

(d) None of the above

Q.37 Which of the following pairs of regular expression are not equivalent?

(a) $(ab)^*\,a$ and $a(ba)^*$

(b) $(a + b)^*$ and $(a^* + b)^*$

(c) $b^*\,ab^*$ and $a^*\,ba^*$

(d) All of the above

Q.38 Can a DFA simulate NFA?

(a) No (b) Yes

(c) Some time (d) Depends on NFA

Q.39 The basic limitation of FSM is that

(a) It can not remember arbitrary large amount of information.

(b) It sometimes fails to recognize grammars that are not regular.

(c) It sometimes fails to recognize grammars that are regular.

(d) All of these

Q.40 Any given transition diagram has an equivalent

(a) Regular expression

(b) NDFSM

(c) DFSM

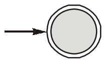
(d) All of these

Q.41 The set $\{a^n\,b^n \mid n = 1, 2, 3, \ldots\}$ can be generated by the CFG

(a) $S \to ab \mid aSb \mid \in$ (b) $S \to aaSbb \mid ab$

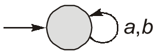
(c) $S \to ab \mid aSb$ (d) None of these

Q.42 The FSM shown in the figure accepts



(a) All strings (b) No strings
(c) ∈-alone (d) None of these

Q.43 The FSM shown in the figure accepts



(a) All strings (b) No strings
(c) ∈-alone (d) None of these

Q.44 Consider the two regular languages:

$$L_1 = (a+b)^* a \text{ and } L_2 = b(a+b)^*$$

The intersection of L_1 and L_2 is given by

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

Q.45 The string 1101 does not belong to the set represented by

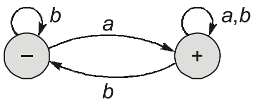
(a) $110^*(0+1)$ (b) $1(0+1)^*101$
(c) $(00+(11)^*0)^*1$ (d) $(10)^*(01)^*(00+11)^*$

Q.46 Which two of the following four regular expressions are equivalent? (∈ is the empty string)?

1. $(00)^*(\in+0)$
2. $(00)^*$
3. 0^*
4. $0(00)^*$

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

Q.47 Consider the FA shown in the figure given below, where "–" is the start and "+" is the ending state. The language accepted by the FA is



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

Q.48 Consider the following table of an FA:

δ	a	b
Start	q_1	q_0
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_4	q_3
q_4	q_4	q_4

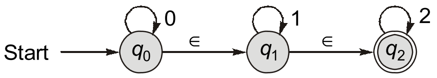
If the final state is q_4, the which of the following strings will be accepted?

1. *aaaaa*
2. *aabbaabbbbb*
3. *bbababababbb*

(a) 1 and 2 (b) 2 and 3
(c) 3 and 1 (d) All of these

Directions for Question 49 and 50:

Consider the following NFA with ∈ moves.



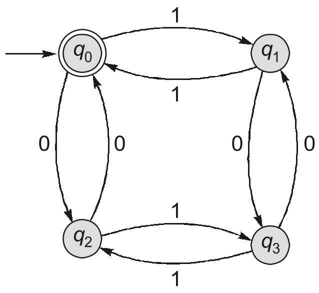
Q.49 If the given NFA is converted to NFA without ∈-moves, which of the following denotes the set of final states?

(a) $\{q_2\}$ (b) $\{q_1, q_2\}$
(c) $\{q_0, q_1, q_2\}$ (d) Can't be determined

Q.50 Which of the following strings will not be accepted by the given NFA?

(a) 00 11 22 (b) 11 22
(c) 21 (d) 22

Q.51 Consider the transition diagram of DFA as given below:

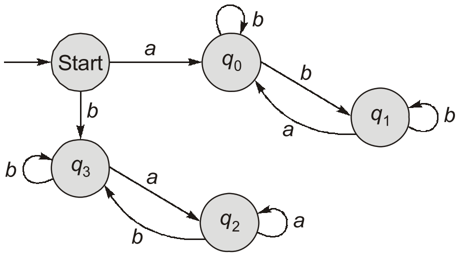


Which is the language of the given DFA?

(a) $L = \{\in\}$
(b) $L = \{\ \}$
(c) $L = \{w \mid w$ has equal no. of 1's and 0's$\}$
(d) None of these

Directions for Question 52, 53 and 54:

Consider the transition diagram of an DFA as given below:



Q.52 Which should be the final state(s) of the DFA if it should accept strings starting with '*a*' and ending with '*b*'?

(a) q_0 (b) q_1

(c) q_0, q_1 (d) q_3

Q.53 Which of the following strings will be accepted if q_0 and q_1 are accepting states?

1. *ababab*
2. *babaaa*
3. *aaaba*

(a) 1, 2 (b) 2, 3

(c) 1, 3 (d) None of these

Q.54 Which of the following represents the set of accepting states if the language to be accepted contains strings having the same starting and ending symbols?

(a) $\{q_0\}$ (b) $\{q_0, q_3\}$

(c) $\{q_3\}$ (d) $\{q_3, q_1\}$

Q.55 Which of the following pairs of RE are equivalent?

(a) $1(01)^*$ and $(10)^*1$

(b) $x(xx)^*$ and $(xx)^*x$

(c) $(a + b)^*$ and $(a^*b^*)^*$

(d) All of the above

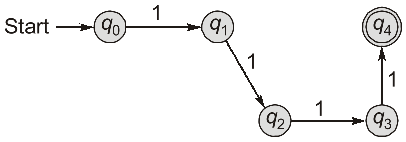
Q.56 Let $\Sigma = \{0,1\}$, then an automaton *A* accepting only those words from Σ having an odd number of 1's requires ___ states including the start state.

(a) 2 (b) 3

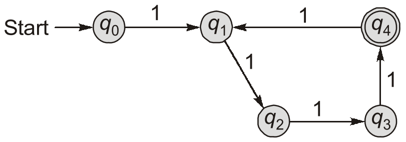
(c) 4 (d) 5

Q.57 Design a FSM to check whether a given unary number is divisible by 3.

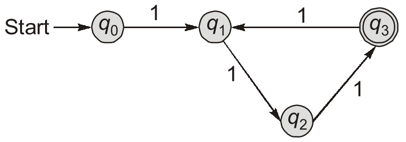
(a)



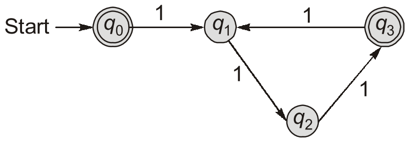
(b)



(c)



(d)



Linked Question 58 and 59:

Q.58 The given transition table is for the FSM that accepts a string if it ends with '*aa*'. Which is the final state?

δ	*a*	*b*
Start	q_0	q_2
q_0	q_1	q_2
q_1	q_1	q_2
q_2	q_0	q_2

(a) q_0 (b) q_1

(c) q_2 (d) Can't be determined

Q.59 A minimum state DFA accepting the same language has how many states?

(a) 1 (b) 2

(c) 3 (d) 4

Directions for Question 60, 61, 62, 63 and 64:

Which is the equivalent Regular Expression (RE) for the following?

Q.60 Strings having exactly one occurrence of *ab* and *ba*

(a) $(a + b)^* (ab + ba) (a + b)^*$

(b) $(ab)^+ (a + b)^* (ba)^+$

(c) Both (*a*) and (*b*)

(d) None of the above

Q.61 Strings having at most one occurrence of *ab* or *ba* but not both

(a) $a^* b^* + b^* a^*$ (b) $b^* ab + a^* ba$

(c) Both (a) and (b) (d) None of these

Q.62 Strings in which every group of three symbols should contain atleast one a.

(a) $[(a + b) (a + b)a]^*$

(b) $[(a + b) (a + b)a]^* [(a + b) (a + b) a]^*$

(c) $[(\in + b + bb) a]^* [\in + b + bb]$

(d) $(abb)^* (bab) b^* (bba)^*$

Q.63 Strings containing at most 2 *a*'s

(a) $b^* ab^*a$

(b) $b^* (a + \in) b^* (a + \in)$

(c) $b^* + b^* ab^* + b^* ab^* ab^*$

(d) None of the above

Q.64 The length of string should be a multiple of 3.

(a) $[(a + b) (a + b) (a + b)]^*$

(b) $(a + b)^* (a + b)^* (a + b)^*$

(c) $[(a + b)^* (a + b) (a + b)]^*$

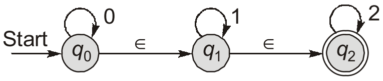
(d) Both (a) and (b)

Q.65 $(a + b)^* \, a \, (a + b)^*$ represents:
(a) Strings of odd length having a at the middle.
(b) All strings containing a and b.
(c) All strings of odd length.
(d) All strings containing atleast one 'a'.

Q.66 Let L be the set of all strings over {0 , 1} of length 6 or less. Write a simple RE corresponding to L.
(a) $(0 + 1)^*$ (b) $(0 + 1)^6$
(c) $(0 + 1 + \in)^*$ (d) $(0 + 1 + \in)^6$

Q.67 Let L be the language, $L = \{x \in \{0 , 1\} \mid x$ ends with 1 and does not contain the substring 0 0}. Give the proper R.E. for the above language.
(a) $(1 + 01)^* (1 + 01)$ (b) $(1 + 01)^*$
(c) Both (a) and (b) (d) None of these

Q.68 Give the R.E. described by the following NFA.



(a) $(0\,1\,2)^*$ (b) $(0 + 1 + 2)^*$
(c) $0^*1^*2^*$ (d) None of these

Q.69 Write the grammar for the regular expression a^*b^*?
(a) $S \to AB,\ A \to aA \mid bB \mid \in,\ B \to bB \mid aA \mid \in$
(b) $S \to AB,\ A \to aA \mid \in,\ B \to bB \mid \in$
(c) $S \to ab \mid \in,\ A \to aA \mid \in,\ B \to bB \mid \in$
(d) None of these

Q.70 $[(a + b)\,(a + b)]^*$ represents:
(a) All strings
(b) All strings of even length.
(c) All strings in which group of 2 symbols has both the symbols same but grouping must be done left to right starting from first symbol.
(d) All strings in which group of 2 symbols has both the symbol same but grouping must be done right to left starting from last symbol.

2. Push Down Automata: CFL & DCFL

Q.71 Which of the following is not true?
(a) The set of languages accepted by deterministic and non deterministic PDAs are not equal.
(b) $L = \{wcw^R \mid w \in (0 + 1)^*$ and $c \notin \{0,1\}\}$ can be accepted by a deterministic PDA.
(c) $L = \{ww^R \mid w \in (0 + 1)^*\}$ can be accepted by a deterministic PDA.
(d) $L = \{0^n1^n \mid n \geq 0\}$ can be accepted by a deterministic PDA.

Q.72 Which of the following statements are true?
(i) The complement of a language is always regular.
(ii) The intersection of regular languages is regular.
(iii) The complement of a regular language is regular.
(a) (i) and (ii) only (b) (ii) and (iii) only
(c) (i) and (iii) only (d) All of these

Q.73 Which of the following is not true?
(a) CFLs are closed under union and concatenation.
(b) Regular languages are closed under union and intersection.
(c) CFLs are not closed under intersection and complementation.
(d) If L is a CFL and R is a regular set then $L \cap R$ is not a CFL.

Q.74 Context-free languages and regular languages are both closed under the operations(s) of
(i) Union
(ii) Intersection
(iii) Concatenation
(a) (i) and (ii) only (b) (ii) and (iii) only
(c) (i) and (iii) only (d) All of these

Q.75 Which of the following languages are context free
$L_1 = \{a^m b^m c^n \mid m \geq 1 \text{ and } n \geq 1\}$
$L_2 = \{a^m b^m c^n \mid n \geq m\}$
$L_3 = \{a^m b^m c^m \mid m \geq 1\}$
(a) Only L_1 (b) L_2 and L_3
(c) Only L_2 (d) Only L_3

Q.76 If L_1 is regular and L_2 is CFL over Σ^* which of the following statement is incorrect?
(a) $L_1 \cup L_2$ is CFL (b) $L_1 \cap L_2$ is regular
(c) L_1^* is regular (d) None of these

Q.77 Context free languages are closed under
(a) Union (b) Intersection
(b) Complementation (d) Set difference

Q.78 Which of the following languages is/are context free?
1. $\{a^n b^n c^m d^m \mid n \geq 1, m \geq 1\}$
2. $\{a^n b^m c^m d^n \mid n \geq 1, m \geq 1\}$
3. $\{a^n b^m c^n d^m \mid n \geq 1, m \geq 1\}$
4. $\{a^m b^n c^m d^n \mid n \geq 1, m \geq 1\}$
(a) 1 and 2 (b) 3 and 4
(c) 2 and 4 (d) 1, 2, 3 and 4

Q.79 Which of the following is undecidable?
(a) Equivalence of regular languages
(b) Equivalence of context free languages
(c) Finiteness check on context free grammar
(d) Emptiness of regular languages

Q.80 Context-free language can be recognized by
(a) Finite state automaton
(b) Linear bounded automata
(c) Push down automata
(d) Both (b) and (c) above

Q.81 Consider the language:
$L_1 = \{a^n\, b^m\, c^n\, d^m \mid n \geq 1,\ m \geq 1\}$ and
$L_2 = \{a^n\, b^m\, c^m\, d^n \mid n \geq 1,\ m \geq 1\}$
(a) Both L_1 and L_2 are context free
(b) L_1 is not context free but L_2 is context free
(c) Both are not context free
(d) L_1 is context free but L_2 is not context free

Q.82 A language L is accepted by a finite automaton if and only if it is
(a) Context-free
(b) Context-sensitive
(c) Recursive
(d) Expressible by a right-linear grammar

Q.83 The set $\{a^n\, b^n \mid n = 1, 2, 3, \ldots\}$ can be generated by the CFG
(a) $S \to ab \mid aSb \mid \in$ (b) $S \to aaSbb \mid ab$
(c) $S \to ab \mid aSb$ (d) None of these

Q.84 The grammar $S \to aaSbb \mid ab$ can generate the set
(a) $\{a^n\, b^n \mid n = 1, 2, 3, \ldots\}$
(b) $\{a^{2n+1}\, b^{2n+1} \mid n = 0, 1, 2, \ldots\}$
(c) $\{a^{2n+1}\, b^{2n+1} \mid n = 1, 2, 3, \ldots\}$
(d) None of the above

Q.85 CFLs are not closed under
(a) Union (b) Concatenation
(c) Closure (d) Intersection

Q.86 CFLs are not closed under
(a) Union (b) Kleen star
(c) Complementation (d) Product

Q.87 The set $A = \{a^n\, b^n\, a^n \mid n = 1, 2, 3, \ldots\}$ is an example of a language that is
(a) Regular (b) Not context-free
(c) Context-free (d) None of these

Q.88 The logic of pumping lemma is a good example of
(a) The divide - and - conquer technique
(b) The pigeon - hole principle
(c) Recursion
(d) Iteration

Q.89 Which of the following languages over $\{a, b, c\}$ is accepted by deterministic push down automata?
(a) $\{w \mid w$ is palindrome over $\{a, b, c\}\}$
(b) $\{ww^R \mid w \in \{a, b, c\}^*\}$
(c) $\{a^n\, b^n\, c^n \mid n \geq 0\}$
(d) $\{wcw^R \mid w \in \{a, b\}^*\}$

Q.90 Regarding the power of recognizing the languages, which of the following statements is false?
(a) The NDFA are equivalent to DFA
(b) NDPDA are equivalent to DPDA
(c) NDTMs are equivalent to DTMs
(d) Multiple tape TMs are equivalent to single tape TMs

Q.91 A grammar that is both left and right recursive for a non-terminal, is
(a) Ambiguous
(b) Unambiguous
(c) Information is not sufficient to decide
(d) None of these

3. Turing Machine: RE, REC and Undecidability

Q.92 Which of the following functions are computable with Turning Machine?
(a) $n*(n-1)*(n-2)* \ldots *2*1*$
(b) $\lceil \log_2 n \rceil$
(c) 2^{2^n}
(d) All of the above

Q.93 A countable union of countable sets is not
(a) Countable
(b) Uncountable
(c) Countably infinite
(d) Denumerable

Q.94 Σ^* over $\{0, 1\}$ and 2^{Σ^*} are respectively,
(a) Uncountably infinite and uncountably infinite
(b) Countably infinite and uncountably infinite
(c) Countably infinite and countably infinite
(d) None of the above

Q.95 Which of the following is undecidable?
(a) Equivalence of regular languages
(b) Equivalence of context free languages
(c) Finiteness check on context free language
(d) Emptiness of regular languages

Q.96 The statement "A TM can't solve halting problem" is
(a) True
(b) False
(c) Still a open question
(d) None of the above

Q.97 A PDA behaves like a TM when the number of auxiliary memory it has, is
(a) 0 (b) 1 or more
(c) 2 or more (d) None of these

Q.98 Halting problem/language is
(a) RE as well as recursive
(b) Recursive and NP
(c) RE but not recursive
(d) Neither recursive nor RE

Q.99 The statement $P \subseteq$ NP is
(a) True
(b) False
(c) Still open for argument
(d) None of the above

Q.100 If e_1 and e_2 are the RE denoting the languages L_1 and L_2 respectively, then which of the following is wrong?
(a) $(e_1)|(e_2)$ is a regular expression denoting $L_1 \cup L_2$
(b) $(e_1)(e_2)$ is a RE denoting $L_1 . L_2$
(c) ϕ is not a RE
(d) $\{e_1\}^*$ is a RE denoting $L_1{}^*$

Q.101 Which of the following is in P?
PATH, HAMPATH, SAT, 3SAT
(a) SAT (b) 3SAT
(c) PATH (d) HAMPATH

Q.102 If $\forall w \in \Sigma^*$, it can be determined in finite time, whether or not $w \in L$, then L is
(a) Decidable (b) Undecidable
(c) Non-deterministic (d) Intractable

Q.103 Consider the following decision problems:
P_1: Does a given finite state machine accept a given string.
P_2: Does a given context free grammar generate an infinite number of strings.
Which of the following statement is true?
(a) P_1 and P_2 are decidable
(b) Neither P_1 nor P_2 are decidable
(c) Only P_1 is decidable
(d) Only P_2 is decidable

Q.104 Consider the following Problem X:
"Given a Turing Machine M over the input alphabet Σ, any state q of M and a word Σ^*, does the computation of M on w visit the state q".
Which of the following statements about X is correct?
(a) X is decidable
(b) X is undecidable but partially decidable
(c) X is undecidable and not even partially decidable
(d) X is not a decision problem

Q.105 Nobody knows yet if $P = NP$. Consider the language L defined as follows:

$$L = \begin{cases} (0+1)^* \text{ if } P = NP \\ \phi \text{ otherwise} \end{cases}$$

Which of the following statements is true?
(a) L is recursive
(b) L is recursively enumerable but not recursive
(c) L is not recursively enumerable
(d) Whether L is recursive or not will be known after we find out if $P = NP$

Q.106 Consider three decision problems P_1, P_2 and P_3. It is known that P_1 is decidable and P_2 is undecidable. Which one of the following is TRUE?
(a) P_3 is decidable if P_1 is reducible to P_3
(b) P_3 is undecidable if P_3 is reducible to P_2
(c) P_3 is undecidable if P_2 is reducible to P_3
(d) P_3 is decidable if P_3 is reducible to P_2's complement

Q.107 Let L_1 be a recursive language, and let L_2 be a recursively enumerable but not a recursive language. Which one of the following is TRUE?
(a) $\overline{L_1}$ is recursive and $\overline{L_2}$ is recursively enumerable.
(b) $\overline{L_1}$ is recursive and $\overline{L_2}$ is not recursively enumerable.
(c) $\overline{L_1}$ and $\overline{L_2}$ are recursively enumerable.
(d) $\overline{L_1}$ is recursively enumerable and $\overline{L_2}$ is recursive.

Q.108 Which of the following problems is undecidable?

(a) Membership problem for CFGs

(b) Ambiguity problem for CFGs

(c) Finiteness problem for FSAs

(d) Equivalence problem for FSAs

Q.109 Which of the following is true for the language $\{a^p \mid p \text{ is a prime}\}$?

(a) It is not accepted by a Turning Machine

(b) It is regular but not context-free

(c) It is context-free but not regular

(d) It is neither regular nor context-free, but accepted by a Turing machine

Q.110 Which of the following are decidable?

1. Whether the intersection of two regular languages is infinite.
2. Whether a given context-free language is regular.
3. Whether two push-down automata accept the same language.
4. Whether a given grammar is context-free.

(a) 1 and 2 (b) 1 and 4

(c) 2 and 3 (d) 2 and 4

Answers Theory of Computation

1.	(c)	2.	(d)	3.	(d)	4.	(d)	5.	(d)	6.	(d)	7.	(b)	8.	(a)	9.	(b)	10.	(a)
11.	(a)	12.	(c)	13.	(b)	14.	(d)	15.	(b)	16.	(b)	17.	(b)	18.	(b)	19.	(c)	20.	(b)
21.	(a)	22.	(c)	23.	(a)	24.	(a)	25.	(d)	26.	(b)	27.	(d)	28.	(c)	29.	(b)	30.	(c)
31.	(d)	32.	(a)	33.	(d)	34.	(d)	35.	(a)	36.	(b)	37.	(c)	38.	(b)	39.	(a)	40.	(d)
41.	(c)	42.	(c)	43.	(b)	44.	(d)	45.	(d)	46.	(c)	47.	(b)	48.	(a)	49.	(c)	50.	(c)
51.	(d)	52.	(b)	53.	(c)	54.	(b)	55.	(d)	56.	(a)	57.	(d)	58.	(b)	59.	(c)	60.	(d)
61.	(a)	62.	(c)	63.	(c)	64.	(a)	65.	(d)	66.	(d)	67.	(a)	68.	(c)	69.	(b)	70.	(b)
71.	(c)	72.	(b)	73.	(d)	74.	(c)	75.	(a)	76.	(b)	77.	(a)	78.	(a)	79.	(b)	80.	(d)
81.	(b)	82.	(d)	83.	(c)	84.	(c)	85.	(d)	86.	(c)	87.	(b)	88.	(b)	89.	(d)	90.	(b)
91.	(c)	92.	(d)	93.	(b)	94.	(b)	95.	(b)	96.	(a)	97.	(c)	98.	(c)	99.	(a)	100.	(c)
101.	(a,c)	102.	(a)	103.	(a)	104.	(b)	105.	(a)	106.	(c)	107.	(b)	108.	(b)	109.	(d)	110.	(b)

Explanation Theory of Computation

1. (c)

For every NFA there exist an equivalent DFA and vice-versa. Power of both NFA and DFA for recognization of language is same.

2. (d)

$ac \in L(r_1)$, since we can take b^* as $\in$ and c^* as c.
$ac \in L(r_3)$, since $(a + b + c)^*$ includes all combinations of a, b and c.

$ac \notin L(r_2)$ since whenever $(a^* b + c)^*$ is taken to include a, "a is always followed by b".
$a^*b =$ b, ab, aab, ... & so on.

3. (d)

$L(r_1)$ is the set of all string starting with 'a'.
$L(r_2)$ is the set of all starting with 'b'.
Since any word belonging to Σ^* either starts with "a" or starts with "b" or is "$\in$", therefore
$L(r_1) \cup L(r_2) \cup \{\in\} = \Sigma^*$

4. (d)

Note that

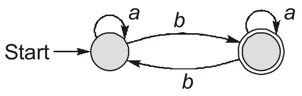
$(b^*a^*)^* = (a^*b^*)^* = (a^* + b^*)^* = (a + b)^*$ so for (a) $L((b^*a^*)^*\ (d^*c^*)^*)$ gives all the combinations of a and b followed by all the combinations of c and d.

for (b) $L((d^*c^*b^*a^*)^*) = (a + b + c + d)^*$ will generate all strings with combination of a, b, c, b and d.

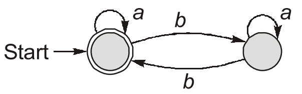
for (c) $L((a^*b^*a^*c^*d^*)^*) = (a + b + c + d + a^*)^*$
$= (a + b + c + d)^*$

5. (d)

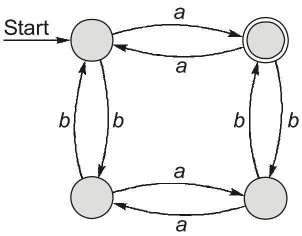
(i)



(ii)



(iii)



As we can construct the DFA for the given languages, hence all of the given languages are regular.

6. (d)

All languages are accepting strings over the alphabet {a, b} that contains exactly two a's but they are not accepting all strings like the first choice just accepts 'aa'. The second choice can't accept 'baa'. The third choice can't accept '$baab$'. The correct regular expression is $b^*\ a\ b^*\ a\ b^*$.

7. (b)

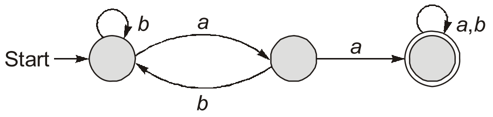
Choice 'a' is incorrect since it does not include the string "a", "b" and "∊" (all of which do not end with ab).

None of choices 'c' or 'd' accept the string 'a', So they can't represent specified language.

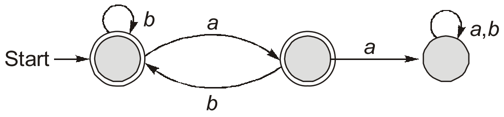
8. (a)

The regular expression corresponding to the language of strings of even lengths over the alphabet of {a, b} is $((a + b)^2)^*$ which is equivalent to $(aa + bb + ba + ab)^*$. In choice (b) the string 'ab' is not present. In choice (c) the string 'aa' is not present. In choice (d) odd length strings are also acceptable.

9. (b)



First draw FSM for accepting all strings containing consecutive a's, as shown above. Now change the final states to non final states and non final states to final states to get the required DFA shown below:



10. (a)

For DFA accepting all the strings with number of a's divisible by 4, four states are required similarly for DFA accepting all the strings with number of b's divisible by 5, five states are required and for their combination, states will be multiplied.

So 5 × 4 = 20 states will be required.

11. (a)

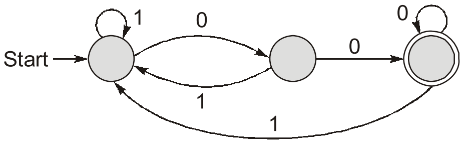
(ii) and (iii) are false since, $x^+\ y^+$ does not necessarily generate equal number of x's and y's and hence are not satisfying the conditions of the given language.

12. (c)

Since $L = X^* = \{0, 1\}^* = (0 + 1)^*$ and $R = \{0^n 1^n \mid n > 0\}$, $L \cup R$ produces $(0 + 1)^*$ which is regular language and R is not regular as there is no regular expression for that R is actually DCFL.

13. (b)

The DFA will be



For these kind of problems the required number of states are always equival to "the length of the string that it is ending with" + 1.

14. (d)

The correct answer is (d), it is a standard identity $(r^* s^*)^* = (r^* + s^*) = (r + s)^*$

15. (b)

Regular languages are closed under union, intersection, complementation, Kleen closure, concatenation.

According to closure properties, all of *a*, *b*, *c*, and *d* are regular. Notice that, $\Sigma^* - L_1 = L_1^C$ so, set $L_1 \cap L_2$ is also regular.

16. (b)

Since L_1 and L_2 both are accepted by deterministic PDA since only one comparison is there. (In both cases, no guessing is required for the push and pop operations while operating the stack) since these languages are accepted by DPDA, it will also be accepted by NPDA.

17. (b)

1. $r_1 = \in$ is a regular expression representing a set $\{\in\}$. It is a regular expression.
2. $r_2 = 0^* 1^*$ is a regular expression representing the language $\{0^n 1^m \mid n, m \geq 0\}$.

18. (b)

The transition function of DFA from one state to another on a given input symbol '*w*' is a function $Q \times \Sigma^*$ to Q. (Definition of DFA). It means at any given state $q \leftarrow Q$, if some input, is given, it will go to some state $q \leftarrow Q$.

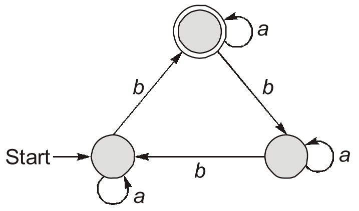
Note: For NFA it is 2^Q. i.e. No. of final states or output state in case of NFA is a subset of 2^Q.

19. (c)

Only option (c) satisfies the given machine, hence $L(M)$ = {words containing *aa* or *bb* as a subword}.

20. (b)

The automata *D* can be constructed as



Hence number of states in the minimised automata is 3.

Note: For any regular language containing $(nk + j)$ times any alphabet, where $(j < n)$

Number of states in the DFA for such language = *n* states.

21. (a)

Here (b), $(1^* 0^*)^* = (0 + 1)^*$ Which contains all strings of 0 and 1. (d) also contains the 00 substrings also (c) $10^* + 1^* = \{(1, 10, 100, ...) + (\in, 1, 11, ...)\}$ also contains '00'.

Only option (a) is correct where no consecutive 0's are possible.

22. (c)

$$\begin{aligned} L_1 &= \{a^n b^n c^n, n \geq 0\} \\ &= \{\in, abc, a^2b^2c^2, ...\} \\ L_2 &= \{a^{2n} b^{2n} c^{2n}, n \geq 0\} \\ &= \{\in, a^2b^2c^2, a^4b^4c^4, ...\} \\ L_3 &= \{a^{2n} b^{2n} c^n, n \geq 0\} \\ &= \{\in, a^2b^2c, a^4b^4c^2, ...\} \end{aligned}$$

as we can easily see that

(i) L_1 contains all the words generated by L_2 and also it contains some extra strings also.
$\therefore\ L_1 \supseteq L_2$. (or $L_2 \subseteq L_1$)

(ii) Since only $\in$ is common in L_2 and L_3
Hence $L_2 \not\subset L_3$.

23. (a)

$$\begin{aligned} L_1 &= \{10, 1\}, \\ L_2 &= \{011, 11\} \end{aligned}$$

By concatenation of L_1 and L_2 we get

$$L_1 \cdot L_2 = \{10011, 10111, 1011, 1111\}$$

Hence, 4 distinct elements are there.

24. (a)

1. 0^*1^* does not ensure at least one 0 at the beginning and one 1 at the end.
2. $00^*(0 + 1)^*\ 1$ ensures the specified condition.
3. $0(0 + 1)^*\ 1$ ensures the specified condition.

25. (d)

The regular expression $(1 + 01)^*$ generates strings which have no consecutive 0's. 10, 11, 01 are all possible substrings.

26. (b)

$S \rightarrow aS \mid bS$ will generate all strings with *a* and *b*
$S \rightarrow a \mid b$ ensure that atleast one character will be there in every word.

Therefore, regular expression of aforesaid grammar is

$$\begin{aligned}(a+b)^+ &= (a+b)(a+b)^* \\ &= (a+b)^*(a+b)\end{aligned}$$

27. (d)

Among the tuples $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$
Q defines the set of states.
Σ includes the set of input alphabets.
Δ defines output alphabets.
δ is the transition function from $\Sigma \times Q \rightarrow Q$
λ is the output function mapping $Q \rightarrow \Delta$
q_0 is the starting state.

28. (c)

δ is the transition function from $\Sigma \times Q \rightarrow Q$.

29. (b)

Binary number that are multiples of two is

$$L = \{0, 10, 100, 1000, 110, 100, ...\}$$

i.e. strings ending with 0
∴ (b) (0/1)*.0 is the correct answer.

30. (c)

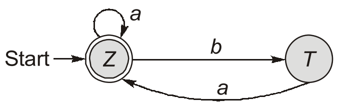
The regular expression corresponding to the language containing "*aa*" as a substring is

$$(a+b)^* \, aa \, (a+b)^*$$

In (a), (b) and (d) by putting $(a+b)^*$ as $\in$ we are getting *ab*, *bb* and *ba* respectively which is the minimal strings and does not contain "*aa*". Hence only (c) necessarily contains the substring "*aa*".

31. (d)

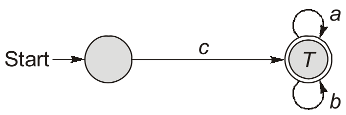
Since the grammar is right linear grammar and hence regular we can design a DFA for it



The language corresponding to the above DFA is
$L = \{\lambda, a, aa, aaa, ..., ba, baba, baba...\}$

$$\therefore \quad L = (a + ba)^*$$

32. (a)



The language generated by the above DFA is starting with *c* followed by any no. of *a*'s and *b*'s.
Hence, $L = c(a+b)^*$

33. (d)

$(a|b)\,(a|b)$ is simply the concatenation of each element of $(a|b)$ with $(a|b)$ itself. Hence the language contains the following elements.

$$L = \{aa, bb, ab, ba\}$$

34. (d)

Since, $(a^* + b^*)^* = (b^* + a^*)^* = (a+b)^* = (b+a)^*$ and it denotes all the combination of *a*'s and *b*'s of any order with any length. Since $(a+b)$ generates single length string hence it is not equivalent to $(a^* + b^*)^*$.

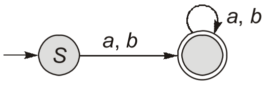
35. (a)

$$R = (ab|abb)^* \, bbab$$

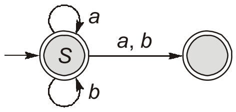
Any string generated by above, must end with "*bbab*". Since the string '*abababb*' is not ending with "*bbab*". So choice (a) is not in $L(R)$. *b*, *c* and *d* can be generated by $L(R)$.

36. (b)

The NFA's construction for the grammars:
$S \rightarrow aS|bS|a|b$ and $S \rightarrow aS|bS|a|b|\in$ are respectively



and



∴ The required languages are:

$$\begin{aligned}L_1 &= (a+b)(a+b)^* \\ L_2 &= (a+b)^*\end{aligned}$$

37. (c)

Options:

(a) $(ab)^* \, a$ is equivalent to $a(b\,a)^*$
it follows form the property

$$P(QP)^* = (PQ)^* \, P$$

(b) $(a+b)^* = (a^* + b^*)^* = (a^* + b)^* = (a + b^*)^*$

(c) $b^* \, ab^* = \begin{cases} a, ba, bba, bbba........ \\ ab, abb, abbb......... \\ bab, bbabb .... \end{cases}$

}

$a^* \, ba^* = \{b, ab, aab, ... \, ba, baa, ...\}$

Clearly it can be seen that the minimal string in $b^* \, ab^*$ is "*a*" and that of $a^* \, ba^*$ is "*b*" which are not same.

38. (b)

Since the language recognization powers of NFA and DFA are always same therefore DFA can always simulates the NFA.

39. (a)

It follows from the basic definition of the FSM. Since finite state machines have finite memory and finite input tape hence it cannot remember arbitrary large amount of information.

Hence option (a) is correct.

Note: FSM always recognizes the regular grammars since regular grammars generates regular languages.

Hence (c) is not correct.

Also since the grammars which are not regular can generate regular languages and can generates non regular languages also.

Hence (b) is true and is not a limitation of FSM.

40. (d)

Since every language can be represented through regular expression and its equivalent machine. Also for the every deterministic machine there exist its equivalent non-deterministic machine.

42. (c)

The FSM shown accept only ∈ because it is the starting as well as final state and no other input is there.

43. (b)

The FSM shown has no final states to accept strings. Hence, it accepts no strings.

44. (d)

Here, $L_1 = (a + b)^*a$ is nothing but set of all strings of a and b ending with a.

$L_2 = b(a + b)^*$ is set of all strings of a and b starting with b.

The intersection of L_1 and L_2 is set of all strings starting with b and ending with a.

$\therefore \quad L_1 \cap L_2 = b(a + b)^*a$

45. (d)

The string 1101 does not belong to (10)* (01)* (00 + 11)* here (10)* (01)* ensure in the starting that there is no consecutive 1's and once they have taken care (00 + 11)* does not allow for single 0's or 1's.

46. (c)

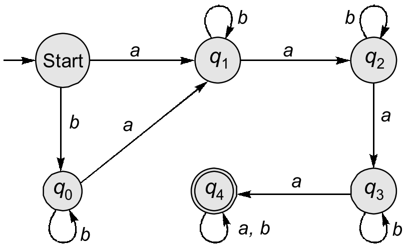
Here $(00)^* (\in + 0)$ is nothing but $(00)^* \in + (00)^*0$ which is union of even and odd strings of 0. (i.e. all combinations). Contracting the same, we get $(00)^* (\in + 0) = 0^*$ which is the 3rd regular expression.

However 2nd and 4th expression represents only the even strings of 0's and only the odd strings of 0's respectively, which are not equivalent to 0*.

47. (b)

This is a standard DFA for all strings ending with "a". Hence the language is $(a + b)^* a$.

48. (a)



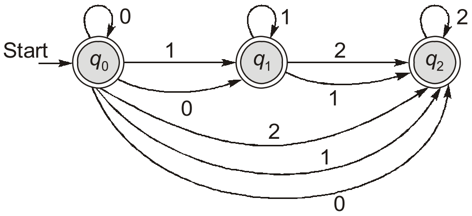
Drawing the FA we have we can clearly see that only

(i) *aaaaa* and

(ii) *aabbaabbbbb* are accepted.

49. (c)

The given NFA when converted to NFA without ∈-moves



Note: The previous state from which there is an ∈-Move, if it goes to the final state then it also becomes the final state as well.

50. (c)

From the above NFA we can clearly see that it accepts strings only if "0 followed by 1 followed by 2" i.e. strings like 2 followed by 0 or 1 and 1 followed by 0 are not accepted. Clearly, (c) is the correct option.

51. (d)

The given FA accepts all combinations of even number of both 0's and 1's.

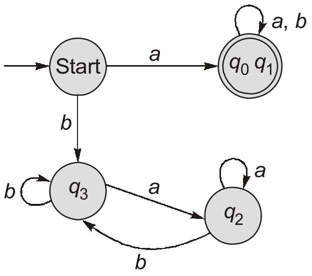
52. (b)

The smallest string starting with '*a*' and ending with '*b*' is '*ab*'.

On giving '*ab*' as input, we can find out that final state is nothing but q_1.

53. (c)

If q_0 and q_1 both are made accepting states or final states, DFA will minimized to



Clearly, the DFA will accept all strings starting with '*a*'. Hence, with accept '*ababab*' and '*aaaba*'.

54. (b)

If the language to be accepted contains strings having the same starting and ending symbol i.e. starting with '*a*' and ending with '*a*' or starting '*b*' ending '*b*'.

Such acceptable states are $\{q_0, q_3\}$. We can check this simply by giving acceptable strings such as *aa*, *bb*, *aba*, *baab* .... as input to the DFA.

55. (d)

All of the pairs of RE are equivalent.

(a) $1(01)^* \cong (10)^*1$

(b) $x(xx)^* \cong (xx)^*x$

are satisfying the property

$P(QP)^* = (PQ)^*P$

(c) $(a+b)^* = (a^*+b^*)$

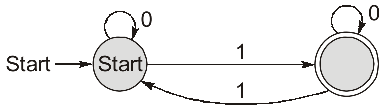
$= (a+b^*)^*$

$= (a^*+b^*)^*$

$= (a^*b^*)^*$

56. (a)

The automaton given below, accepts the words with odd of 1's.



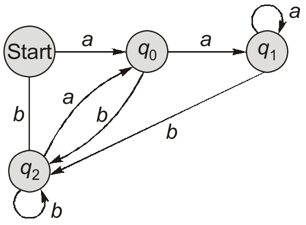
Clearly, the automaton requires only 2 states.

57. (d)

The unary number has a single symbol and the length of the string is its value. So we have to accept strings where, no of 1's is divisible by 3. On first symbols of number (i.e.), we will move to remainder 1 state, (q_1) when next input is received we move to remainder 2 state (q_2) and on next we move to 0 remainder state (i.e. q_3) as shown in the choice (d).

58. (b)

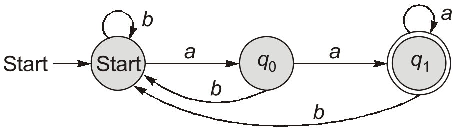
Drawing the FSM for the given transition diagram:



We can clearly see that if 'q_1' is the final state then only the given FSM can generate words which are all ending with "*aa*".

59. (c)

The minimum state DFA accepting the above language is



60. (d)

The correct regular expression is

$$a^*bb^*aa^* + b^*aa^*bb^*$$

61. (a)

b^*ab will generate *bab*, which contains both "*ab*" and "*ba*".

62. (c)

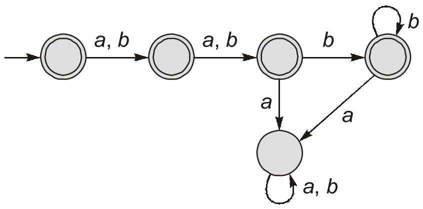
Option **(a)** cannot generate the valid string "*aaaba*." (a) and (b) will only generate those strings, in which string-length is multiple of 3.

Option (b) cannot generate the valid string "*bbababba*."

Option (d) generate the invalid string "*abbbabbba*".

63. (c)

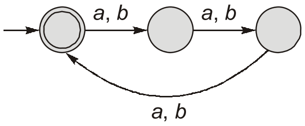
DFA for given language is:



At most 2 *a*'s can be broken into union of exactly 0 "*a*", exactly 1 "*a*" (b^*ab^*) and exactly 2 *a*'s ($b^*ab^*ab^*$), hence choice (c) is correct. (b) is wrong because it will not be able to generate strings with 2 a's and ending with *b* like *babab*, *abab* etc.

64. (a)

DFA for given language is:



The length of string being multiple of 3 is

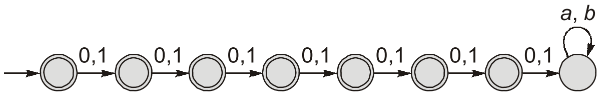
$$[(a+b)(a+b)(a+b)]^*$$

65. (d)

$(a+b)^*$ means all combination of '*a*' and '*b*'. Hence '*a*' in $(a+b)^* a(a+b)^*$ signifies that at least one '*a*' must be there.

66. (d)

L = set of all strings over {0, 1} of length of 6 or less



$$L = (0+1+\in)^6$$

67. (a)

As we know that $(1+01)^*(1+01)$ is R.E. with all strings of '0' and '1' ending with '1' and not contain the substring '00'.

Choice (b) $(1+01)^*$ is incorrect, since $(1+01)^*$ will generate "$\in$" which is not ending with 1.

68. (c)

Clearly 0's followed by 1's followed by 2's is being accepted so choice (c) $0^*\ 1^*\ 2^*$ is correct.

69. (b)

The language $L = \{w \mid w \in$ any number of a's followed by any number of *b*'s}

has regular expression $a^*\ b^*$. The corresponding grammar is

$$S \to AB,\ A \to aA \mid \in, B \to bB \mid \in$$

Here $A \to aA \mid \in \quad B \to bB \mid \in$ denotes any number of *a*'s and any number of *b*'s respectively.

$S \to AB$ gives the number of *a*'s followed by number of *b*'s.

70. (b)

$((a+b)(a+b))$ generates all the 2 length strings *aa*, *ba*, *ab* and *bb*.

So $[(a+b)(a+b)]^*$ will generates all the even length strings, including $\in$.

71. (c)

$L = \{ww^R \mid w \in (0+1)^*\}$ cannot be accepted by the deterministic PDA since guessing of the alphabet after which reversal is done is required. One must have to guess the end of *w*, so that start of w^R is found. The word cannot be determined deterministically as $w \in (0+1)^*$ which includes multiple combinations.

72. (b)

According to the closure properties regular language is closed under union, intersection, Kleen closure, concatenation, complementation. But the complementation of any language need not to be always regular.

73. (d)

According to closure properties regular languages are closed under the all finite basic operations like: $\cup$, $\cap$, ., *, L^C.

But CFL is closed only under $\cup$, ., *

Also Regular intersection of CFL i.e. CFL intersected with regular language.

If $L_1 \in$ CFL, $L_2 \in$ Regular, then $L_1 \cap L_2$ is CFL.

74. (c)

Regular and context free languages are both closed under union and concatenation while only regular is closed under intersection.

75. (a)

In L_1 only one stack is required since only one comparison is there. While for L_2 and L_3 two stacks are required since there are 2 comparisons.

So, only L_1 is context free language as it is accepted by PDA.

76. (b)

Since L_1 and L_2 are regular and context free languages respectively CFL are closed under regular $\cup$, $\cap$, therefore

(a) $L_1 \cup L_2$ is CFL and therefore regular too.

(b) $L_1 \cap L_2$ is not regular

(c) $L_1{}^*$ is regular

77. (a)

Out of the four operations given, as per the closure properties CFL are closed under only union.

78. (a)

Here for the languages (i) and (ii) we can design a PDA Hence they are CFL.

In (iii) and (iv) we cannot conclude whether number of (a) = number of (c) and number of (b) = number of (d) using only one stack memory so they aren't CFL.

80. (d)

Since the power of recognization of NFA is less then the PDA. Also CFL can be recognized by the one which has equal or higher power than the PDA. Both PDA and LBA can recognize the CFL.

81. (b)

Since, L_1 cannot be recognized by PDA but L_2 can be recognized by a PDA. L_1 is not context free but L_2 is context free.

For, $L_2 = \{a^n\, b^m\, c^m\, d^n \mid n \geq 1,\ m \geq 1\}$

a's are pushed into stack, then b's are pushed then b's are popped from the stack checking with each arrival of corresponding c. After all b's are popped and if number of b's popped = number of c's arrived then, as a will be at top of stack, when the first d arrives.

So, number of b's = number. of c's [$m = m$]. Similarly when d's arrive, we can check that Number of a's = Number of d's. [$n = n$], by popping an a for every d and checking that at end of input stack is empty.

This can't be done in the case of L_1, since after pushing a's and then b's, when c's appear in input, they have to be compared with a's which are unfortunately at the bottom of the stack and cannot be popped to do a comparison.

82. (d)

The language recognization powers of finite automata is very restricted. It can only accept regular language. If a grammar is only left linear or only right linear but not both then it is a regular grammar and the language expressible by it is always regular.

83. (c)

The language set $\{a^n b^n \mid n = 1, 2, 3 \,.....\}$ can be generated by (c) $\rightarrow ab \mid aSb$

(a) is false since it is generating $\in$ also.

(b) is false since $aabb$ is not generated.

84. (c)

$S \rightarrow aa\ S\ bb \mid ab\ \ S \rightarrow aaSbb$ generates the sentential from $a^{2n}Sb^{2n}$.

Now by substituting

$S \rightarrow ab$, we get $\{a^{2n+1}\, b^{2n+1} \mid n \geq 1\}$.

$S \rightarrow ab$ can directly generate "ab".

So, $L = \{a^{2n+1}\, b^{2n+1},\ n \geq 1\}$

85. (d)

It follows from the closure properties of CFL'S. CFL'S are closed under union, concatenation and Kleen closure, but not under intersection and complement.

86. (c)

CFL's are not closed under complementation.

87. (b)

For set $A = \{a^n b^n a^n \mid n = 1, 2, 3, ...\}$ the number of comparisons are 2 first one for checking number of a = number of b and, second one for again checking number of b = number of a's following the b's. As we know, a PDA can do only one of these comparisons.

Hence the given language is not context free.

88. (b)

The logic of pumping lemma is a good example of pigeon-hole principle.