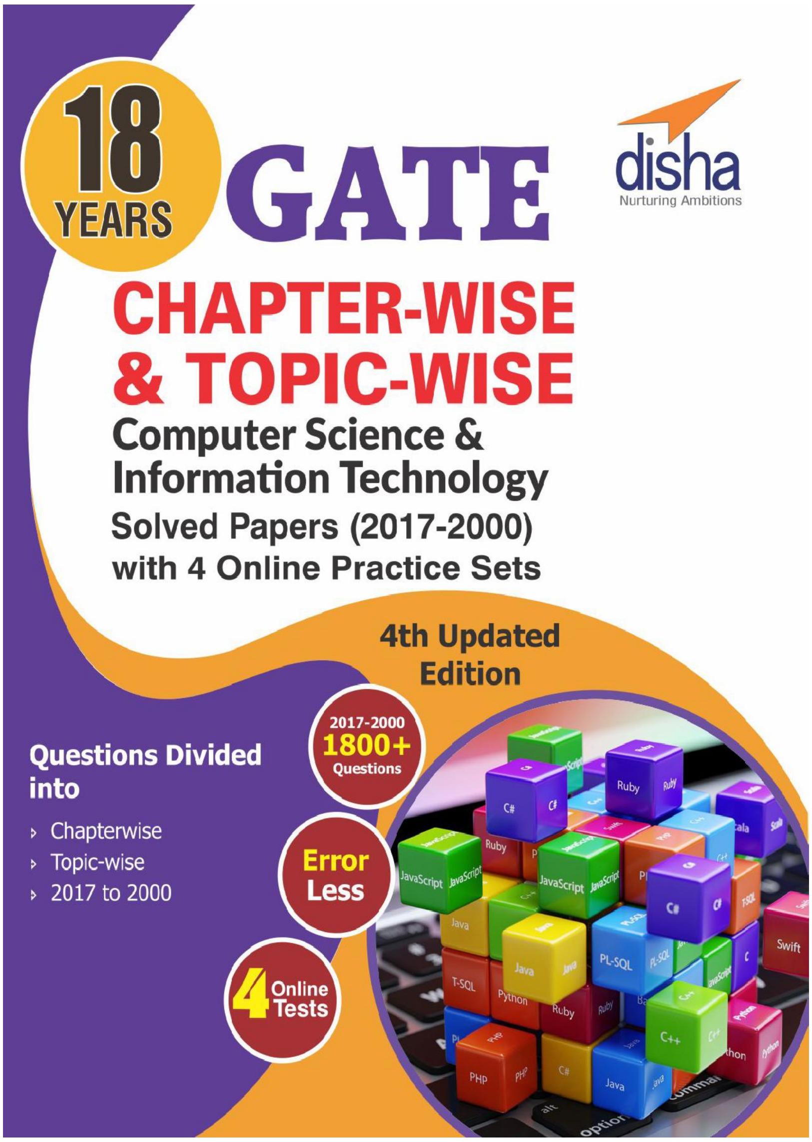
THEORY OF COMPUTATION

This Chapter "Theory of Computation" is taken from our Book:



ISBN : 9789386629043

THEORY OF COMPUTATION 5

Regular Expression and Finite Automata

1. Consider the language L given by the regular expression $(a + b)*b(a + b)$ over the alphabet $\{a.b\}$. The smallest number of states needed in a deterministic finite-state automata (DFA) accepting L is ________.
[2017, Set 1, 1 Mark]

2. Let δ denote the transition function and δ denote the extended transition function of the $\in$-NFA whose transition table is given below: **[2017, Set 2, 2 Marks]**

δ	$\in$	a	b
$\rightarrow q_0$	$\{q_2\}$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_0\}$	ϕ	ϕ
q_3	ϕ	ϕ	$\{q_2\}$

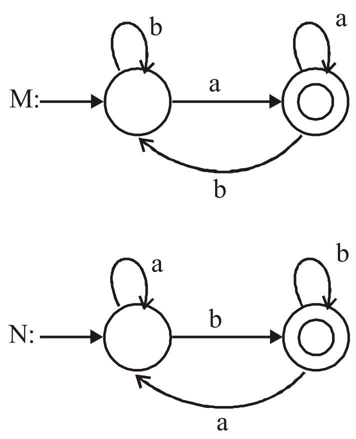
Then $\delta(q_2, aba)$ is
(a) ϕ (b) $\{q_0, q_1, q_3\}$
(c) $\{q_0, q_1, q_2\}$ (d) $\{q_0, q_2, q_3,\}$

3. Which one of the following regular expressions represents the language: *the set of all binary strings having two consecutive* 0*s and two consecutive* 1*s*?
[2016, Set 1, 1 Mark]
(a) (0+1)*0011(0+1)*+(0+1)*1100(0+1)*
(b) (0+1)*(00(0+1)*11+11(0+1)*00)(0+1)*
(c) (0+1)*00(0+1)*+(0+1)*11(0+1)*
(d) 00(0+1)*11+11(0+1)*00

4. The number of states in the minimum sized DFA that accepts the language defined by the regular expression $(0 + 1)*(0 + 1)(0 + 1)*$ is ________.
[2016, Set 2, 1 Mark]

5. Consider the following two statements:
I. If all states of an NFA are accepting states then the language accepted by the NFA is Σ^*.
II. There exists a regular language A such that for all languages B, $A \cap B$ is regular.
Which one of the following is **CORRECT**?
[2016, Set 2, 2 Marks]
(a) Only I is true
(b) Only II is true
(c) Both I and II are true
(d) Both I and II are false

6.



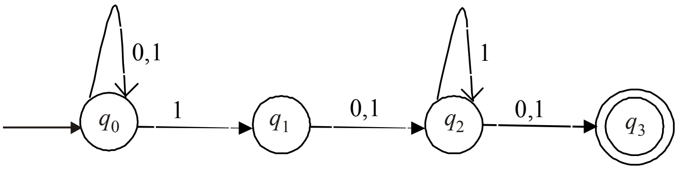
Consider the DFAs M and N given above. The number of states in a minimal DFA that accepts the languages $L(M) \cap L(N)$ is ________. **[2015, Set 1, 2 Marks]**

7. The number of states in the minimal deterministic finite automata corresponding to the regular expression $(0 + 1)*(10)$ is _______. **[2015, Set 2, 2 Marks]**

8. Consider the alphabet $\Sigma = \{0, 1\}$, the null/empty string λ and the sets of strings X_0, X_1, and X_2 generated by the corresponding non-terminals of a regular grammar X_0, X_1, and X_2 are related as follows.
$X_0 = 1\,X_1$
$X_1 = 0\,X_1 + 1\,X_2$
$X_2 = 0\,X_1 + \{\lambda\}$
Which one of the following choices precisely represents the strings in X_0? **[2015, Set 2, 2 Marks]**
(a) 10(0*+(10)*)1
(b) 10(0*+(10*)*1
(c) 1(0+10)*1
(d) 10(0+10)*1+110(0+10)*1

9. Let L be the language represented by the regular expression $\sum *0011\sum *$ where $\sum = \{0,1\}$.
What is the minimum number of states in a DFA that recognizes $\overline{L}$ (complement of L)? **[2015, Set 3, 1 Mark]**
(a) 4 (b) 5
(c) 6 (d) 8

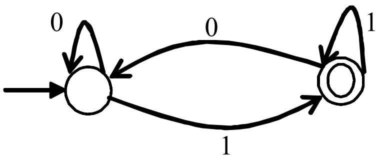
10. Consider the finite automata in the following figure.



Which is the set of reachable states for the input string 0011? **[2014, Set-1, 1 Mark]**

(a) $\{q_0, q_1, q_2\}$ (b) $\{q_0, q_1\}$
(c) $\{q_0, q_1, q_2, q_3\}$ (d) $\{q_3\}$

11. Which of the regular expressions given below represent the following DFA? **[2014, Set-1, 2 Marks]**



I 0*1(1+00*1)*
II 0*1*1+11*0*1
III (0+1)*1

(a) I and II only (b) I and III only
(c) II and III only (d) I, II, and III

12. Let $L_1 = \{w \in \{0, 1\}^* \mid w$ has at least as many occurrences of (110)'s as (011)'s$\}$. Let $L_2 = \{w \in \{0, 1\}^* \mid w$ has at least as many occurrences of (000)'s as (111)'s$\}$. Which one of the following is TRUE? **[2014, Set-2, 2 Marks]**

(a) L_1 is regular but not L_2
(b) L_2 is regular but not L_1
(c) Both L_1 and L_2 are regular
(d) Neither L_1 nor L_2 are regular

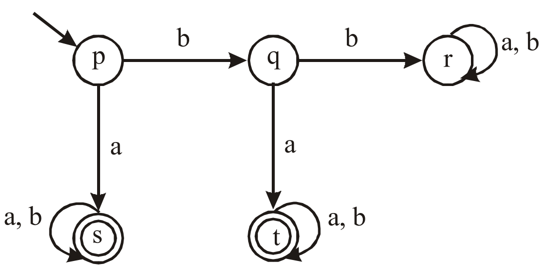
13. Let Σ be a finite non-empty alphabet and let 2^{Σ^*} be the power set of Σ^*. Which one of the following is TRUE? **[2014, Set-3, 1 Mark]**

(a) Both 2^{Σ^*} and Σ^* are countable
(b) 2^{Σ^*} is countable and Σ^* is uncountable
(c) 2^{Σ^*} is uncountable and Σ^* is countable
(d) Both 2^{Σ^*} and Σ^* are uncountable

14. The length of the shortest string NOT in the language (over $\Sigma = \{a, b\}$) of the following regular expression is ______________. **[2014, Set-3, 1 Mark]**

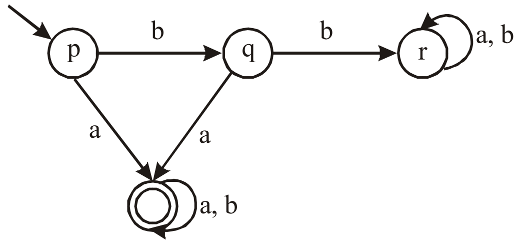
a*b* (ba)*a*

15. A deterministic finite automata (DFA)D with alphabet $\Sigma = \{a, b\}$ is given below. **[2011, 2 Marks]**

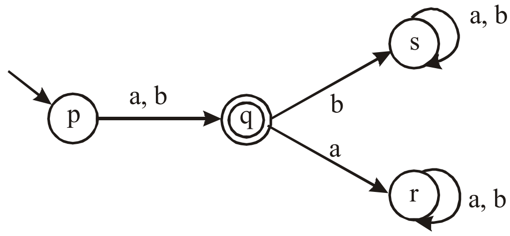


Which of the following finite state machines is a valid minimal DFA which accepts the same language as D?

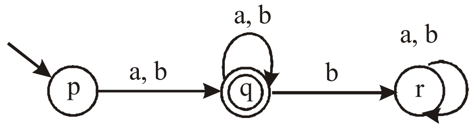
(a)



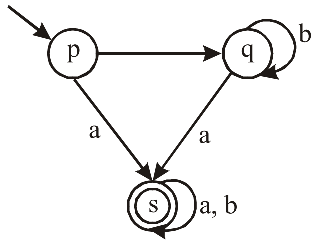
(b)



(c)



(d)



16. Definition of a language L with alphabet {a} is given as following and n is a positive integer constant. What is the minimum number of states needed in a DFA to recognize L? **[2011, 2 Marks]**

(a) k + 1 (b) n + 1
(c) 2n + 1 (d) 2k + 1

17. Let W be any string of length n in {0, 1}*. Let L be the set of all sub-strings of W. What is the minimum number of states in a non-deterministic finite automata that accept L? **[2010, 2 Marks]**

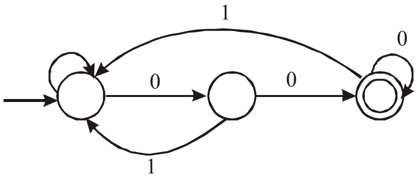
(a) n – 1 (b) n
(c) n + 1 (d) 2^{n-1}

18. Let $L = \{W \in (0, 1)^* \mid W$ has even number of 1s$\}$, i.e., L is the set of all bit strings with even number of 1's. Which one of the regular expressions below represents L? **[2010, 2 Marks]**

(a) (0* 10* 1)* (b) 0 * (10* 10*)*
(c) 0 * (10 *1)* 0* (d) 0 * 1(10 * 1) * 10 *

19. $S \to aSa \mid bSb \mid a \mid b$; The **language** generated by the above grammar over the alphabet {a, b} is the set of **[2009, 1 Mark]**

(a) all palindromes
(b) all odd length palindromes
(c) strings that begin and with the same symbol
(d) all even length palindromes

20. 

The above DFA accepts the set of all strings over {0, 1} that
(a) begin either with 0 or 1 **[2009, 2 Marks]**
(b) end with 0
(c) end with 00
(d) contain the substring 00

21. Given the following state table of an FSM with two states A and B, one input and one output

Present State A	Present State B	Input	Next State A	Next State B	Output
0	0	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	1	0	0
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	0	0	1

If the initial state is A = 0, B = 0, what is the minimum length of an input string, which will take the machine to the state A = 0, B = 1 with output = 1? **[2009, 2 Marks]**
(a) 3 (b) 4
(c) 5 (d) 6

22. 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) * ? **[2009, 1 Mark]**
(a) The set of all strings containing the substring 00
(b) The set of all strings containing at most two 0's
(c) The set of all strings containing at least two 0's
(d) The set of all strings that begin and end with either 0 or 1.

23. Which of the following are regular sets? **[2008, 2 Marks]**
1. $\left\{a^n b^{2m} \mid n \geq 0, m \geq 0\right\}$
2. $\left\{a^n b^m \mid n = 2m\right\}$
3. $\left\{a^n b^m \mid n \neq m\right\}$
4. $\{xcy \mid x, y, \in \{a, b\}^*\}$

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

24. Which of the following statements is false? **[2008, 2 Marks]**
(a) Every NFA can be converted to an equivalent DFA
(b) Every non-deterministic turing machine can be converted to an equivalent deterministic turing machine
(c) Every regular language is also a context-free language
(d) Every subset of a recursively enumerable set is recursive

25. Given below are two finite state automata (→ indicates the start state and F indicates a final state) **[2008, 2 Marks]**

Y:

	a	b
→ 1	1	2
2 (F)	2	1

Z:

	a	b
→ 1	2	2
2 (F)	1	1

Which of the following represents the product automata Z × Y?

(a)

	a	b
→ P	Q	R
Q	R	S
R(F)	Q	P
S	Q	P

(b)

	a	b
→ 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

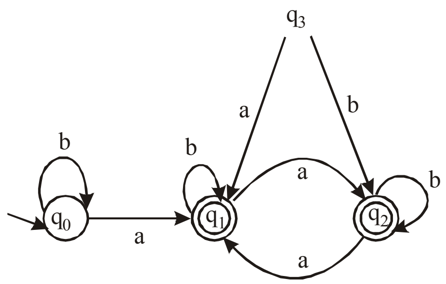
(d)

	a	b
→ P	S	Q
Q	S	R
R(F)	Q	P
S	Q	P

26. Which of the following is true? **[2007, 1 Mark]**
(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 regular
(d) Infinite union of finite sets is regular

27. A minimum state deterministic finite automata accepting the language L = {W | W ∈ {0, 1}*, number of 0's and 1's in W are divisible by 3 and 5, respectively as. **[2007, 2 Marks]**
(a) 15 states (b) 11 states
(c) 10 states (d) 9 states

Common Data for Questions 28 and 29 :
Consider the following Finite State Automata:

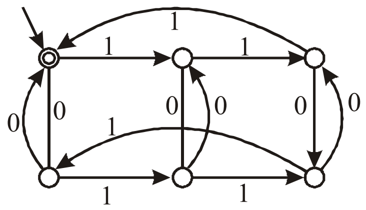


28. The language accepted by this automata is given by the regular expression **[2007, 2 Marks]**
(a) b * ab * ab * ab (b) (a + b)*
(c) b* a (a + b)* (d) b * ab * ab *

29. The minimum state automata equivalent to the above FSA has the following number of states **[2007, 2 Marks]**

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

30. The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively. **[2004, 2 Marks]**

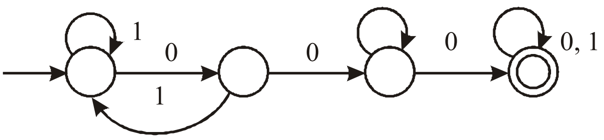


(a) divisible by 3 and 2 (b) odd and even
(c) even and odd (d) divisible by 2 and 3

31. The regular expression 0* (10)* denotes the same set as **[2003, 1 Mark]**

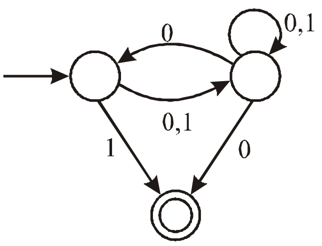
(a) (1* 0)*1* (b) 0 + (0 + 10)*
(c) (0 + 1)* 10 (0 + 1)* (d) None of these

32. Consider the following deterministic finite state automata M:
Let S denotes 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 **[2003, 2 Marks]**



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

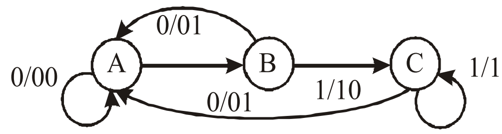
33. Consider the NFA, M, shown below:



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? **[2003, 2 Marks]**

(a) $L_1 = \{0, 1\}^* - L$ (b) $L_1 = \{0, 1\}^*$
(c) $L_1 \subseteq 1$ (d) $L_1 = L$

34. The finite state machine described by the following state diagram with A as starting state, where an arc label is $\frac{x}{y}$ and x stands for 1-bit input and y stands for 2 bit output **[2002, 2 Marks]**



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

35. The smallest finite automata which accepts the language {x| length of x is divisible by 3} has **[2002, 2 Marks]**

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

36. Given an arbitrary Non-deterministic Finite Automata (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least**[2001, 1 Mark]**

(a) N^2 (b) 2^N
(c) 2N (d) N!

37. Consider a DFA over Σ = {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? **[2001, 2 Marks]**

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

38. Let S and T be languages over Σ = {a, b} represented by the regular expressions (a + b*)* and (a + b)*, respectively. Which of the following is true? **[2000, 1 Mark]**

(a) $S \subset T$ (b) $T \subset S$
(c) S = T (d) $S \cap T = \phi$

Context Free Grammars and Pushdown Automata

39. Consider the following context-free grammar over the alphabet $\sum = \{a, b, c\}$ with S as the start symbol: **[2017, Set 1, 1 Mark]**

$S \to abScT \mid abcT$
$T \to bT \mid b$

Which one of the following represents the language generated by the above grammar?

(a) $\{(ab)^n(cb)^n \,]\, n \ge 1\}$
(b) $\{(ab)^n cb^{m_1} cb^{m_2} \dots cb^{m_n} \mid n, m_1 m_2 ..., m_n \ge 1\}$
(c) $\{(ab)^n (cb^m)^n \mid m, n \ge 1\}$
(d) $\{(ab)^n (cb^n)^m \mid m,n \ge 1\}$

40. Consider the following grammar: **[2017, Set 1, 1 Mark]**

$P \to xQRS$
$Q \to yz \mid z$
$R \to w \mid \varepsilon$
$S \to y$

What is FOLLOW(Q)?

(a) $\{R\}$ (b) $\{w\}$
(c) $\{w, y\}$ (d) $\{w, \$\}$

41. If G is a grammar with productions **[2017, Set 1, 2 Marks]**
$S \rightarrow SaS \mid aSb \mid bSa \mid SS \mid \in$
where S is the start variable, then which one of the following strings is not generated by G?
(a) *abab* (b) *aaab*
(c) *abbaa* (d) *babba*

42. Consider the context-free grammars over the alphabet $\{a, b, c\}$ given below. S and T are non-terminals.
[2017, Set 1, 2 Marks]
$G_1: S \rightarrow aSb/T,\ T \rightarrow cT|\in$
$G_2: S \rightarrow bSa \backslash T,\ T \rightarrow cT|\in$
The language $L(G_1) \cap L(G_2)$ is
(a) Finite.
(b) Not finite but regular.
(c) Context-Free but not regular.
(d) Recursive but not context-free.

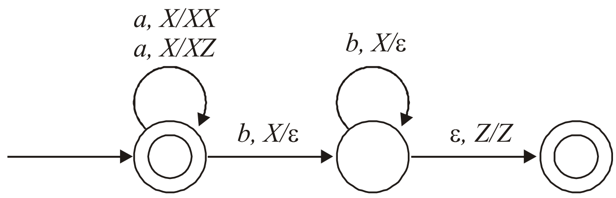
43. Consider the following grammar: **[2017, Set 1, 2 Marks]**
stmt - > **if** expr **then** expr **else** expr; stmt | o
expr - > term **relop** term | term
term - > id | number
id - > **a** | **b** | **c**
number - > [**0-9**]
where relop is a relational operator (e.g., <, >, ...), o refers to the empty statement, and if, then, else are terminals.
Consider a program P following the above grammar containing ten if terminals. The number of control flow paths in P is ________. For example, the program
if e_1 **then** e_2 **else** e_3
has 2 control flow paths, $e_1 \rightarrow e_2$ and $e_1 \rightarrow e_3$.

44. Consider the following expression grammar G :
[2017, Set 2, 2 Marks]
E → E – T | T
T → T + F | F
F → (E) | id
Which of the following grammars is not left recursive, but is equivalent to G?
(a) E → E – T | T
T → T + F | F
F → (E) | id
(b) E → TE'
E' → –TE' | ∈
T → T + F | F
F → (E) | id
(c) E → TX
X → –TX | ∈
T → FY
Y → +FY | ∈
E → (E) | id
(d) E → TX | (TX)
X → –TX | +TX | ∈
T → –id

45. Which of the following decision problems are undecidable?
[2016, Set 1, 1 Mark]
I. Given NFAs N_1 and N_2, is $L(N_1) \cap L(N_2) = \Phi$?
II. Given a CFG $G = (N, \Sigma, P, S)$ and a string $x \in \Sigma^*$, does $x \in L(G)$?
III. Given CFGs G_1 and G_2 is $L(G_1) = L(G_2)$?
IV. Given a TM M, is $L(M) = \Phi$?
(a) I and IV only (b) II and III only
(c) III and IV only (d) II and IV only

46. Consider the following context-free grammars:
[2016, Set 1, 2 Mark]
$G_1: S \rightarrow aS \mid B,\ \ B \rightarrow b \mid bB$
$G_2: S \rightarrow aA \mid bB,\ \ A \rightarrow aA \mid B \mid \varepsilon,\ \ B \rightarrow bB \mid \varepsilon$
Which one of the following pairs of languages is generated by G_1 and G_2, respectively?
(a) $\{a^m b^n | m > 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$
(b) $\{a^m b^n | m > 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n \geq 0\}$
(c) $\{a^m b^n | m \geq 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$
(d) $\{a^m b^n | m \geq 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n > 0\}$

47. Consider the transition diagram of a PDA given below with input alphabet $\Sigma = \{a, b\}$ and stack alphabet $\Gamma = \{X, Z\}$. Z is the initial stack symbol. Let L denote the language accepted by the PDA. **[2016, Set 1, 2 Mark]**



Which one of the following is **TRUE**?
(a) $L = \{a^n b^n | n \geq 0\}$ and is not accepted by any finite automata
(b) $L = \{a^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$ and is not accepted by any deterministic PDA
(c) L is not accepted by any Turing machine that halts on every input
(d) $L = \{a^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$ and is deterministic context-free

48. Language L_1 is defined by the grammar: $S_1 \rightarrow aS_1b|\varepsilon$
Language L_2 is defined by the grammar: $S_2 \rightarrow abS_2|\varepsilon$
Consider the following statements:
P: L_1 is regular
Q: L_2 is regular
Which one of the following is TRUE?
[2016, Set 2, 1 Mark]
(a) Both P and Q are true
(b) P is true and Q is false
(c) P is false and Q is true
(d) Both P and Q are false

49. Which one of the following grammars is free from left recursion? **[2016, Set 2, 2 Marks]**
(a) S → AB
A → Aa | b
B → c
(b) S → Ab | Bb | c
A → Bd | ε
B → ε
(c) S → Aa | B
A → Bb | Sc | ε
B → d
(d) S → Aa | Bb | c
A → Bd | ε
B → Ae | ε

50. A student wrote two context-free grammars **G1** and **G2** for generating a single C-like array declaration. The dimension of the array is at least one. For example,

int a [10] [3];

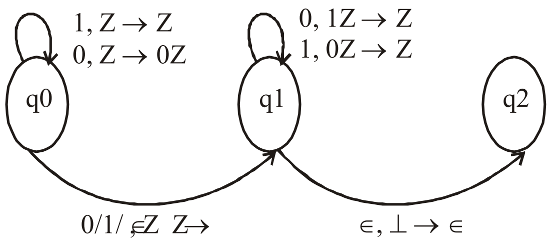
The grammars use D as the start symbol, and use six terminal symbols int ; id [ ]num.

Grammar **G1**	Grammar **G2**
$D \rightarrow$ **int** L;	$D \rightarrow$ **int** L;
$L \rightarrow$ **id** [E	$L \rightarrow$ **id** E
$E \rightarrow$ **num**]	$E \rightarrow$ E[**num**]
$E \rightarrow$ **num**] [E	$E \rightarrow$ [**num**]

Which of the grammars correctly generate the declaration mentioned above? **[2016, Set 2, 2 Marks]**

(a) Both **G1** and **G2** (b) Only **G1**
(c) Only **G2** (d) Neither **G1** nor **G2**

51. Consider the NPDA < Q = {q0, q1, q2}, Σ = {0, 1}, Γ = {0, 1,}⊥δ, q0, ⊥, F = {q2}>, where (as per usual convention) Q is the set of states, Σ is the input alphabetδ is the stack alphabet, δ is the state transition function, q0 is the initial state, ⊥ is the initial stack symbol, and F is the set of accepting states. The state transition is as follows:



Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automata? **[2015, Set 1, 2 Marks]**

(a) 10110 (b) 10010
(c) 01010 (d) 01001

52. In the context of Abstract-Systax-Tree (AST) and Control-Flow-Graph (CFG), which one of the following is TRUE? **[2015, Set 2, 1 Mark]**

(a) In both AST and CFG, let node N_2 be the successor of node N_1. In the input program, the code corresponding to N_2 is present after the code corresponding to N_1
(b) For any input program, neither AST nor CFG will contain a cycle
(c) The maximum number of successors of a node in an AST and a CFG depends on the input program
(d) Each node in AST and CFG corresponds to at most one statement in the input program.

53. Which of the following is/are undecidable?

1. G is a CFG. Is L(G) = Φ?
2. G is a CFG. Is L(G) = Σ*?
3. M is a Turing machine. Is L(M) regular?
4. A is a DFA and N is an NFA. Is L(A) = L(N)?

[2013, 2 Marks]

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

54. Which of the following pairs have different expressive power? **[2011, 1 Mark]**

(a) Deterministic Finite Automata (DFA) and Non-deterministic Finite Automata (NFA)
(b) Deterministic Push Down Automata (DPDA) and Non-deterministic Push Down Automata (NPDA)
(c) Deterministic single-tape turing machine and non-deterministic single tape turing machine
(d) Single-tape turing machine and multi-tape turing machine

55. Which one of the following is false? **[2009, 1 Mark]**

(a) There is unique minimal DFA for every regular language
(b) Every NFA can be converted to an equivalent PDA
(c) Complement of every context-free language is recursive
(d) Every non-deterministic PDA can be converted to an equivalent deterministic PDA

56. Which of the following statements are true? **[2008, 2 Marks]**

1. Every left-recursive grammar can be converted to a right-recursive grammar and vice-versa.
2. All ε-productions can be removed from any context-free grammar by suitable transformations.
3. The language generated by a context-free grammar all of whose productions are of the form X → W or X → WY (where, W is a string of terminals and Y is non-terminal), is always regular.
4. The derivation trees of strings generated by a context-free grammar in Chomsky Normal Form are always binary trees.

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

57. Which of the following problems is undecidable? **[2007, 1 Mark]**

(a) Membership problem for CFGs
(b) Ambiguity problem for CFGs
(c) Finiteness problem for FSAs
(d) Equivalent problem for FSAs

58. Consider the following statements about the context-free grammar: **[2006, 2 Marks]**

G = {S → SS, S → ab, S → ba, S → ∈}

1. G is ambiguous.
2. G produces all strings with equal number of a's and b's.
3. G can be accepted by a deterministic PDA.

Which combination below expresses all the true statements about G?

(a) 1 only (b) 1 and 3
(c) 2 and 3 (d) 1, 2 and 3

59. Let N_f and N_p denote the classes of languages accepted by non-deterministic finite automata and non-deterministic push-down automata, respectively. Let D_f and D_p denote the classes of languages accepted by deterministic finite automata and deterministic push-down automata respectively. Which one of the following is true? **[2005, 2 Marks]**

(a) $D_f \subset N_f$ and $D_p \subset N_p$ (b) $D_f \subset N_f$ and $D_p = N_p$
(c) $D_f = N_f$ and $D_p = N_p$ (d) $D_f = N_f$ and $D_p \subset N_p$

60. Consider the following grammar G:

$S \rightarrow bS|aA|b$

$A \rightarrow bA\ |aB$

$B \rightarrow bB|aS|\ a$

Let N_a (W) and N_b (W) denote the number of a's and b's in a string W respectively.

The language $L(G) \subseteq \{a, b\}^+$ generated by G is

(a) $\{W|N_a(W) > 3N_b(W)\}$ **[2004, 2 Marks]**

(b) $\{W|N_b(W) > 3N_a(W)\}$

(c) $\{W|N_a(W) = 3k, k\in\{0, 1, 2,...\}\}$

(d) $\{W|N_b\ (W) = 3k, k \in\{0, 1, 2,...\}\}$

61. Let G = ({S}, {a, b}, R, S) be a context-free grammar where the rule set R is $S\rightarrow aSb|SS|\ \in$

Which of the following statements is true?

[2003, 2 Marks]

(a) G is not ambiguous

(b) There exist X, Y $\in$L(G) such that XY $\in$L(G)

(c) There is a deterministic push-down automata that accepts L(G)

(d) We can find a deterministic finite state automata that accepts L(G)

62. Consider the following decision problems:

[2000, 2 Marks]

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 statements is true?

(a) Both 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

Regular and Context Free Language

63. Consider the following languages over the alphabet $\Sigma = \{a, b, c\}$. **[2017, Set 1, 2 Marks]**

Let $L_1 = \{ a^n b^n c^m \mid m,n \geq 0\}$ and $L_2 = \{ a^m b^n c^n \mid m, n \geq 0\}$.

Which of the following are context-free languages?

I. $L_1 \cup L_2$

I. $L_1 \cap L_2$

(a) I only (b) II only

(c) I and II (d) Neither I nor II

64. Let L_1 L_2 be any two context-free languages and R be any regular language. Then which of the following is/are **CORRECT**? **[2017, Set 2, 1 Mark]**

I. $L_1 \cup L_2$ is context-free.

II. L_1 is context-free.

III. $L_1 - R$ is context-free.

IV. $L_1 \cap L_2$ is context-free.

(a) I. II and IV only (b) I and III only

(c) II and IV only (d) I only

65. Identify the language generated by the following grammar, where S is the start variable. **[2017, Set 2, 1 Mark]**

$S \rightarrow XY$

$X \rightarrow aX \mid a$

$Y \rightarrow aYb \mid \in$

(a) $\{a^m b^n \mid m \geq n, n > 0\}$

(b) $\{a^m b^n \mid m \geq n, n \geq 0\}$

(c) $\{a^m b^n \mid m > n, n \geq 0\}$

(d) $\{a^m b^n \mid m > n, n > 0\}$

66. The minimum possible number of states of a deterministic finite automata that accepts the regular language $L = \{w_1 a w_2 | w_1, w_2 \in \{a, b\}^*, |w_1| = 2, |w_2| \geq 3\}$ is ________.

[2017, Set 2, 1 Mark]

67. Consider the following languages.

[2017, Set 2, 2 Marks]

$L_1 = \{a^p \mid p$ is a prime number$\}$

$L_2 = \{a^n b^m c^{2m} \mid n \geq 0, m \geq 0\}$

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

$L_4 = \{a^n b^n \mid n \geq 1\}$

Which of the following are **CORRECT**?

I. L_1 is context-free but not regular.

II. L_2 is not context-free.

III. L_3 is not context-free but recursive.

IV. L_4 is deterministic context-free.

(a) I, II and IV only (b) II and III only

(c) I and IV only (d) III and IV only

68. Which of the following languages is generated by the given grammar? **[2016, Set 1, 1 Mark]**

$S \rightarrow aS \mid bS \mid \varepsilon$

(a) $\{a^n b^m | n, m \geq 0\}$

(b) $\{w \in \{a, b\}^* | w$ has equal number of a's and b's$\}$

(c) $\{a^n | n \geq 0\} \cup \{b^n \mid n \geq 0\} \cup \{a^n b^n | n \geq 0\}$

(d) $\{a, b\}^*$

69. Let X be a recursive language and Y be a recursively enumerable but not recursive language. Let W and Z be two languages such that $\overline{Y}$ reduces to W, and Z reduces to $\overline{X}$ (reduction means the standard many-one reduction). Which one of the following statements is **TRUE**?

[2016, Set 1, 2 Mark]

(a) W can be recursively enumerable and Z is recursive.

(b) W can be recursive and Z is recursively enumerable.

(c) W is not recursively enumerable and Z is recursive.

(d) W is not recursively enumerable and Z is not recursive.

70. Consider the following types of languages: L_1 : Regular, L_2 : Context-free, L_3 : Recursive, L_4 : Recursively enumerable. Which of the following is/are TRUE?

[2016, Set 2, 1 Mark]

I. $\overline{L_3} \cup L_4$ is recursively enumerable

II. $\overline{L_2} \cup L_3$ is recursive

III. $L_1^* \cap L_2$ is context-free

IV. $L_1 \cup \overline{L_2}$ is context-free

(a) I only

(b) I and III only

(c) I and IV only

(d) I, II and III only

71. Consider the following languages:
$L_1 = \{a^n b^m c^{n+m} : m, n \geq 1\}$
$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$
Which one of the following is **TRUE**?
[2016, Set 2, 2 Marks]
(a) Both L_1 and L_2 are context-free.
(b) L_1 is context-free while L_2 is not context-free.
(c) L_2 is context-free while L_1 is not context-free.
(d) Neither L_1 nor L_2 is context-free.

72. For any two languages L_1 and L_2 such that L_1 is context-free and L_2 is recursively enumerable but not recursive, which of the following is/are necessarily true?
I. $\overline{L_1}$ (complement of L_1) is recursive
II. $\overline{L_2}$ (complement of L_2) is recursive
III. $\overline{L_1}$ is context – free
IV. $\overline{L_1} \cup L_2$ is recursively enumerable
[2015, Set 1, 1 Mark]
(a) I only (b) III only
(c) III and IV only (d) I and IV only

73. Which of the following languages is/are regular?
L_1: $\{wxw^R | w, x \in \{a, b\}^*$ and $|w|, |x| > 0\}$, w^R is the reverse of string w
L_2: $\{a^n b^m \mid m \neq n$ and $m, n \geq 0\}$
L_3: $\{a^p b^q c^r \mid p, q, r \geq 0\}$ **[2015, Set 2, 2 Marks]**
(a) L_1 and L_3 only (b) L_2 only
(c) L_2 and L_3 only (d) L_3 only

74. Which of the following languages are context-free?
$L_1 = \{a^m b^n a^n b^m \mid m, n \geq 1\}$
$L_2 = \{a^m b^n a^m b^n \mid m, n \geq 1\}$
$L_3 = \{a^m b^n \mid m = 2n+1\}$ **[2015, Set 3, 2 Marks]**
(a) L_1 and L_2 only (b) L_1 and L_3 only
(c) L_2 and L_3 only (d) L_3 only

75. Which one of the following is **TRUE** ?
[2014, Set-1, 1 Mark]
(a) The language $L = \{a^n b^n | \geq 0\}$ is regular.
(b) The language $L = \{a^n | \ n$ is prime$\}$ is regular.
(c) The language $L = \{w \mid w$ has 3k + 1b's for some $k \in N$ with $\Sigma = \{a, b\}\}$ is regular.
(d) The language $L = \{ww \mid w \in \Sigma^*$ with $\Sigma = \{0, 1\}\}$ is regular.

76. Let $A \leq_m B$ denotes that language A is mapping reducible (also known as many-to-one reducible) to language B. Which one of the following is FALSE?
[2014, Set-2, 1 Mark]
(a) If $A \leq_m B$ and B is recursive then A is recursive.
(b) If $A \leq_m B$ and A is undecidable then B is undecidable.
(c) If $A \leq_m B$ and B is recursively enumerable then A is recursively enumerable.
(d) If $A \leq_m B$ and B is not recursively enumerable then A is not recursively enumerable.

77. Let L be a language and $\overline{L}$ be its complement. Which one of the following is NOT a viable possibility?
[2014, Set-1, 2 Marks]
(a) Neither L nor $\overline{L}$ is recursively enumerable (r.e.).
(b) One of L and $\overline{L}$ is r.e. but not recursive; the other is not r.e.
(c) Both L and $\overline{L}$ are r.e. but not recursive.
(d) Both L and $\overline{L}$ are recursive.

78. If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, consider
[2014, Set-2, 1 Mark]
(I) $L_1.L_2$ is a regular language
(II) $L_1.L_2 = \{a^n b^n \mid n \geq 0\}$
Which one of the following is CORRECT?
(a) Only (I) (b) Only (II)
(c) Both (I) and (II) (d) Neither (I) nor (II)

79. Consider the following languages over the alphabet $\Sigma = \{0, 1, c\}$:
$L_1 = \{0^n 1^n \mid n \geq 0\}$
$L_2 = \{wcw^r \mid w \in \{0, 1\}^*\}$
$L_3 = \{ww^r \mid w \in \{0, 1\}^*\}$
Here, w^r is the reverse of the string w. Which of these languages are deterministic Context-free languages?
[2014, Set-3, 2 Marks]
(a) None of the languages
(b) Only L_1
(c) Only L_1 and L_2
(d) All the three languages

80. Consider the following languages:
$$L_1 = \left\{0^p 1^q 0^r \mid p, q, r \geq 0 \right\}$$
$$L_2 = \left\{0^p 1^q 0^r \mid p, q, r \geq 0, p \neq r \right\}$$
Which one of the following statements is false?
[2013, 2 Marks]
(a) L_2 is context-free
(b) $L_1 \cap L_2$ is context-free
(c) Complement of L_2 is recursive
(d) Complement of L_1 is context-free but not regular

81. Consider the languages $L_1 = \Phi$ and $L_2 = \{a\}$. Which one of the following represents $L_1 L_2^* \cup L_1^*$? **[2013, 1 Mark]**
(a) $\{\varepsilon\}$ (b) Φ
(c) a^* (d) (ε, a)

82. Let W(n) and A(n) denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. Which of the following is always true?
[2012, 1 Mark]
(a) $A(n) = \Omega(W(n))$ (b) $A(n) = \Theta(W(n))$
(c) $A(n) = O(W(n))$ (d) $A(n) = o(W(n))$

83. Consider the languages, L_1, L_2 and L_3 as given below
$$L_1 = \left\{0^p 1^q \mid p, q \in N\right\}$$ **[2011, 2 Marks]**
$$L_2 = \left\{0^p 1^q \mid p, q \in N \text{ and } p = q\right\} \text{ and}$$

$L_3 = \{0^p 1^q 0^r \mid p, q, r \in N \text{ and } p = q = r\}$

Which of the following statements is not true?

(a) Push Down Automata (PDA) can be used to recognize L_1 and L_2
(b) L1 is a regular language.
(c) All the three languages are context-free
(d) Turning machines can be used to recognize all the languages

84. Let P be a regular language and Q be a context free language such that $Q \subseteq P$. For example, let P be the language represented by the regular expression p*q* and Q be $(Q^n p^n q^n | n \in N)$. Then which of the following is always regular? **[2011, 1 Mark]**

(a) $P \cap Q$ (b) $P - Q$
(c) $\sum^* - P$ (d) $\sum^* - Q$

85. Let L_1 be a recursive **language**. Let L_2 and L_3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true? **[2010, 1 Mark]**

(a) $L_2 - L_1$ is recursively enumerable
(b) $L_1 - L_3$ is recursively enumerable
(c) $L_2 \cap L_1$ is recursively enumerable
(d) $L_2 \cup L_1$ is recursively enumerable

86. Consider the languages $L_1 = \{0^i 1^j \mid i \neq j\}$, $L_2 = \{0^i 1^j \mid i = j\}$, $L_3 = \{0^i 1^j \mid i = 2j + 1\ j\}$, $L_4 = \{0^i 1^j \mid i \neq 2j\}$. Which one of the following statements is true? **[2010, 2 Marks]**

(a) Only L_2 is context-free
(b) L_2 and L_3 are context-free
(c) L_1 and L_2 is context-free
(d) All are context free

87. Let $L_1 = L_1 \cap L_2$, where L_1 and L_2 are languages as defined below

$L_1 = \{a^m b^m c\, a^n b^n \mid m, n \geq 0\}$

$L_2 = \{a^i b^j c^k \mid i, j, k \geq 0\}$

Then L is **[2009, 2 Marks]**

(a) not recursive
(b) regular
(c) context-free but not regular
(d) recursively enumerable but not context-free

88. If L and $\overline{L}$ are recursively enumerable, then L is **[2008, 1 Mark]**

(a) regular (b) context-free
(c) context-sensitive (d) recursive

89. Which of the following are decidable? **[2008, 1 Mark]**

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

90. Which of the following is true for the language $\{a^p \mid p \text{ is a prime}\}$? **[2008, 1 Mark]**

(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

91. The language $L = \{0^i 2 1^i \mid i \geq 0\}$ over the alphabet {0, 1, 2} is

(a) not recursive **[2007, 2 Marks]**
(b) is recursive and is a deterministic CFL
(c) is a regular language
(d) is not a deterministic CFL but a CFL

92. Which of the following languages is regular?

(a) $\{WW^R \mid W \in \{0,1\}^+\}$ **[2007, 2 Marks]**
(b) $\{WW^R X \mid X, W \in \{0,1\}^+\}$
(c) $\{WXW^R \mid X, W \in \{0,1\}^+\}$
(d) $\{XWW^R \mid X, W \in \{0,1\}^+\}$

93. Let L_1 be a regular language, L_2 be a deterministic context-free language and L_3 a recursively enumerable but not recursive language. Which one of the following statements is false? **[2006, 1 Mark]**

(a) $L_1 \cap L_2$ is a deterministic CFL
(b) $L_2 \cap L_1$ is recursive
(c) $L_1 \cap L_2$ is context-free
(d) $L_1 \cap L_2 \cap L_3$ is recursively enumerable

94. For $S \in (0 + 1)^*$ let d(s) denotes the decimal value of s (e.g., d(101) = 5) **[2006, 1 Mark]**

Which one of the following statements is true?

(a) L is recursively enumerable but not recursive
(b) L is recursively but not context-free
(c) L is context-free but not regular
(d) L is regular

95. If s is a string over (0 + 1)* then let $n_0(s)$ denotes 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? **[2006, 2 Marks]**

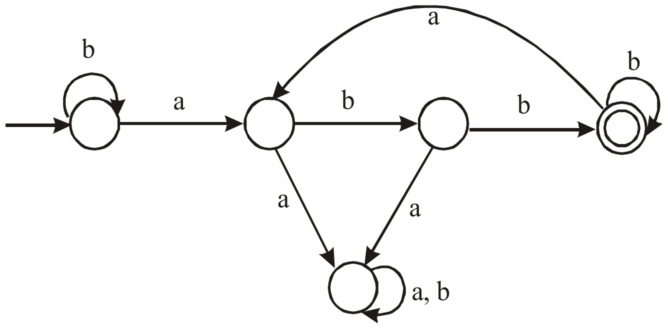
(a) $L = \{s \in (0 + 1)^* \mid n_0(s) \text{ is a 3-digit prime}\}$
(b) $L = \{s \in (0 + 1)^* \mid \text{for every prefix } s' \text{ of } s, |n_0(s') - n_1(s') \leq 2\}$
(c) $L = \{s \in (0 + 1)^* \mid n_0(s) - n_1(s)| \leq 4$
(d) $L = \{s \in (0 + 1)^* \mid n_0(s) \bmod 7 = n_1(s) \bmod 5 = 0\}$

96. Let $L_1 = \{0^{n+m}1^n0^m \mid n, m \geq 0\}$, $L_2 = \{0^{n+m}1^{n+m}0^m \mid n, m \geq 0\}$, and $L_3 = \{0^{n+m}1^{n+m}0^{n+m} \mid n, m \geq 0\}$. Which of these languages is/are not context free? **[2006, 1 Mark]**
(a) L_1 only (b) L_3 only
(c) L_1 and L_2 (d) L_2 and L_3

97. Consider the languages: **[2005, 2 Marks]**
$L_1 = \{WW^R \mid W \in \{0, 1\}^*\}$
$L_2 = \{W \# W^R \mid W \in \{0, 1\}^*\}$ where # is a special symbol
$L_3 = \{W W \mid W \in \{0, 1\}^*\}$
Which one of the following is true?
(a) L_1 is a deterministic CFL
(b) L_2 is a deterministic CFL
(c) L_3 is a CFL but not a deterministic CFC
(d) L_3 is a deterministic CFL

98. 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? **[2005, 2 Marks]**
(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 recursive enumerable and $\overline{L_2}$ is recursive

99. Consider the machine M: **[2005, 2 Marks]**



The language recognized by M is
(a) $\{W \in (a, b)^* \mid$ every a in W is followed by exactly two b's$\}$
(b) $\{W \in (a, b)^* \mid$ every a in W is followed by at least two b's$\}$
(c) $\{W \in (a, b)^* \mid$ W contains the substring abb$\}$
(d) $\{W \in (a, b)^* \mid$ W does not contain aa as a substring$\}$

100. L_1 is a recursively enumerable language over Σ. An algorithm A effectively enumerates its words as $W_1, W_2, W_3, \ldots$ define another language L_2 over $\Sigma \cup \{\#\}$ as $\{W_i \# W_j : W_1, W_j \in L_1, i < j\}$. Here, # is a new symbol. Consider the following assertions.
S_1 –L_1 is recursive implies L_2 is recursive
S_2 –L_2 is recursive implies L_1 is recursive
Which of the following statements is true? **[2004, 2 Marks]**
(a) Both S_1 and S_2 are true
(b) S_1 is true but S_2 is not necessarily true
(c) S_2 is true but S_1 is not necessarily true
(d) Neither is necessarily true

101. The language $\{a^mb^nc^{m+n} \mid m,n \geq 1\}$ is
(a) regular **[2004, 2 Marks]**
(b) context-free but not regular
(c) context-senstitive but not context-free
(d) type-0 but not context-sensitive

102. If the strings of a language L can be effectively enumerated in lexicographic (i.e., alphabetic) order, which of the following statements is true? **[2003, 1 Mark]**
(a) L is necessarily finite
(b) L is regular but not necessarily finite
(c) L is context-free but not necessarily regular
(d) L is recursive but not necessarily context-free

103. Consider two languages L_1 and L_2, each on the alphabet Σ. Let $f : \Sigma \rightarrow \Sigma$ be a polynomial time computable bijection such that $(\forall X)$ $[X \in L_1$ iff $f(X) \in L_2]$.
Further, let f^{-1} be also polynomial time computable.
Which of the following cannot be true?
[2003, 2 Marks]
(a) $L_1 \in P$ and L_2 is finite
(b) $L_1 \in NP$ and $L_2 \in P$
(c) L_1 is undecidable and L_2 is decidable
(d) L_1 is recursively enumerable and L_2 is recursive

104. Define languages L_0 and L_1 as follows
$L_0 = \{< M, W, 0 > \mid M$ halts on $W\}$
$L_1 = \{<M, W, 1> \mid$ M does not halts on $W\}$
Here $< M, W, i >$ is a triplet, whose first component, M is an encoding of a turing machine, second component W is a string and third component *i* is a bit.
Let $L = L_0 \cup L_1$. Which of the following is true?
[2003, 2 Marks]
(a) L is recursively enumerable, but $\bar{L}$ is not
(b) $\bar{L}$ is recursively enumerable, but L is not
(c) Both L and $\bar{L}$ are recursive
(d) Neither L nor $\bar{L}$ is recursively enumerable

105. The language accepted by a push-down automata in which the stack is limited to 10 items is best described as
(a) context-free **[2002, 1 Mark]**
(b) regular
(c) deterministic context-free
(d) recursive

106. Which of the following is true? **[2002, 1 Mark]**
(a) The complement of a recursive enumerable language is recursive
(b) The complement of a recursively enumerable language is recursively enumerable
(c) The complement of a recursive language is either recursive or recursively enumerable
(d) The complement of a context-free language is contxt-free

107. The C language is **[2002, 1 Mark]**

(a) a context-free language
(b) a context-sensitive language
(c) a regular language
(d) parsable fully only be a turing maching

108. Consider the languages **[2002, 2 Marks]**

$$L_1 = \left\{a^n b^n c^m \mid n, m > 0\right\} \text{ and } L_2 = \left\{a^n b^m c^m \mid n, m > 0\right\}$$

Which one of the following statements is false?

(a) $L_1 \cap L_2$ is a context-free language
(b) $L_1 \cup L_2$ is a context-free language
(c) L_1 and L_2 are context-free language
(d) $L_1 \cap L_2$ is a context-sensitive language

109. Which of the following statements is true? **[2001, 1 Mark]**

(a) If a language is context-free it can always be accepted by a deterministic push-down automata
(b) The union of two context-free languages is context-free
(c) The intersection of two context-free languages is context-free
(d) The complement of a context-free language is context-free

110. Consider the following languages:

$L_1 = \{WW \mid W \in \{a, b\}^*\}$
$L_2 = \{WW^R \mid W \in \{a, b\}^*, W^R$ is the reverse of W$\}$
$L_3 = \{0^{2i} \mid$ i si an integer$\}$
$L_4 = \{0^{i^2} \mid$ i is an integer$\}$

Which of the languages are regular? **[2001, 2 Marks]**

(a) L_1 and L_2 (b) L_2, L_3 and L_4
(c) L_3 and L_4 (d) Only L_3

111. Consider the following two statements:

$S_1 - \{0^{2n} \mid n \geq 1|\}$ is regular language.
$S_2 - \{0^m 1^n 0^{m+n} \mid m \geq 1$ and $n \geq 1|\}$ is a regular language

Which of the following statements is correct?

(a) Only S_1 is correct **[2001, 1 Mark]**
(b) Only S_2 is correct
(c) Both S_1 and S_2 are correct
(d) None of these

112. Let L denotes the language generated by the grammar $S \to 0S0/00$. Which of the following is true?

(a) $L = 0^+$ **[2000, 1 Mark]**
(b) L is regular but not 0^+
(c) L is context-free but not regular
(d) L is not context-free

113. What can be said about a regular language L over {a} whose minimal finite state automaton has two states?

(a) L must be $\{a^n \mid n$ is odd$\}$ **[2000, 2 Marks]**
(b) L must be $\{a^n \mid n$ is even$\}$
(c) L must be $\{a^n \mid \geq 0\}$
(d) Either L must be $\{a^n \mid n$ is odd$\}$, or L must be $\{a^n \mid n$ is even$\}$

Pumping Lemma and NP Completeness

114. Match the List I with List II and select the correct answer using the codes given below the lists. **[2008, 2 Marks]**

[redacted]

(a) E - P, F - R, G - Q, H - S
(b) E - R, F - P, G - S, H - Q
(c) E - R, F - P, G - Q, H - S
(d) E - P, F - R, G - S, H - Q

115. Let $SHAM_3$ be the problem of finding a Hamiltonian cycle in a graph G = (V, E) with |V| divisible by 3 and $DHAM_3$ be the problem of determining if a Hamiltonian cycle exists in such graphs. Which one of the following is true? **[2006, 2 Marks]**

(a) Both $DHAM_3$ and $SHAM_3$ are NP-hard
(b) $SHAM_3$ are NP-hard but $DHAM_3$ is not
(c) $DHAM_3$ is NP-hard, and $SHAM_3$ is not
(d) Neither $DHAM_3$ nor $SHAM_3$ is NP-hard

116. Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial time reducible to R. Which one of the following statements is true? **[2006, 1 Mark]**

(a) R is NP-complete (b) R is NP-hard
(c) Q is NP-complete (d) Q is NP-hard

117. Consider the following two problems on undirected graphs:

α — Given, G (V, E), does G have an independent set of size |V| – 4?
β — Given, G (V, E), does G have an independent set of size 5?

Which one of the following is true? **[2005, 2 Marks]**

(a) α is in P and β is NP-complete
(b) α is NP-complete β is in P
(c) Both α and β are NP-complete
(d) Both α and β are in P

118. The problems 3-SAT and 2-SAT are **[2004, 1 Mark]**

(a) both in P
(b) both NP-complete
(c) NP-complete and in P respectively
(d) undecidable and NP-complete respectively

119. Ram and Shyam have been asked to show that a certain problem Π is NP-complete. Ram shows a Polynomial time reduction from the 3-SAT problem to II, and Shyam shows a polynomial time reduction from $\prod$ to 3-SAT. Which of the following can be inferred from these reductions? **[2003, 1 Mark]**

(a) $\prod$ is NP-hard but not NP-complete
(b) $\prod$ is in NP, but is not NP-complete
(c) $\prod$ is NP-complete
(d) $\prod$ is neither NP-hard, nor in NP

120. Nobody knows yet, if P = NP. Consider the language L de as follows:

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

Which of the following statements is true? **[2003, 1 Mark]**
(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

Turing Machines

121. Let A and B be finite alphabets and let # be a symbol outside both A and B. Let f be a total function from A^* to B^*. We say f is *computable* if there exists a Turing machine M which given an input x in A^*. always halts with $f(x)$ on its tape. Let L_f denote the language $\{x \# f(x) | \ x \in A^*\}$. Which of the following statements is true: **[2017, Set 1, 2 Marks]**
(a) f is computable if and only if L_f is recursive.
(b) f is computable if and only if L_f is recursively enumerable.
(c) If f is computable then L_f is recursive, but not conversely.
(d) If f is computable then L_f is recursively enumerable, but not conversely.

122. Let $L(R)$ be the language represented by regular expression R. Let $L(G)$ be the language generated by a context free grammar G. Let $L(M)$ be the language accepted by a Turing machine M. Which of the following decision problems are undecidable? **[2017, Set 2, 2 Marks]**
I. Given a regular expression R and a string w, is $w \in L(R)$?
II. Given a context-free grammar G. is $L(G) = \phi$?
III. Given a context-free grammar G. is $L(G) = \Sigma^*$ for some alphabet Σ ?
IV. Given a Turing machine M and a string w, is $w \in L(M)$?
(a) I and IV only (b) II and III only
(c) II, III and IV only (d) III and IV only

123. Consider the following languages:
L_1 = {<M> | M takes at least 2016 steps on some input};
L_2 = {<M> | M takes at least 2016 steps on all inputs} and
L_3 = {<M> | M accepts ε};
where for each Turing machine M, <M> denotes a specific encoding of M. Which one of the following is **TRUE**? **[2016, Set 2, 2 Marks]**
(a) L_1 is recursive and L_2, L_3 are not recursive
(b) L_2 is recursive and L_1, L_3 are not recursive
(c) L_1, L_2 are recursive and L_3 is not recursive
(d) L_1, L_2, L_3 are recursive

124. Consider the following statements.
I. The complement of every Turing decidable language is Turing decidable
II. There exists some language which is in NP but is not Turing decidable
III. If L is a language in NP, L is Turing decidable

Which of the above statements is/are true? **[2015, Set 2, 1 Mark]**
(a) Only II (b) Only III
(c) Only I and II (d) Only I and III

125. Let < M > be the encoding of a Turing machine as a string over $\Sigma = \{0, 1\}$. Let L = {<M>| *M is a Turing machine that accepts a string of length* 2014}. Then, L is **[2014, Set-2, 2 Marks]**
(a) decidable and recursively enumerable
(b) undecidable but recursively enumerable
(c) undecidable and not recursively enumerable
(d) decidable but not recursively enumerable

126. Which of the following statements is/are false?
1. For every non-deterministic Turing machine, there exists an equivalent deterministic Turing machine.
2. Turing recognisable languages are closed under union and complementation.
3. Turing decidable languages are closed under intersection and complementation.
4. Turing recognisable languages are closed under union and intersection. **[2013, 1 Mark]**

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

127. 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? **[2005, 2 Marks]**
(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

128. A single tape turing maching M has two states q_0 and q_1, of which q_0 is the starting state. The tape alphabet of M is {0, 1, B} and its input alphabet is {0, 1}. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

	0	1	B
q_0	q_1, 1, R	q_1, 1, R	Halt
q_1	q_1, 1, R	q_0, 1, L	q_0, B, L

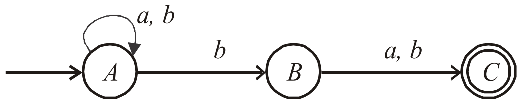
The table is interpreted as illustrated below.
The entry (q_1, 1, R) in row q_0 and column 1 signifies that if M is in state q_0 and reads 1 on the current tape square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q_1
Which of the following statements is true about M? **[2003, 2 Marks]**
(a) M does not halt on any string in $(0 + 1)^+$
(b) M does not halt on any string in $(00 + 1)^*$
(c) M halts on all strings ending in a 0
(d) M halts on all strings ending in a 1

129. Consider the following problem X:
Given a turing machine M over the input alphabet Σ, any state q of M and a word $W \in \Sigma^*$, does the computation of M on W visit the state q?
Which of the following statements about X is correct? **[2001, 2 Marks]**
(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

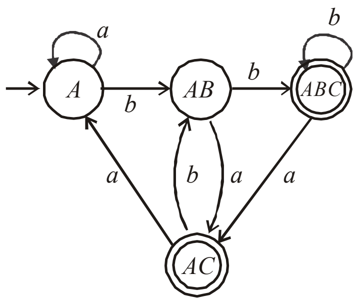
Hints & Solutions

Regular Expression and Finite Automata

1. (4) The given regular expression is $(a + b)^* b (a + b)$.
In this all string whose second last bit is 'b'.
So, the minimal NFA is

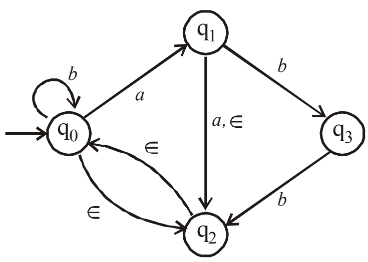


It can be described as "All string over $\{a, b\}$ ending with 'ba' or 'bb'. Then the minimal (DFA) accepting (L), find by converting it into minimal DFA by subset construction Algorithm.



Hence, it is a minimal DFA with 4 states.

2. (c) Converting the table to a state diagram then we get.



then, $\hat{\delta}(q_2, aba) =$ All states reachable from q_2 by aba.

If aba is broken as $\in, a, \in, \in, b, a.$ then from q_2 it can reach q_1 and from there by null transaction to reach state q_2 as well as q_0.

Consider, $(q_2, a) = (q_2, \in a) = \{q_1, q_2 q_0\}$.

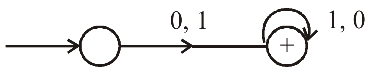
$$(q_2, ab) = (q_1, b) = \{q_0, q_3, q_2\}$$
$$(q_2, b) = \{q_0, q_2\}$$
$$(q_0, b) = \{q_0, q_2\}.$$
$$(q_2, aba) = (q_0, a) = \{q_1, q_2, q_0\}$$
$$(q_1, a) = \{q_2, q_0\}$$
$$(q_2, a) = \{q_1, q_2, q_0\}$$
$$(q_3, a) = \{q_3\}$$

Then, $\hat{\delta}(q_2, aba) = \{q_1, q_2, q_0\}$
$$= \{q_0, q_1, q_2\}$$

Hence option (c) is correct.

3. (b) Option (a) contains 00 and 11 consecutively which does not fulfill the required condition. Option (c) does not give assurance that both 00 and 11 will be exist in the string.
According to option (d), string should start with 11 and ends with 00 or vice versa. Hence option (b) is the correct answer.

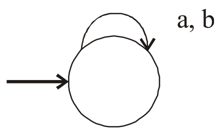
4. **2**



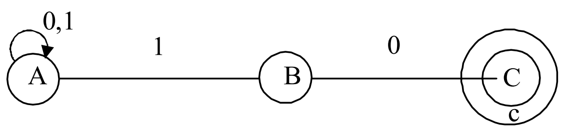
So, the minimum number of states is 2.

5. (b) Statement I is not true, because if all the states of DFA are accepting states then the language accepted by the DFA is $\sum^*$
Statement II is true because one can have regular language A = [ ] [Empty Language] which satisfies the given condition.

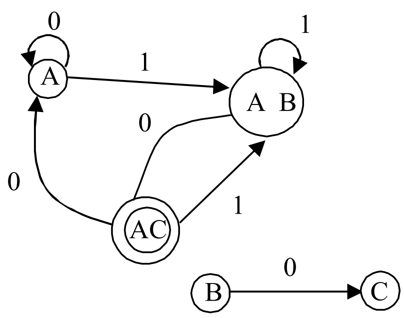
6. **1** M accepts the strings which end with a and N accepts the strings which end with b. Their intersection should accept empty language.



7. **3** (0 +1)*10



	0	1
A	A	AB
B	C	–
C	–	–
AB	AC	AB
AC	A	AB



K total minimum DFA states = 3

8. (c)
9. (b)
10. (a) Following paths can be taken by the finite Automata for the input string "0011":—

$$q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0$$

Reachable states

$\{q_0\}$

$$q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_1$$

$\{q_0, q_1\}$

$$q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2$$

(q_0, q_1, q_2)

We note that no other path is possible for the input string "0011".

So, finally union of all three cases gives us the set of Reachable states which is $\{q_0, q_1, q_2\}$

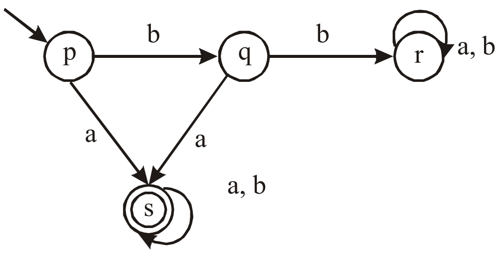
11. (b) (I) $0*1(1+00*1)*$
(II) $0*1*1+11*0*1$
(III) $(0+1)*1$
(I) and (III) represent DFA.
(II) Doesn't represent as the DFA accepts strings like 11011, but the given regular expression doesn't accept.

12. (a) L_1 is regular but L_2 is not

13. (c) Σ^* is countably finite
$2\Sigma^*$ is power set of Σ^*
The powerset of countabily infinite set is uncountable
$\therefore$ $2\Sigma^*$ is uncountable and Σ^* is countable.

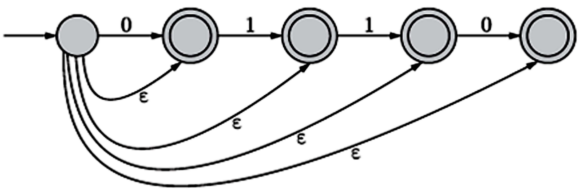
14. **3** a * b * (ba) * a *
Length O is present (as it accepts $\in$)
Length 1 is present (a, b)
Length 2 is present (aa, ab, ba, bb)
Length 3 is not present (bab not present)
$\therefore$ it is 3

15. (a) As state (s) and (t) both are final states and accepting a* + b*, we can combine both states and we will get



16. (b) As n is constant atleast n + 1 states will required to design a^{nk}.

17. (c) *L* is the set of all substrings of *w* where *w* ! {0,1})
Any string in *L* would have length 0 to *n*, with any no. of 1's and 0's
The *NDFA*

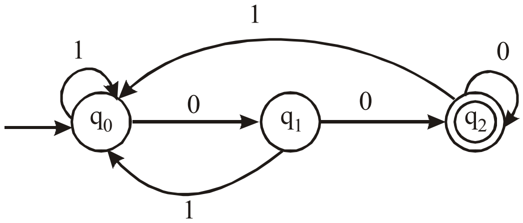


Here $n = 4$
So to accept all the substrings the no. of states required are $n + 1 = 4 + 1 = 5$
Hence (c) is correct option.

18. (b) It is given that L is the set of all bit strings with even number of 1's so the regular expression should exhibit the same.
Now, the min string should be ε and the string should be ε, 0, 11, 101, ...
The string obtained from such expression is 0* (10* 10 *)*

19. (b) Given grammar $S \to aSabSbab$.
The strings generated through this grammar is definitely palindromes,but not all it can only generate palindromes of odd length only so (A) & (D) are false, (B) is correct.
Also it can generate palindromes which start and end with same symbol, but not all strings eg. *aabababba*.

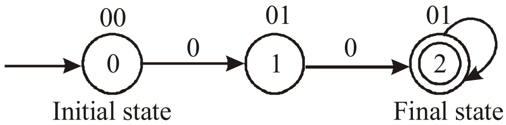
20. (c)



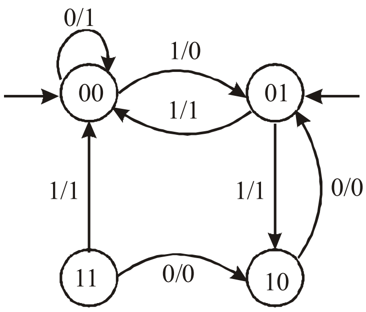
State	Input	Output
q_0 to q_0	1, 0	00
q_1 to q_0	1, 0	00
q_2 to q_0	1, 0	00

Therefore, the above DFA ends with 00.

21. (a) The initial state = 00
Final state required = 01
Let us construct the transition diagram for the given.



We get the total number of states to be 3 when getting the desired output. The transition diagram can further be elaborated as below.
There can be 4 states 00, 01, 10, 11.
With this, the FSM can be designed as



The desired output is obtained with the input string 101, however, the concern is number of states which we found to be 3.

22. (c) We are given with the relation
$(0+1)*0(0+1)*0(0+1)*$
Here, the accepting languages are
L= {00, 000, 100, 001, 010, 0000, 0001, 1000, 1001, 0100, 1100, 0010, 0011, 0110, 0101, 1010, ...}
The common feature in the accepting languages can be seen that they consist of atleast two 0's.

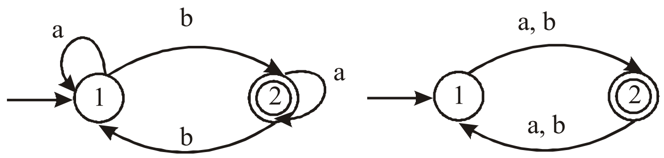
23. (a) Lets gather some knowledge for the regular expressions before going to True or False. The regular sets are defined in such a way that they have to follow some conventions. Conventions on regular expressions
1. Bold face is not used for regular expressions when the expression is not confusing. So, for example, (r + s) is used instead (r + s).

2. The operation * has precedence over concatenation, which further has precedence over union (+). Thus, the regular expression (a + b(c*))) is written as a + bc*.
3. The concatenation of k r's, where r is regular expression, is written as r^k. Thus, for example rr = r^2. The language corresponding to rk is L_r^k, where L_r is language corresponding to the regular expression r. For a recursive definition of L_r^k.
4. The (r^+) is used as a regular expression to represent L_r^+.

   Based on the concept, only 1 and 4 are found to be regular since in 1, L can be written as a* (bb)* and 4, (a + b) * c(a + b)* can be written for the 4.

24. (d) Option (a) True: Non-deterministic finite automata can be converted into the deterministic finite automata.
Option (b) True: With reference to option (a), same is with the non-deterministic turing machine.
Option (c) True: Regular language is always context-free but the reverse is not true.
Option (d) False: We know that a set is a subset of itself and hence, every subset of recursively enumerable set is not recursive.

25. (a) **Given:** Transition table for Y and Z
The number of states of Z = 2
The number of states of Y = 2
Z × Y



Y Z

No. of states of the product of Z and Y = 2 × 2 = 4
Now, the states as per the given options are P, Q, R and S. The finite state automata is

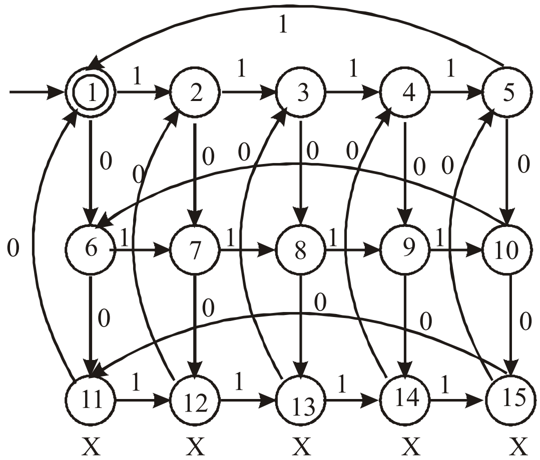
	a	b
→ P	S	R
Q	R	S
R(F)	Q	P
S	Q	P

Table for Z × Y is

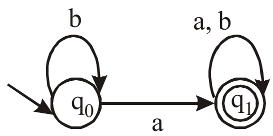
	a	b
→ (1, 2)	(2, 1)	(2, 2)
(1, 2)	(2, 2)	(2, 1)
(2, 1)	(1, 1)	(1, 2)
F(2, 2)	(1, 2)	(1, 1)

26. (b) Finite subset of non-regular set is regular as we know that Pumping Lemma can be applied on all the finite sets that states that all finite sets are regular. Infinite union of finite set is not regular because regular sets can never be closed under infinite union.

27. (a) It is given that the 0's and 1's are divisible by 3 and 5 and we know that 3 and 5 do not have any factor other than themselves or 1 i.e., these cannot be further breakdown.
Therefore, number of states = 3 × 5 = 15
The schematic representation is as follows:



28. (b) The FSA as obtained in the previous question is b* a (a + b)*
The minimum number of states are thus given by



q_0 and q_1 are the state: that are required at most and hence the minimum number of states is 2 (q_0 and q_1).

29. (b) The behaviour as per the given PDA is an seen below
q_0 to q_1 → b*a
q_1 to q_2 → (a + b) *
Regular expression = b * a (a + b)*

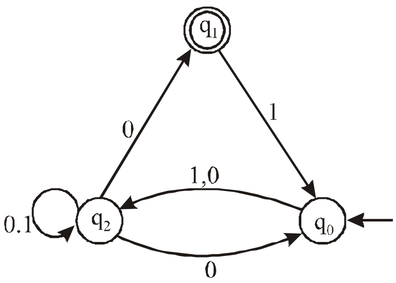
30. (a) The given finite state machine accepts any string W ∈(0, 1)* in which the number of 1's is multiple of 3 and the number of 0's is multiple of 2.

31. (d) Option (a) solves to (1*0*)1*
Option (b) solves to 0 + (0*1*0*)
Option (c) solves to 0*1*10(0*1*)
Therefore, none of the statement has the output equivalent to the given.

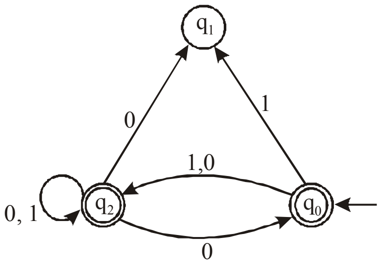
32. (c) The strings accepted by the given automata are of type. Option
1 2 3 4 5 6 7
1– – 1 – – 1
The four blank spaces can have a probability of having 0 or 1, so total 2(pow,4)= 16 strings are possible, but the given automata does not accept all of those.
1. 1 1 1 1 0 0 1
2. 1 1 0 1 0 0 1
3. 1 0 1 1 0 0 1
4. 1 0 0 1 0 0 1

5. 1 0 0 1 0 0 1
6. 1 0 0 1 1 0 1
7. 1 0 0 1 1 1 1
Hence (c) is correct option

33. (b) The given machine M is



Now, the complementary machine $\overline{M}$ is



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

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

Clearly, $\lambda = L(\overline{M})$, since q_0 is accepting it (0 + 1) (0 + 1)* $\in L(\overline{M})$, since q_2 is accepting it.

$\therefore$ $L(\overline{M}) = L_1 = \lambda + (0 + 1)\ (0 + 1)^*$

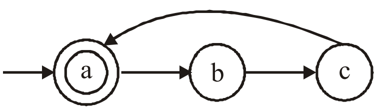
$\therefore$ $L_1 = (0 + 1)^* = \{0, 1\}^*$

34. (a) As per the given diagram.

State	Input	Output
A to A	10	00
B to A	10	00
C to A	10	00

Therefore, the finite state machine outputs the sum of the present and previous bits of the input.

35. (b) Answer in the book is wrong

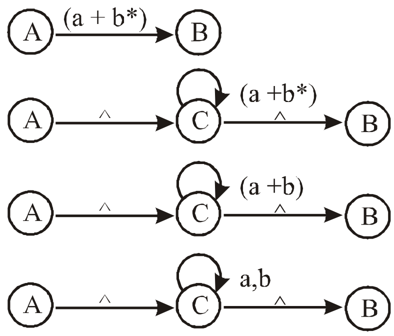


Start & end are same as (a) hence the minimum no. of states required are3. Option (b) is correct. If string traversal doesn't stop at (a) then string length is not divisibleby 3.

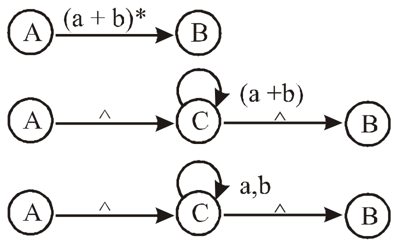
36. (b) In DFA any subset of the N states (for N element set N subsets possible) can become a new state and they can remain even when the DFA is minimized. So, maximum we can get N states for the minimized DFA.

37. (d) We construct a DFA for strings divisible by 6. It requires minimum 6 states as length of string mod 6 = 0, 1, 2, 3, 4, 5
We construct a DFA for strings divisible by 8. It requires minimum 8 states as length of string mod 8 = 0, 1, 2, 3, 4, 5, 6, 7
If first DFA is minimum and second DFA is also minimum then after merging both DFAs resultant DFA will also be minimum. Such DFA is called as compound automata.
So, minimum states in the resultant DFA = 6 * 8 = 48
Thus, option (D) is the answer.

38. (c) Lets consider the regular expressions as given in the questions:



Second, (a + b)*



$\Rightarrow$ (a + b*)* = (a + b*)
Therefore, we can say that S = T

Context Free Grammars & Pushdown Automata

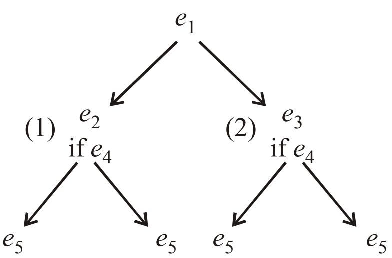
39. (c) The given context free Grammar
$\Sigma = \{a, b, c\}$ with S as the start symbol.
$S \to abScT / abcT$
$T \to bT / b$
The minimum length string generated by the grammar is 1.
Consider first case $(S \to abcT)$ then $(S \to abcb)$.
Hence all the variable s are greater than 1.
Consider second case
$S \to ab\boldsymbol{S}cT$
$\to ababScTcT$
$\to ababScTcTcT$
....
....
$\to (ab)^n (CT)^n$
Here T can generate any number of b's string with single b, so $(T = b^m)$
Hence, the language is
$L = [(ab)^n (cb^m)^n \mid m, n \geq 1]$
Hence option (c) is correct.

40. (c) Follow (Q) ?
Follow (Q) is First (R) Hence First (R) = $\{W, E\}$.
We add 'W' in Follow (Q) and for $\in$.
We calculate :
First (S) = $\{y\}$
So, follow (Q) = First (R S)
$= \{\{W\} - \in\} \cup$ First (S)
Follow (Q) = $\{W, y\}$ $(\therefore$ First $(S) = \{y\})$
Hence option (c) is correct.

41. (d) The given grammar with productions,
$S \to SaS \mid aSb \mid bSa \mid SS \mid \in$
Now consider,
$S \to aSb \mid bSa \mid SS \mid \in$
This grammar generates all strings with equal number of 'a' and 'b'.
Now, $S \to SaS$ can only generate strings where 'a' is more than 'b'. Since on left and right of 'a' in SaS, S will have only strings with $n_a = n_b$ or $n_a > n_b$.
Now consider each options
1. $S \to SS \to aSbS \to abS \to abaSb \to abab$, when, $S \to \in$
2. $S \to aSb \to aSaSb \to aaaSb \to aaab$, when $S \to \in$
3. $S \to SS \to aSbS \to abS \to abbSa \to abbSaSa \to abbaa$, when $(S \to \in)$
4. "$babba$" which is a string with $nb > na$ is not possible to generate by the given grammar.
Hence, option (d) is correct.

42. (b) The context free grammar given over alphabets $\Sigma = \{a, b, c\}$, with S and T are nonterminals.
Given, $G_1 : S \to aSb \mid T,\ T \to cT \mid \in$
$G_2 : S \to bSa \mid T,\ T \to cT \mid \in$
Lets $L(G_1)$ is the language for Grammar (G_1) and $L(G_2)$ is the language for Grammar (G_2).
where $L(G_1) = \{a^n c^m b^n \mid m, n \geq 0\}$
$L(G_2) = \{b^n c^m a^n \mid m, n \geq 0\}$
then $L(G_1) \cap L(G_2) = \{a^n c^m b^n\} \cap \{b^n c^m a^n\}$
$= \{c^m \mid m \geq 0\} = C^*$
Since the only common strings will be those strings with only 'C', so the intersection is not finite but regular.

43. (1024) The number of control Paths is depends on number of if statement that is 2^n, where, n is the number. of if statements
For (2 "if statements") $2^2 = 4$ control flow Paths are possible:



These four control flow are :
(1) $e_1 \to e_2$
(2) $e_1 \to e_3$
(3) (e_2 if e_4) $\to e_5$
(4) (e_3 if e_4) $\to e_5$
So, for (10 "if statements"), $2^{10} = 1024$.
Control flow path are possible. Hence, answer is **1024**.

44. (c) $E \to E - T \mid T$
$T \to T + F \mid F$
$F \to (E) \mid id$
Using the rule for removal of left recursion is
$A \to A\alpha / \beta$
$A \to \beta A'$
$A' \to \alpha A' / \in$
Then, the given grammar is written as :-
$E' \to -TE' / \in$
$E \to +TE'$
$T' \to +FT' / \in$
$T \to FT'$
$F \to (E) \mid id$
Now by putting E' as X and T' as Y, then
$X \to -TX / \in$
$E \to TX$
$Y \to +FY / \in$
$T \to FY$
$F \to (E) \mid id$
Hence option (c) is correct.

45. (c) There is no known algorithm to verify whether the language accepted by TM is empty or not. Similarly there is no algorithm to verify whether language of CFG's are equivalent.

46. (d) In G1, there will be atleast 1 b because S->B and B->b. But no. of A's can be 0 as well and no. of A and B are independent. In G2, either we can take S->aA or S->bB. So it must have atleast 1 a or 1 b. So option (d) is correct.

47. (d) Initial state is also the final state.
So a^n is also accepted.

48. (c) L1 has the property that no. of a's should be equal to no. of b's in a string, and all a's should precede all b's. Hence extra memory will be required to check this property of a string (Finite Automata can't be built for this type of language). Hence this is not regular language. Therefore P is False. L2 has the property that no. of a's should be equal to no. of b's, but order of a's and b's is different here, it is (ab)*, which will require no extra memory to be accepted. (Finite Automata can be built for this language). Hence L2 is regular language. Therefore Q is True.

49. (b) Grammar A has direct left recursion because of the production rule: A→Aa. Grammar C has indirect left recursion because of the production rules:S→Aa and A→Sc Grammar D has indirect left recursion because of production rules : A→Bd and B→Ae Grammar B doesn't have any left recursion (neither direct nor indirect).

50. (a) Both G_1 and G_2 can correctly generate the declarations.

51. (d)

52. (c) Option (a) is false when CFG contains cycle
Option (b) is false as CFG can contain cycle
Option (d) is false as a single node can contain block of statements.

53. (d) 1. G is a CFG. Is L(G) = ϕ?
Decidable
2. G is a CFG. Is L(G) = Σ * ?
Undecidable
3. M is a Turing Machine. Is L (M) regular?
Undecidable.
4. A is a DFA and N is an NFA. Is L(A) = L(N)?
Decidable.
Hence, the correct answer is (d) 2 and 3 only.

54. (b) DPDA and NPDA because an NPDA cannot be converted into DPDA.

55. (d) (a) true, since minimal *DFA* for every regular language is possible.
(b) true, *NFA* can be converted into an equivalent *PDA*.
(c) *CG'S* are not recursive but their complements are.
(d) false, since non deterministic *PDA* represents, non deterministic. *CFG*, since *NDCFG* and *CFG* are proper subsets so conversion required.

56. (c) Statement 1 is **true**: Using GNF we can convert Left recursive grammar to right recursive and by using reversal of CFG and GNF we can convert right recursive to left recursive.
Statement 2 is **false**: because if ∈ is in the language then we can't remove ∈ production from Start symbol. (For example L = a*)
Statement 3 is **true** because right linear grammar generates regular set
Statement 4 is **true**, only two non-terminals are there in each production in CNF. So it always form a binary tree.

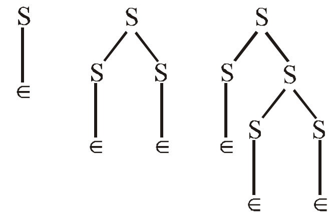
57. (b) Finite state automata (*FSA*) has no undecidability *CFL* membership problem is also decidable.
So option (b) i.e Ambiguity of *CFL* cannot be decidable.

58. (b) Due to S → *S* this Grammar is ambiguous right hand side has twoNon terminals. Also the strings like *aaabbb* have equal no. of *a's* & *b's* but can't be produced by this grammar. So 2 is false.
Statement 3 is true since it is a *CFG* so accepted by *PDA*.

59. (d) We know that, the languages accepted by non-deterministic finite automata are also accepted by deterministic finite automata. This may not be in the case of context-free languages.
Therefore, $D_f = N_f$ and $D_p \subset N_p$

60. (c) Here, we have
S → bS
S → baA (S → aA)
S → baaB (A → aB)
S → baaa (B → a)
Therefore, | Na (w) | = 3.
Also, if we use A → bA instead of A → aB,
S → baA
S → babA
To terminate A, we would have to use A → aB as only B terminates at a (B → a).
S → baA
S → babA
S → babaB
S → babaa
Thus, here also, | Na (w) | = 3.
So, C is the correct answer.

61. (c) The grammar is S→ aSb|SS|∈
(a) G is not ambiguous is false, since ∈ which belongs to L(G), has infinite number of derivation trees, which makes G ambiguous. Some derivation trees are



(b) There exists x, y ∈ L(G) such that xy ∉L(G) is false, since S→SS, can be used derive xy, whenever x ∈L(G) and y∈L(G).
(c) It is true, since this language is L(G) = {W|n_a(W) = n_b (W) and n_a(v) ≥ n_b (V) where V is any prefix of W}
This language happens to be deterministic context-free language.
∴ There exists a dpda that accepts it.
(d) It is false, as the given language is not regular.
∴ number of DFA exists to accept it.

62. (a) In P_1, A given finite state machine accepts a given string if it is decidable because in decidable problem there exists a turing machine that gives the correct answer for every statement in the domain of the problem. A class of problems with two outputs yes or no is said to be decidable (solvable), if there exist some definite algorithm which always terminates (halts) with one of the two outputs yes or no.
Similarly, in P_2, the context-free grammar generates.

Regular and Context Free Language

63. (a) The given language over Alphabets $\Sigma = (a, b, c)$ and the given language are :-
$L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$ is a CFL
$L_2 = \{a^m b^n c^n \mid n, m \geq 0\}$ is also a CFL
then union of two CFL is always CFL,
$L_1 \cup L_2 = \{a^n b^m c^k \mid (n = m) \text{ or } (m = k), n, m \geq 0\}$ is a context free language.
Intersection of two CFL is may or may not be a context free language.
$L_1 \cap L_2 = \{a^n b^m c^k \mid (n = m) \text{ or } (m = k), n, m \geq 0\}$ or $\{a^n b^n c^n \mid n \geq 0\}$ is a non context free language.
Hence, option (a) is correct.

64. (b)

1. $L_1 \cup L_2$ is context-free because the union of two context free language is always a context free or CFL are closed under union operation . So, it is correct.

2. $\overline{L_1}$ is not context-free because CFL are not closed under complement operation. So, it is incorrect.

3. $L_1 – R$ is context-free because if context-free language is intersection to the complement of regular language then it is always context-free. $L_1 - R = L_1 \cap \overline{R_1}$, so it is correct.

4. $L_1 \cap L_2$ is not context-free because context-free languages are not closed under complement operations. So it is incorrect.

65. (c) The given grammar with S as start symbol is
$S \to XY$

$X \to ax|a \Rightarrow X \to (a^m \mid m \geq 1)$

$Y \to ayb|E \Rightarrow Y \to (a^n b^n \mid n \geq 0)$

$S \to XY$

$S \to \{a^m b^n \mid m > n,\ n \geq 0\}$

because, from Non terminal X we can generate any number of a's including a single 'a' and from Y equal number of a's and b's.

Hence $L = \{a^m b^n \mid m > n,\ n \geq 0\}$

$m > n$, because at least one will be attached on left of $a^m b^n$.

So, option (c) is correct.

66. (8) The Minimum possible number of states of deterministic finite Automata that accepts the regular language.

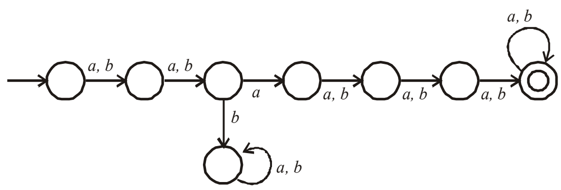
As given that,

$L = \{W_1 a W_2 \mid W_1, W_2 \in \{a,b\}^*, |W_1| = 2, |W_2| \geq 3\}$ is

The regular expression for L is

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

The minimal DFA accepting L is :



Hence, the minimum number of state are **8**.

67. (d) Consider the given languages.

$L_1 = \{a^P | P \text{ is a Prime Number}\}$,
$L_2 = \{a^n b^m c^{2m} \mid n \geq 0,\ m \geq 0\}$
$L_3 = \{a^n b^n c^{2n} \mid n \geq 0\}$, $L_4 = \{a^n b^n \mid n \geq 1\}$.

Now, consider each option:

(i) L_1 is context free but not regular is **Incorrect** because it is not CFL, it is a CSL (Context sensitive language).

(ii) L_2 is not CFL is also **Incorrect** because it is CFL.

(iii) L_3 is not context free but recursive is **correct** because L_3 is CSL (context sensitive language) and every CSL is recursive.

(iv) L_4 is deterministic context free is **correct** because L_4 is a proper subset of context free language.

Hence option (d) is correct.

68. (d)

69. (c)

70. (d) Statement (IV) : $\overline{L_2}$ may or may not be context free because CFL are not closed under complementation. So it is not true.

71. (b) L_2 is not context free. No. of b's will match with no. of a's leaving c's to be matched with no one. So L_2 cannot be context free.

72. (d) **I** $\Rightarrow \overline{L_1}$ is recursive

This one is true, because L_1 is context free which is nonentity but recursive, recursive language is closed under complement. Hence it is true.

II $\Rightarrow \overline{L_2}$ (complement of L_2) is recursive

If L_2 and L_2 both are recursive enumerable then L_2 is recursive

Hence, statement II is false

III $\Rightarrow \overline{L_1}$ is context free

Which is false because context free language does not closed under complement.

IV $\Rightarrow \overline{L_1}$ L_2 is recursive enumerable.

$\overline{L_1}$ recursive, because every recursive language is also recursive enumerable.

$\overline{L_2}$ recursive enumerable.

$\overline{L_1} \cup L_2 \Rightarrow$ recursive enumerable, because recursive enumerable language closed under union.

73. (a) $L_1 = \{wxw^R | w, x \in \{a, b\}^* \text{ and } |w|, |x| > 0\}$

w^R is reverse of string w.

It is regular

Regular expression

a (a + b)* a + b (a + b)* b

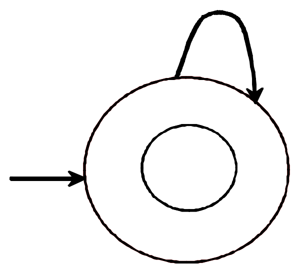
$L_2 = \{a^n b^m | m \neq n, m, n \geq 0\}$

Hence m ≠ n, that mean n is greater than m, or m is greater than n.

So we need memory, so we can't draw DfA for it.

$L_3 = \{a^p b^q c^r | p, q, r \geq 0\}$

a, b, c



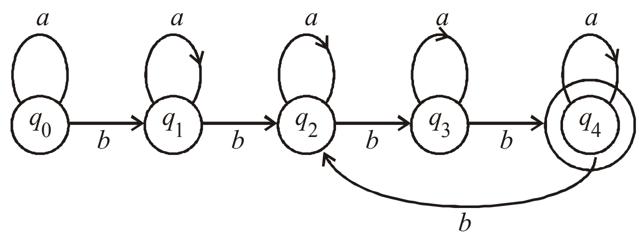
74. (c) $L_1 = a^m b^n a^n b^m \Rightarrow$ This one is CFL

$L_2 = a^m b^n a^m b^n \Rightarrow$ by pumping lemma this one is not CFL.

$L_3 = a^m b^n \, |m = 2n + 1 \Rightarrow$ This is CFL.

75. (c) We know that a language L is regular if an equivalent finite Automata can be constructed for it. DFA can be constructed for L as follows:–

Note that L contains the strings that has $3k + 1$ number of b's (that is 4, 7, 10, ...n ...... no. of b's and any no. of a's)



q_0 is the initial state. q_4 is the final state.
FA state remains same when input symbol is "a" at any point of time.
When 1st 'b' is read, state is changed from q_0 to q_1.
When 2nd 'b' is read, state is changed from q_1 to q_2.
When 3rd 'b' is read, state is changed from q_2 to q_3.
When 4th 'b' is read, state is changed from q_3 to q_4.
If no more 'b' is encountered, the string is accepted when last input symbol is read.
However if 5th "b" is there is string, state is changed from q_1 to q_2 so that to accept the string 2 more bs must be there in the string at least.
Continuing in this way only those strings are accepted that has $3k + 1$ $(k \in N)$ number of b's. (i.e. 4, 7, 10 ... number of b's).

76. (d) If B is not recursively enumerable then A need not be recursively enumerable.

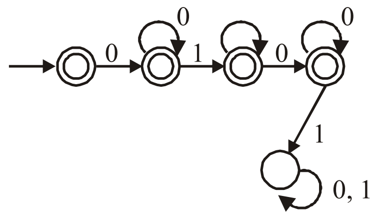
77. (c) Both L and $\overline{L}$ are recursively enumerable but not recursive.
Set of recursive languages is subset of the set of recursively enumerable languages.
So, if a language is recursive, It must be R.E. also.
(a) May be true as a language L and its complement $\overline{L}$ need not be recursively enumerable.
(b) May be true if L is r.e. but not recursive and $\overline{L}$ is not recursively enumerable
(d) May be true as L and $\overline{L}$ both can be recursive.
However (c) is not possible because if Both L and $\overline{L}$ are recursively enumerable than by a well known theorem of complexity theory either L or $\overline{L}$ has to be recursive.

78. (a)

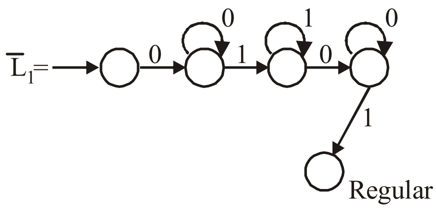
79. (c) L1 and L2 have deterministic push down automata but for L3 only not-deterministic PDA is possible. So L3 is not deterministic.

80. (d) $L_1 = \{0^p1^q,0^r \mid p, q, r \geq 0\}$
$L_2 = \{0^p1^q0^r \mid p, q, r \geq 0, p \neq r\}$
(a) L_2 is context Free – true
We can accept, or reject L_2 with single stack.
Insert P 0's into stack skip q 1's.
For each 0 corresponding to r, remove 0 from stack.
(b) $L_1 \cap L_2$ is context Free – true
Here $L_1 \cap L_2 = L_2$ which is context Free.
(c) Complement of L_2 is recursive – true
L_2 is Context-Free language
Complement of CFL may or may not be CFL.
Complement by CFL is definately recursive.
(d) Complement of L_1 is Context Free, but not regular false.

$$L_1 = \{0^p q^q 0^r \big|_{p,q,r\geq 0}\}$$



Which is regular,



Hence, the answer is (d).

81. (a) Considering the languages $L_1 = \phi$ and $L_2 = \{a\}$
For all languages L it is known that $\phi.\ L = \phi$
Suppose, $\exists\, a$ string $s \in \phi.L$
$\exists\, s'$ such that $s = s'\ .\ s''$
and $s' \in \phi s'' \in L$
But $s' \in \phi$ $\qquad\qquad \because \phi$ is an empty language.
$\therefore \quad \phi\ .\ L = \phi$
$\{\varepsilon\} \subseteq L^*$ for all languages L.
As it means taking letters from the language and concatenating them '0' times creating a '0' length string which is possible for all languages.
$\therefore \quad \varepsilon$ is in ϕ^*
$L_1\ .\ L_2 = \phi$
and $L_1 = \{\varepsilon\}$
$L_1\ .\ L_2\ ^* \cup L^*_1 = (L_1\ .\ L_2) \cup L^*_1$
$\therefore \quad \phi \cup \{\varepsilon\} = \{\varepsilon\}$

82. (c) As we have given,
A(n) $\rightarrow$ Average case complexity
W(n) $\rightarrow$ Worst case complexity
As we know that average case will always be less than or equal to worst case complexity.

$$A(n) \leq W(n)$$

Consider option (a):
$A(n) = \Omega\ (W(n))$
$\therefore$ It says W(n), worst case complexity is less than the average case complexity which is wrong because Ω asymptotic notation shows ($\geq$) sign. Hence, this option is false.
Consider option (b)
$A(n) = \theta\ (W(n))$
$\therefore$ It says A(n), worst case complexity is same as worst case complexity because θ notation shows equality sign. Hence, this is false.
Consider option (c)
$A(n) = O\ (W(n))$
$\therefore$ It says that average case complexity A(n) is less than or equal to W(n) worst case complexity. Hence, this option is correct.
Consider option (d)
$A(n) = O\ (W(n))$
$\therefore$ It says that A(n) is strictly less than W (n), so this option is false.

83. (c) $L_1 = \{0^p 1^q \mid p, q \in N\}$ is regular language

$L_2 = \{0^p 1^q \mid p, q \in N \text{ and } p = q\}$ is context-free language

$L_3 = \{0^p 1^q 0^r \mid p, q, r \in N \text{ and } p = q = r\}$ is not context-free.

84. (c) $\sum^* - P$ and $\sum^* - P$ is the complement of P and complement of regular language is also regular.

85. (b) $L_1 \rightarrow$ recursive
$L_2, L_3 \rightarrow$ recursively enumerable but not recursive.
So L_1 can be recursive enumerable.
$RE - RE = RE$
So $L_1 - L_3$ is recursively enumerable.

86. (d) These sort of languages are accepted by *PDA*, so all should be context free languages. L_2 & L_3 are definitely *CFL* since accepted by stockof *PDA*.
And also L_1 & L_4 are linear comparisons of i & j so can also be represented using *PDA*. So all are context free languages.

87. (c) It is given that $L = L_1 \cap L_2$
Let's first analyze the given language and check whether it is context-free or not. The context-free languages are defined as below–

A context-free language is $L = \{a^n b^n : n \geq 1\}$ that is the language of all non-empty even-length strings. It consists of
1. the entire first halves of which are a's.
2. the entire second halves of which are b's.

L is generated by the grammar $S \rightarrow aSb \mid ab$, and is accepted by the push-down automata $M = (\{q_0, q_1, q_f\}, \{a, b\}, \{a, z\}, \delta\, q_0, \{q_f\})$ where δ is defined as
$\delta(q_0, a, z) = (q_0, a)$
$\delta(q_0, a, a) = (q_0, a)$
$\delta(q_0, b, a) = (q_1, x)$
$\delta(q_1, b, a) = (q_1, x)$
$\delta(q_1, \lambda, z) = (q_f, z)$
δ (state$_1$, read pop) = (state$_2$, push)
where z is initial stack symbol and x means pop action.
Here, in we are given, where a and b are of equal lengths and followed by c which is of different length. This definitely shows that the languages are context-free but not regular.

88. (d) LL is recursively enumerable means a TMTM accepts all strings in LL. $L^{-}L^{-}$ is recursively enumerable means a TMTM accepts all strings in $L^{-}L^{-}$. So, we can always decide if a string is in LL or not, making LL recursive.

If a language L and its complement $\bar{L}$ are both recursively enumerable, then both languages are recursive. If L is recursive, then $\bar{L}$ is also recursive, and consequently both are recursively enumerable.

Proof: If L and $\bar{L}$ are both recursively enumerable, then there exist. Turing machines M and $\widehat{M}$ that serve as enumeration procedures for L and $\bar{L}$, respectively. The first will produce w_1, w_2, ... in L, the second $\hat{w}_1, \hat{w}_2$,.... in $\bar{L}$. Suppose now we are given any $w \in \Sigma^+$. We first let M generate w_1 and compare it with w. If they are not the same, we let $\widehat{M}$ generate $\widehat{w_1}$ and compare again. If we need to continue, we next let M generate w_2, then $\widehat{M}$ generate $\hat{w}_2$, and so on.

Any $w \in \Sigma^+$ will be generated by either M or $\widehat{M}$, so eventually we will get a match. If the matching string is produced by M, w belongs to L, otherwise it is in $\hat{L}$. The process is a membership algorithm for both L and $\bar{L}$. The process is a membership algorithm for both L and $\bar{L}$, so they are both recursive.

For the converse, assume that L is recursive. Then there exists a membership algorithm for it. But this becomes a membership algorithm for $\bar{L}$ by simply complementing its conclusion. Therefore, $\bar{L}$ is recursive. Since any recursive language is recursively enumerable, the proof is completed.

89. (b) **Statement 1 Decidable:** The algorithm can be used to check the finiteness/infiniteness on the DFA and also the two given DFAs, a product DFA can be constructed.
Statement 2 Undecidable: It is not decidable since the language is context-free.
Statement 3 Undecidable: It is also undecidable that whether two push-down automata accept the same language.
Statement 4 Decidable: If the LHS of each production has one and only one variable then it is a context-free grammar.

90. (d) $\{a^P P$is a prime no.$\}$
This prime no. is extra constraint so this language is neither *LFG* nor *RG* but it can be accepted by turing machine.

91. (b) $L = \{0^i 2l^i \mid i > 0\}$, this language can't be accepted by *DFA* to regular, but it is recursive & can be accepted by *PDA* to *CFL*.

92. (c) Lets study the regular language.
Conventions on regular expressions
1. Bold face is not used for regular expressions when the expression is not confusing. So, for example, (r + s) is used instead (r + s).
2. The operation * has precedence over concatentation, which further has precedence over union (+). Thus, the regular expression (a + b(c*))) is written as a + bc*.
3. The concatenation of k r's, where r is regular expression, is written as r^k. Thus, for example rr = r^2. The language corresponding to r^k is L_r^k, where L_r is language corresponding to the regular expression r. For a recursive definition of L_r^k.

4. The (r^+) is used as a regular expression to represent L_r^+.

Since, language L can be expressed as
$r = [0(0+1)*0]+[1(0+1)*1]$
and follows the above convention, therefore is regular

93. (b) L_1 is regular language
L_2 is *CFL*.
L_3 is recursively enumerable but not *REC*.
(a) $L_1 \cap L_2$ is *CFL* is true
(b) $L_3 \cap L_1$ is recursive, not necessary so false.
(c) & (d) are also true.

94. (d) Given that, d(s) mod 5 = 2 and d(s) mod 7 ¹ 4}
Both the languages are regular languages, since there exists deterministic finite automata for both of them. We also have an algorithm to check the regular nature of the language therefore L is regular.

95. (c) Option (a), (b) & (d) can be accepted by *DFA*, & there is no linear relationship between the no. of 0'*s* &1'*s* in the string but in (c)$n_0(S)$ - $n_1(S)$ <=4 can't be accepted by *DFA*, we require a *PDA*. Hence it is not regular.

96. (d) The language is context free or not, this can be proved by finding out the grammar for all the languages.

$$L_1 = \{0^{n+m}1^n0^m \mid n, m \geq 0\}$$

The production for L_1 as follows:

$$S \to 0S_0 | 0A_1 | \varepsilon$$

$$A \to 0A_1 | \varepsilon$$

Now, we need to apply these values of the production individually to generate

$$0^{n+m} \Big| \, {}^n 0^m$$

that is,
$0^m0^n1^n0^m \to$ To prove
Applying $0S_0$ (m times)
$S \to 0\,S_0$, $S \to 0^mS_0^{\,m}$
Applying, $S \to 0A_1$
$\to S \to 0^m\,0A_1\,0^m$
Here applying, $A \to 0\,A_1$, n – 1 times
$S \to 0^m\,0\,0^{n-1}\,A_1^{\,n-1}\,10^m$
$\to S \to 0^m\,0^n1^n0^m$
Hence, proved, so L_1 is context-free.
Going through the same procedure, we can see that two comparisons are made in the L_2 language, So it is not context-free.
$L_3 \to$ Going through the same procedure again, we can see that two comparisons are made in the L_3 language, so it is not context free.

97. (b) In all the options there is linear relationship among strings so all *CFL*'*s* , but L_1 & L_3 can be accepted by *PDA*, L_2 can be accepted by deterministic *CFL* due to presence of special symbol D which tells the middle of the string, so deterministic.

98. (b) The rules here used will be.
All those languages which are recursive their complements are also recursive.
So option (a) & (b) can be correct.
Now languages which are recursively enumerable but not recursive, their complements can't be recursively enumerable.
So only option (b) is correct.
Hence (b) is correct option.

99. (b) a is followed by two or more than 2b's so the language recognized by M is $\{W \in \{a, b\}^* \mid$.

100. (a) S_1 is TRUE.
If L_1 is recursive L_2 must also be recursive. Because to check. If a word w = w_i # w_j belong to L_2, we can give w_iw_j to the decider for L_1 and if both are accepted then w belong to L_1 and not otherwise.
S_2 is TRUE.
With a decider for L_2 we can make a decider for L_1 as follows. Let w_1 be the first string enumerated by algorithm A for L_1. Now, to check if a word w belongs to L_1, make a string w′ = w_1 # w and give it to the decider for L_2 and if accepted, then w belongs to L_1 and not otherwise.
So, answer must be A.

101. (b) Language is not regular bcoz we need to match count of c's is equal to count of b's + count of a's and that can implement by PDA

$\partial(q_0,a,\wedge) = (q_0,a)$	[push a in stack, as per language a comes first]
$\partial(q_0,a,a) = (q_0,aa)$	[push all a's into stack]
$\partial(q_0,b,a) = (q_1,ba)$	[push b in stack, state change to q_1 that sure b comes after a]
$\partial(q_1,b,b) = (q_1,bb)$	[push all b's in stack]
$\partial(q_1,c,b) = (q_2,\wedge)$	[pop one b for one c]
$\partial(q_2,c,b) = (q_2,c)$	[pop one b's for each c and continue same]
$\partial(q_2,c,a) = (q_3,\wedge)$	[pop one a for one c , when there is no more b in stack]
$\partial(q_3,c,a) = (q_3,\wedge)$	[pop one a for each c and continue same]
$\partial(q_3,\wedge,\wedge) = (q_f,\wedge)$	[if sum of c's is sum of a's and b's then stack will be empty when there is no c in input]

Hence, language is context- free but not regular.

102. (d) The strings of a language L can be effectively enumerated means a Turing machine exists for language L which will enumerate all valid strings of the language.
If the string is in lexicographic order then TM will accept the string and halt in the final state.
But, if the string is not lexicographic order then TM will reject the string and halt in non-final state.
Thus, L is recursive language.
We cannot construct PDA for language L. So, the given language is not context free. Thus, option (D) is correct.

103. (c) Give F and F inverse are polynomial time computable functions.
$\Rightarrow$ we can reduce L_1 to L_2 in polynomial time and L_2 to L_1 in polynomial time.

if L_1 is Un-decidable then L_2 should also be undecidable .
if L_2 is decidable then L_1 should also be decidable.
It is possible to convert L_1 to L_2 and L_2 to L_1 , if both are decidable are , both are un-decidable.

104. (a) If $L_0 \cup L_1$ is recursively enumerable, it means we can find out for all $W \in \Sigma^*$, where M halts or does not halt. This means that if $L_0 \cup L_1$ is recursively enumerable, the halting problem would be decidable. But we know, the halting problem is undecidable. Therefore $L = L_0 \cup L_1$ is not RE.

Since, $\overline{L} = (L_0 \cup L_1)^c = L_0^c \cap L_1^c = \phi$ which is regular language and hence is RE.

Therefore, $\overline{L}$ is RE.

So, correct option is (b), which is $\overline{L}$ is RE but L is not.

105. (b) It is given that the stack is limited to 10 items therefore the stack function is a type of recursion. This directly implies that the language accepted by push-down automata is a regular language.

106. (a) Option (a) True : The complement of recursive language is always recursive.
Option (b) False : The recursively enumerable language is the language, when taken its complement, lose its recursively enumerable nature.
Option (c) False : The complement of recursive language is always recursive and never recursively enumerable.
Option (d) False : The complement of context free language is never context-free.

107. (b) C language is CSL
- In C the grammar is context free.But the language is context sensitive e.g. I can't declare a variable twice.
- Most of the programming languages like C, C++, Java etc. can be well approximate by CFG, and the compilers are made taking into account CFGs. However, C language itself contains context sensitive properties which cannot be deal with CFG.

108. (a) L_1 and L_2 are context-free languages and therefore $L_1 \cap L_2$ may or may not be context-free, since CFLs are not closed under intersection. Now, let us look at $L_1 \cap L_2$.

$L_1 \cap L_2 = \{a^n b^n c^n \mid n > 0\}$

which is clearly not context-free but is context sensitive.

109. (b) Option (a) False: If the language is context-free then it cannot be said that the language will be accepted by the deterministic push-down automaton as there are some cases defined where the context-free language is not accepted.
Option (b) True: The union of context-free languages always produces a context-free language but it not the case with intersection of two context-free languages.
Option (c) False: As discussed in Option (b).
Option (d) False: Similar to intersection, complement of a context-free language is not always a context-free language.

110. (d) Lets study the regular language first and then will choose which language is regular.
Conventions on regular expression.
1. Bold face is not used for regular expressions when the expression is not confusing. So, for example, (r + s) is used instead of (r + s).
2. The operation * has precedence over concatenation, which further has precedence over union (+). Thus, the regular expression (a + b(c*))) is written as a + bc*.
3. The concatenation of k r's, where r is a regular expression, is written as r^k. Thus, for example $rr = r^2$. The language corresponding to r^k is L_r^k, where L_r is the language corresponding to the regular expression r. For a recursive definition of L_r^k.
4. The (r^+) is used as a regular expression to represent L_r^+.

$L_3 = \{0^{2i} | i \text{ is an integer}\}$ follows all the conventions of the regular language and therefore, is regular.

111. (a) Lets consider both the statements separately to find the correct option.
$S_1 : \{0^{2n} | n \geq 1|\}$
Applying the values of n,
$\rightarrow S_1 = 00, 0000, 000000, ...$
The behaviour shown by the output is regular and hence, the language is a regular language.
$S_2 : \{0^m 1^n 0^{m+n} | \; m \geq 1 \text{ and } n \geq 1|\}$
Applying the values of *m* and *n*,
$S_2 = 0100, 00110000, 000111000000, ...$
Here the values of *m* and *n* are kept same so thay are showing the output in symmetry but if we use the different values of m and n then the output will display a behaviour which is not regular. Therefore, confirmed is that S_1 is a regular language.

112. (b) $S \rightarrow 0S_0/00$
$S \rightarrow 0S_0$ (type 2 production)
This is a context-free grammar as in CFG, the left hand side of the production rule has no left and right context.
$S \rightarrow 00$
means $S \rightarrow$ ^ (null)* (type 3 production)
This is called the regular grammar.

113. (d) Lets take an example to solve this.
$L_1 = \{n, nnn, nnnnn, ...\}$
$= \{n^{ood}\}$
$= \{n^{2n+1}; \text{ for } n = 0, 1, 2, 3, 4, 5, 6,...\}$
and $L_2 = \{n, nnnn, nnnnnn, ...\}$
$= \{n^{even}\}$
$= \{n^{2n}; \text{ for } n = 0, 1, 2, 3, 4, 5, 6 ...\}$
Therefore, either L must be $\{a^n | n \text{ is odd}\}$, or L must be $\{a^n \; | n \text{ is even}\}$

Pumping Lemma and NP Completeness

114. (c) The matched sequence is as given

[redacted]

115. (a) There is a difference between $SHAM_3$ and $DHAM_3$ that $SHAM_3$ is used for finding a Hamiltonian cycle in a graph but $DHAM_3$ is the problem of determining if a graph exists in a Hamiltonian cycle. Also it is given that |V| is divisible by 3, hence the problem can be reduced from 3 SAT which further determines that $SHAM_3$ and $DHAM_3$ are NP hard.

116. (b) (A) Incorrect because R is not in NP. A NP Complete problem has to be in both NP and NP-hard.
(B) Correct because a NP Complete problem S is polynomial time educable to R.
(C) Incorrect because Q is not in NP.
(D) Incorrect because there is no NP-complete problem that is polynomial time Turing-reducible to Q.

117. (a)

118. (c) 3 SAT $\to$ NP-complete problem
2 SAT $\to$ P type problem

119. (c) Since, 3 SAT is reduced to $\prod$, this implies 3 SAT $\le \prod$ and since, $\prod$ is reduced to 3 SAT, this implies $\pi \le$ 3 SAT.
We know that, if $L_1 \le L_2$ and L_1 is NP-complete, then L_2 is also NP-complete.
Since, 3 SAT is NP-complete so $\prod$ is also NP complete.

120. (a) Consider the case where L = (0 + 1)*
Here the answer of any turing machine would be yes since some output is displayed.
Consider the case where L = ϕ
Here, the answer of any turing machine would be No since no output is displayed.
This type of behaviour is displayed by the recursive language and therefore, the language L is recursive language and therefore, the language L is recursive.

Turing Machines

121. (a) A Turing Machine (M) is a recursive if F it halts for every input string either in accept or reject state. Here a computable function is defined in simple way such as $\{X \# f(X) \mid X \in A^*\}$
So, f is computable if and only if L_f is recursive.
So, option (a) is correct.

122. (d) Consider each options :
(a) statement is membership problem of regular language and it is decidable for finite state machine and regular expression.
(b) statement is emptyness problem of CFL emptyness problem is decidable for CFG by checking usefulness of start symbol.
(c) statement is a problem of CFL, and it is undecidable problem, we can not check whether $L(G) = \Sigma^*$ or not but rather we can check complement of $L(G)$ is ϕ.
(d) statement is a membership problem of a Turing Machine and it is undecidable problem for turning machine.
So, options (c) and (d) are undecidable.
Hence option (d) is correct.

123. (c)

124. (d) Turing decidable $\Rightarrow$ Recursive language
Turing recognizable $\Rightarrow$ Recursive enumerable language
(I) Complement of turning decidable language is decidable which is true.
(III) True (Theorem)
Which violates (II). Hence correct option is (d).

125. (b) The language is recursive enumerable and it is undecidable

126. (c) 1. Non-deterministic Turing Machine can be simulated by a deterministic Turing Machine with exponential time true.
2. Turing recongnizable language are "not" closed under complementation. For any Turing recognizable language the Turing Machine 'T' recognizing 'L' may not terminate on inputs $x \notin L$ - False
3. Turing decidable languages are CLOSED under union and complementation. It is easy to determine if turing machine is decidable-*True*
So, answer is option (c) only 2.

127. (c) According to the option, P_2 is reducible to P_3.
Also it gives that P_2 is undecidable.
From (a) and (b) ;
P_3 is undecidable.

128. (a) This turning machine starts at 90 if it doesn't get any input symbol but B then it stops.
So if (00 + 1) * is chosen then the M/C can halt.
Option (b) is wrong.
Option (c) & (d) are possible but not necessary.
Option (a) (0 + 1) * 1 or more occurrence of 0 or 1.
So 0, 1, 00, 01, 10, 11........are valid strings & the machine doesn't halt for them.
Hence (a) is correct option

129. (b) Option (d) is incorrect as the problem is a decision problem. Now, we need to check whether the problem is decidable or not.
The problem is not decidable since the word $\omega \in \Sigma$.
Also, the problem is not completely undecidable since, the input and the state is given so after a certain point of time the probability of finding out the answer increases.