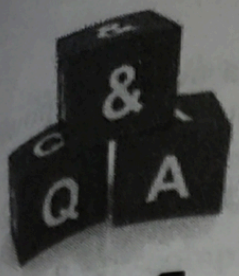

11 Automata Theory

Instructions

A. The following abbreviations are used in this chapter.

FSM	–	Finite State Machine
DFSM	–	Deterministic Finite State Machine
NDFSM	–	Non-Deterministic Finite State Machine
PDM	–	Push Down Machine
DPDM	–	Deterministic Push Down Machine
NDPDM	–	Non-Deterministic Push Down Machine
TM	–	Turing Machine
UTM	–	Universal Turing Machine
CFG	–	Context Free Grammar
CF	–	Context Free
CFL	–	Context Free Language
CSG	–	Context Sensitive Grammar

B. In Transition diagrams, states are represented by circles.
The start state is represented by a circle pointed to by an arrow.
A final state is represented by a circle encircled by another.

C. In a CFG, unless stated otherwise, grammar symbol on the left hand side of the first production, is the start symbol.

*1. Which of the following pairs have DIFFERENT expressive power? (GATE 2011)

(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

*2. The lexical analysis for a modern computer language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense? (GATE 2011)

(a) Finite state automata
(b) Deterministic pushdown automata
(c) Non-deterministic pushdown automata
(d) Turing machine

*3. 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 $\{p^n q^n \mid n \in \mathrm{N}\}$). Then which of the following is ALWAYS regular? (GATE 2011)

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

*4. Consider the languages L_1, L_2 and L_3 as given below. (GATE 2011)

$L_1 = \{0^p 1^q \mid p, q \in \mathrm{N}\}$,
$L_2 = \{0^p 1^q \mid p, q \in \mathrm{N} \text{ and } p = q\}$ and
$L_3 = \{0^p 1^q 0^r \mid p, q, r \in \mathrm{N} \text{ and } p = q = r\}$.

Which of the following statements is NOT TRUE?

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

*5. Which of the following problems is undecidable? (GATE 2007)

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

*6. The recognizing capability of NDFSM and DFSM

(a) is different (b) sometimes different
(c) is the same (d) none of these

7. FSM can recognize

(a) any grammar (b) only CFG
(c) any unambiguous grammar (d) only regular grammar

8. Pumping lemma is generally used for proving

(a) a given grammar is regular
(b) a given grammar is not regular
(c) whether two given regular expressions are equivalent
(d) none of the above

*9. Which of the following are not regular?

(a) Strings of 0's whose length is a perfect square.
(b) Set of all palindromes made up of 0's and 1's.
(c) Strings of 0's, whose length is a prime number.
(d) Strings of odd number of zeroes.

*10. Which of the following pairs of regular expressions are equivalent?
(a) `1(01)*` and `(10)*1`
(b) `x(xx)*` and `(xx)*x`
(c) `(ab)*` and `a*b*`
(d) `x*` and `x*x*`

*11. Assuming P ≠ NP, which of the following is **TRUE**? (GATE 2012)
(a) NP-complete = NP
(b) NP-complete ∩ P = ∅
(c) NP-hard = NP
(d) P = NP-complete

*12. Pick the correct statements.
The logic of Pumping lemma is a good example of
(a) the pigeon-hole principle
(b) the divide and conquer technique
(c) recursion
(d) iteration

*13. The basic limitation of an FSM is that
(a) it does not have the capability to remember information
(b) it sometimes recognizes grammars that are not regular
(c) it sometimes fails to recognize grammars that are regular
(d) all of the above

14. Palindromes can't be recognized by any FSM because
(a) an FSM does not have the capability to remember information
(b) an FSM can't deterministically fix the mid-point
(c) even if the mid-point is known, an FSM can't find whether the second half of the string matches the first half
(d) none of the above

15. An FSM can be considered a TM
(a) of finite tape length, rewinding capability and unidirectional tape movement
(b) of finite tape length, without rewinding capability and unidirectional tape movement
(c) of finite tape length, without rewinding capability and bidirectional tape movement
(d) of finite tape length, rewinding capability and bidirectional tape movement

16. TM is more powerful than FSM because
(a) the tape movement is confined to one direction
(b) it has no finite state control
(c) it has the capability to remember arbitrary long sequences of input symbols.
(d) none of the above

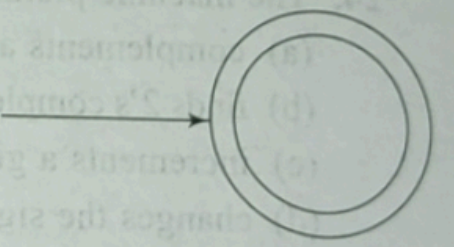

Fig. 11.1

*17. The FSM pictured in Fig. 11.1 recognizes
(a) all strings
(b) no string
(c) ε – alone
(d) none of the above

18. The FSM pictured in Fig. 11.2 is a
(a) Mealy machine
(b) Moore machine
(c) Kleene machine
(d) none of the above

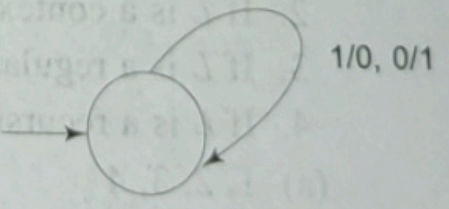

Fig. 11.2

19. The machine pictured in Fig. 11.2
(a) complements a given bit pattern
(b) generates all strings of 0's and 1's
(c) adds 1 to a given bit pattern
(d) none of the above

20. The language of all words (made up of a's and b's) with at least two a's can be described by the regular expression
(a) (a+b)*a(a+b)*a(a+b)*
(b) (a+b)*ab*a(a+b)*
(c) b*ab*a(a+b)*
(d) a(a+b)*a(a+b)*(a+b)*

21. Which of the following pairs of regular expression are not equivalent?
(a) (ab)*a and a(ba)*
(b) (a+b)* and (a*+b)*
(c) (a*+b)* and (a+b)*
(d) none of the above

*22. What is the complement of the language accepted by the NFA shown below? Assume $\Sigma = \{a\}$ and ε is the empty string. (GATE 2012)

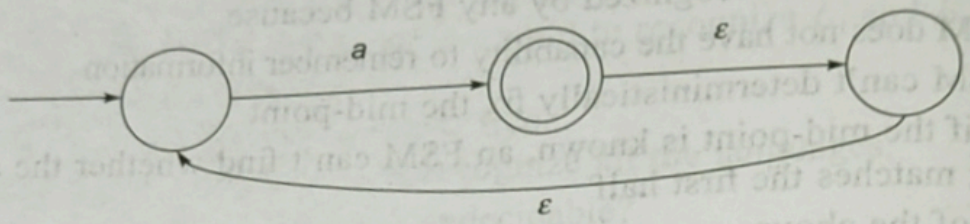

Fig. 11.3

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

23. Set of regular languages over a given alphabet set, is not closed under
(a) union (b) complementation
(c) intersection (d) none of the above

*24. The machine pictured in Fig. 11.4.
(a) complements a given bit pattern
(b) finds 2's complement of a given bit pattern
(c) increments a given bit pattern by 1
(d) changes the sign bit

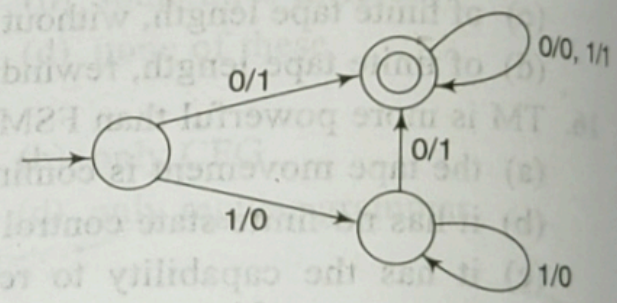

Fig. 11.4

*25. Which of the following problems are decidable? (GATE 2012)
1. Does a given program ever produce an output?
2. If L is a context-free language, then, is $\overline{L}$ also context-free?
3. If L is a regular language, then is $\overline{L}$ also regular?
4. If L is a recursive language, then is $\overline{L}$ also recursive?
(a) 1, 2, 3, 4 (b) 1, 2 (c) 2, 3, 4 (d) 3, 4

*26. The FSM pictured in Fig. 11.5 recognizes

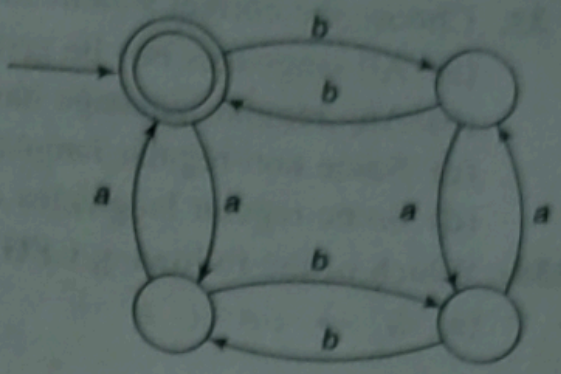

Fig. 11.5

(a) any string of odd number of a's
(b) any string of odd number of a's and even number of b's
(c) any string of even number of a's and even number of b's
(d) any string of even number of a's and odd number of b's

27. Any given Transition graph has an equivalent
(a) regular expression (b) DFSM
(c) NDFSM (d) none of the above

28. The following CFG

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

is equivalent to the regular expression
(a) (a*+b)*
(b) (a+b)*
(c) (a+b)(a+b)*
(d) (a+b)*(a+b)

*29. Any string of terminals that can be generated by the following CFG

S → XY
X → aX | bX | a
Y → Ya | Yb | a

(a) has at least one b
(b) should end in an 'a'
(c) has no consecutive a's or b's
(d) has at least two a's

*30. The following CFG

S → aB | bA
A → a | aS | bAA
B → b | bS | aBB

generates strings of terminals that have
(a) equal number of a's and b's
(b) odd number of a's and odd number b's
(c) even number of a's and even number of b's
(d) odd number a's and even number of b's

31. Let L(G) denote the language generated by the grammar G. To prove set $A = L(G)$,
(a) it is enough to prove that an arbitrary member of A can be generated by grammar G
(b) it is enough to prove that an arbitrary string generated by G, belongs to set A
(c) both the above comments (a) and (b) are to be proved
(d) either of the above comments (a) or (b) is to be proved

*32. The set $\{a^n b^n \mid n = 1,2,3...\}$ can be generated by the CFG
(a) S → ab | aSb
(b) S → aaSbb | ab
(c) S → ab | aSb | ε
(d) S → aaSbb | ab | aabb

33. Choose the correct statements.
(a) All languages can be generated by CFG.
(b) Any regular language has an equivalent CFG.
(c) Some non-regular languages can't be generated by any CFG.
(d) Some regular languages can't be generated by any CFG.

*34. Which of the following CFG's can't be simulated by an FSM?
(a) S → Sa | a
(b) S → abX
X → cY
Y → d | aX
(c) S → aSb | ab
(d) None of the above

35. CFG is not closed under
(a) union (b) Kleene star (c) complementation (d) product

36. The set $A = \{a^n b^n a^n \mid n=1,2,3,\ldots\}$ is an example of a grammar that is
(a) regular (b) context free
(c) not context free (d) none of the above

37. Let $L1 = \{a^n b^n a^m \mid m, n = 1,2,3,\ldots\}$
$L2 = \{a^n b^n a^n \mid m, n = 1,2,3,\ldots\}$
$L3 = \{a^n b^n a^n \mid n = 1,2,3,\ldots\}$
Choose the correct statements.
(a) $L3 = L1 \cap L2$
(b) L1 and L2 are CFL but L3 is not a CFL
(c) L1 and L2 are not CFL but L3 is a CFL
(d) L1 is a subset of L3

38. Choose the correct answers.
$L = \{a^n b^n a^n \mid n=1,2,3,\ldots\}$ is an example of a language that is
(a) context free
(b) not context free
(c) not context free but whose complement is context free
(d) context free but whose complement is not context free

39. The intersection of a context free language and a regular language
(a) need not be regular (b) need not be context free
(c) is always regular (d) is always context free

*40. Given the language $L = \{ab, aa, baa\}$, which of the following strings are in L^*?
(GATE 2012)
1. *abaabaaabaa*
2. *aaaabaaaa*
3. *baaaaabaaaab*
4. *baaaaabaa*

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

41. A PDM behaves like a TM when the number of auxiliary memory it has is
(a) 0
(b) 1 or more
(c) 2 or more
(d) none of the above

*42. Consider the set of strings on {0, 1} in which, *every substring of 3 symbols* has at most *two* zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below. (GATE 2012)

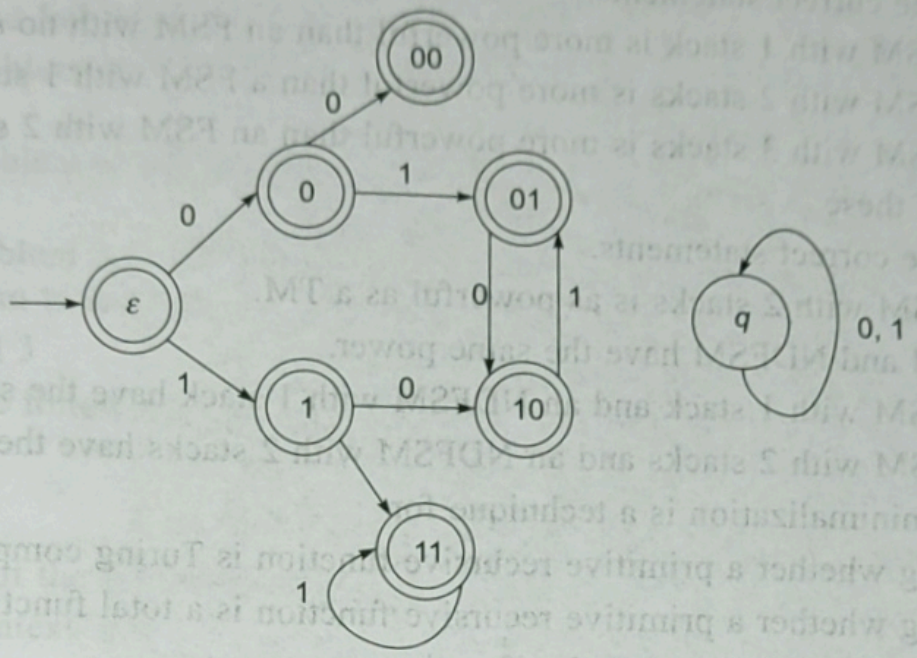

Fig. 11.6

The missing arcs in the DFA are

(a)

	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

(b)

	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

(c)

	00	01	10	11	q
00	1	1			0
01		1			
10			0		
11		0			

(d)

	00	01	10	11	q
00		1			0
01				1	
10	0				
11			0		

43. Which of the following is accepted by an NDPDM, but not by a DPDM?
(a) All strings in which a given symbol is present at least twice.
(b) Even palindromes (i.e., palindromes made up of even number of terminals).
(c) Strings ending with a particular terminal.
(d) None of the above

44. CSG can be recognized by a
(a) FSM
(b) DPDM
(c) NDPDM
(d) linearly bounded memory machine

45. Choose the correct statements.
(a) An FSM with 1 stack is more powerful than an FSM with no stack.
(b) An FSM with 2 stacks is more powerful than a FSM with 1 stack.
(c) An FSM with 3 stacks is more powerful than an FSM with 2 stacks.
(d) All of these.

46. Choose the correct statements.
(a) An FSM with 2 stacks is as powerful as a TM.
(b) DFSM and NDFSM have the same power.
(c) A DFSM with 1 stack and an NDFSM with 1 stack have the same power.
(d) A DFSM with 2 stacks and an NDFSM with 2 stacks have the same power.

47. Bounded minimalization is a technique for
(a) proving whether a primitive recursive function is Turing computable
(b) proving whether a primitive recursive function is a total function
(c) generating primitive recursive functions
(d) generating partial recursive functions

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

*49. Which of the following statements is/are **FALSE?** (GATE 2013)
1. For every non-deterministic Turning machine, there exists an equivalent deterministic Turing machine.
2. Turing recognizable languages are closed under union and complementation.
3. Turing decidable languages are closed under intersection and complementation.
4. Turing recognizable languages are closed under union and intersection.
(a) 1 and 4 only (b) 1 and 3 only (c) 2 only (d) 3 only

50. Choose the correct statements.
(a) A total recursive function is also a partial recursive function.
(b) A partial recursive function is also a total recursive function.
(c) A partial recursive function is also a primitive recursive function.
(d) A primitive recursive function is also a partial recursive function.

51. A language L for which there exists a TM, T, that accepts every word in L and either rejects or loops for every word that is not in L, is said to be
(a) recursive (b) recursively enumerable
(c) NP-HARD (d) none of the above

52. Choose the correct statements.
(a) $L = \{a^n b^n a^n \mid n = 1, 2, 3, \ldots\}$ is recursively enumerable.
(b) Recursive languages are closed under union.
(c) Every recursive language is recursively enumerable.
(d) Recursive languages are closed under intersection.

*53. Which of the following statements are **TRUE**?
1. The problem of determining wether there exists a cycle in an undirected graph is in P.
2. The problem of determining whether there exists a cycle in an undirected graph is in NP.
3. If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.

(GATE 2013)
(a) 1, 2 and 3 (b) 1 and 2 only (c) 2 and 3 only (d) 1 and 3 only

*54. Consider the following languages.

$$L_1 = \{0^p\ 1^q\ 0^r \mid p, q, r \geq 0\}$$
$$L_2 = \{0^p\ 1^q\ 0^r \mid p, q, r \geq 0, p \neq r\}$$

(GATE 2013)
Which one of the following statements is **FALSE**?
(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 mot regular.

*55. Which of the following is **TRUE**?
(a) Every subset of a regular set is regular. (GATE 2007)
(b) Every finite subset of a non-regular set is regular.
(c) The union of two non-regular sets is not regular.
(d) Infinite union of finite sets is regular.

*56. The language $L = \{0^i 21^i \mid i \geq 0\}$ over the alphabet $\{0, 1, 2\}$ is (GATE 2007)
(a) not recursive (b) is recursive and is a deterministic CFL
(c) is a regular language (d) is not a deterministic CFL but a CFL

*57. Which of the following is true for the language $\{a^p \mid p \text{ is a prime}\}$? (GATE 2008)
(a) It is not accepted by a Turing 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.

*58. Consider the DFA A given below.

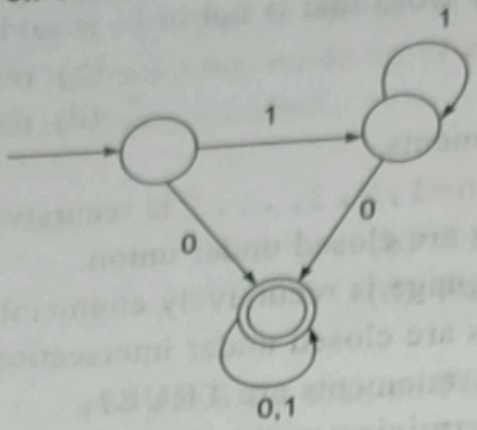

Fig. 11.7

Which of the following are **FALSE?**

1. Complement of $L(A)$ is context-free.
2. $L(A) = L((11^*0 + 0)(0 + 1)^*0^*1^*)$
3. For the language accepted by A, A is the minimal DFA.
4. A accepts all strings over $\{0, 1\}$ of length at least 2.

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

59. If there exists a TM which when applied to any problem in the class, terminates if the correct answer is yes, and, may or may not terminate otherwise is said to be

(a) stable (b) unsolvable (c) partially solvable (d) unstable

*60. If L and $\bar{L}$ are recursively enumerable then L is (GATE 2008)

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

*61. Let w be any string of length n in $\{0, 1\}^*$. Let L be the set of all substrings of w. What is the minimum number of states in a non-deterministic finite automaton that accepts L? (GATE 2010)

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

*62. Let P, Q, and R be three languages. If P and R are regular and if $PQ = R$, then

(a) Q has to be regular (b) Q cannot be regular
(c) Q need not be regular (d) Q has to be a CFL

63. Consider the grammar

S → PQ | SQ | PS
P → x
Q → y

To get a string of n terminals, the number of productions to be used is

(a) n^2 (b) $n + 1$ (c) $2n$ (d) $2n - 1$

*64. Choose the correct statements.
A class of languages that is closed under
(a) union and complementation has to be closed under intersection
(b) intersection and complementation has to be closed under union
(c) union and intersection has to be closed under complementation
(d) all of the above

65. The following grammar is

$$S \rightarrow a\alpha b \mid b\alpha c \mid aB$$
$$S \rightarrow aS \mid b$$
$$S \rightarrow \alpha bb \mid ab$$
$$b\alpha \rightarrow bdb \mid b$$

(a) context free (b) regular
(c) context sensitive (d) LR (k)

*66. Which of the following definitions generates the same language as L, where $L = \{x^n y^n,\ n \geq 1\}$?
I. $E \rightarrow xEy \mid xy$
II. $xy \mid x^+xyy^+$
III. x^+y^+
(a) I only (b) I and II (c) II and III (d) II only

*67. A finite state machine with the following state table has a single input x and a single output z.

Present state	Next state, z	
	$x = 1$	$x = 0$
A	D, 0	B, 0
B	B, 1	C, 1
C	B, 0	D, 1
D	B, 1	C, 0

If the initial state is unknown, then the shortest input sequence to reach the final state C is
(a) 01 (b) 10 (c) 101 (d) 110

*68. Let $A = \{0, 1\}$ and $L = A^*$. Let $R = \{0^n1^n,\ n>0\}$. The languages $L \cup R$ and R are respectively
(a) regular, regular (b) not regular, regular
(c) regular, not regular (c) not regular, not regular

*69. Consider the languages $L1 = \{0^i1^j \mid i \neq j\}$, $L2 = \{0^i1^j \mid i = j\}$, $L3 = \{0^i1^j \mid i = 2j+1\}$, $L4 = \{0^i1^j \mid i \neq 2j\}$. Which one of the following statements is true? (GATE 2010)
(a) Only $L2$ is context free. (b) Only $L2$ and $L3$ are context free.
(c) Only $L1$ and $L2$ are context free. (d) All are context free.

*70. 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 1s. Which one of the regular expressions below represents L? (GATE 2010)

(a) $(0^*10^*1)^*$ (b) $0^*(10^*10^*)^*$ (c) $0^*(10^*1)^*0^*$ (d) $0^*(00^*1)^*0^*$

*71. A CFG is said to be in Chomsky Normal Form (CNF), if all the productions are of the form $A \rightarrow BC$ or $A \rightarrow a$. Let G be a CFG in CNF. To derive a string of terminals of length x, the number of productions to be used is

(a) $2x - 1$ (b) $2x$ (c) $2x + 1$ (d) 2^x

*72. $S \rightarrow aSa \mid bSb \mid a \mid b$

The language generated by the above grammar over the alphabet $\{a,b\}$ is the set of (GATE 2009)

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

*73. Let π_A be a problem that belongs to the class NP. Then which one of the following is TRUE? (GATE 2009)

(a) There is no polynomial time algorithm for π_A
(b) If π_A can be solved deterministically in polynomial time, then $P = NP$.
(c) If π_A is NP-hard, then it is NP-complete.
(d) π_A may be undecidable.

*74. Which one of the following languages over the alphabet $\{0,1\}$ is described by the regular expression: (GATE 2009)

$(0 + 1)^*0(0 + 1)^*0(0 + 1)^*$?

(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.

*75. Let $L = L_1 \cap L_2$, 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 (GATE 2009)

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

*76.

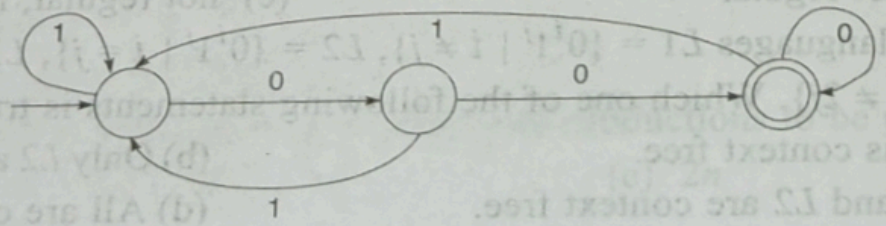

Fig. 11.8

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

*77. Let L1 be a recursive language. Let L2 and L3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true? (GATE 2010)

(a) L2 – L1 is recursively enumerable
(b) L1 – L3 is recursively enumerable
(c) L2 ∩ L3 is recursively enumerable
(d) L2 ∪ L3 is recursively enumerable

ANSWERS

1. b	2. a	3. c	4. c
5. b	6. c	7. d	8. b
9. a, b, c	10. a, b, d	11. b	12. a
13. a	14. a, b, c	15. b	16. c
17. c	18. a	19. a	20. a, b, c
21. d	22. b	23. d	24. c
25. d	26. c	27. a, b, c	28. b, c, d
29. d	30. a	31. c	32. a, d
33. b, c	34. c	35. c	36. c
37. a, b	38. b, c	39. c, d	40. c
41. c	42. d	43. b	44. d
45. a, b	46. a, b, d	47. c	48. c
49. c	50. a, d	51. b	52. a, b, c, d
53. a	54. d	55. b	56. b
57. d	58. d	59. c	60. d
61. c	62. c	63. d	64. a, b
65. c	66. a	67. b	68. c
69. d	70. b	71. a	72. b
73. b	74. c	75. c	76. c
77. b			

EXPLANATIONS

1. DFA and NFA have the same power. NPDA is more powerful than DPDA.
2. The primary function of a lexical analyzer is to tokenize the incoming stream of characters. In modern languages, an identifier has to be declared before use, and keywords are reserved. These constraints facilitate the use of finite state automata to implement lexical analysis.
3. Regular languages are closed under complementation. Σ^* is the universal set. $\Sigma^* - P$ represents the complement of P.
4. L_1 is regular. Any regular language is also context free (and hence recognizable by a PDA).

   L_2 can be generated by the CFG,

   $S \rightarrow 0S1 \mid 01$

   So, L_2 is context free (and hence recognizable by PDA).

   Is L_3 context free?

   If it is context free, there will be a PDA that accepts all strings of the form $0^p1^p0^p$ (and accepts only strings of this form). However to recognize such strings, an automaton needs two stacks. Since a PDA has only one stack, there cannot exist a PDA. Hence L_3 is not context free. Pumping lemma can also be used to show that L_3 is not context free.
5. Some of the other famous undecidable problems from automata theory are the following.
   - Determining if two given CFG's generate the same set of strings
   - For two given CFG's, determining if the set of strings generated by one is a subset of the set of strings generated by the other
   - For two given CFG's, determining if the set of strings generated by them are disjoint
6. DFSM is a special case of NDFSM. Corresponding to any given NDFSM, one can construct an equivalent DFSM. A DFSM can be called an NDFSM. So they are equally powerful.
9. Strings of odd number of zeroes can be generated by the regular expression `(00)*0`. Pumping lemma can be used to prove the non-regularity of the other options.
10. Two regular expressions `R1` and `R2` are equivalent if any string that can be generated by `R1` can be generated by `R2` and vice-versa. In option (c), `(ab)*` will generate `abab`, which is not of the form a^nb^n (because a's and b's should come together). All other options are correct (check it out!).
11. Is P = NP is still an open question. However, if the assumption is that P ≠ NP, then NP-complete and P should be disjoint. This is because if they are not disjoint, they will have at least one in common. This means there is at least one NP-complete problem solvable in polynomial time. This contradicts our assumption that P ≠ NP because one NP-complete problem can reduced (transformed) to another.
12. Pigeon-hole principle is that if 'n' balls are to be put in 'm' boxes, then at least one box will have more than one ball if $n>m$. Though this is obvious, it is a useful technique.

13. That's why it can't recognize strings of equal number of a's and b's, well-formedness of nested parenthesis etc.

17. Here the `final` state and the `start` state are one and the same. No transition is there. But by definition, there is an (implicit) ε-transition from any state to itself. So, the only string that could be accepted is ε.

22. Write out some of the strings accepted by this automaton. You can see that what is accepted is any string of a's. Since $\Sigma = \{a\}$, the only string that is not accepted is ε. The complement of a language accepts what is not accepted by the language and rejects what is accepted by the language.

24. Let `011011` be the input to the FSM and let it be fed from the right (i.e., least significant digit first). If we add `1` to `011011` we should get `011100`. But how did we obtain it? Whenever we add `1` to an `1`, we make it `0` and carry `1` to the next stage (state) and repeat the process. If we add `1` to a `0`, then first make it `1` and all the more significant digits will remain the same, i.e., a `0` will be `0` and an `1` will be `1`. That's what the given machine does. Hence the answer is (c).

25. 1. This statement is essentially a version of the halting problem which is undecidable.
    2. Context-free languages are closed under union, concatenation, and Kleene star. They are not closed under intersection, difference, and complement.
    3. If L is a regular language, so is its complement language. An FSM accepting this complement language can be constructed by flipping the start and end states.
    4. Recursive languages are closed under union, concatenation, Kleene star, intersection, difference, and complement.

26. Here the initial and the final states are one and the same. If you carefully examine the transition diagram, to move right you have to consume a 'b', to move left a 'b', to go up an 'a' and to go down an 'a'. Whenever we move right, we have to move left at some stage to get back to the final state. This implies, a 'b' essentially has an associated another 'b'. Same is the case with 'a' (since any up (down) has a corresponding down (up)). So, even number of a's and b's have to be present.

29. `S` is the `start` state. `X`→`a`, `Y`→`a` are the only productions that could terminate a string derivable from `X` and `Y` respectively. So at least two a's have to come anyway. Hence the answer is (d).

30. We have `S` → `aB` → `aaBB` → `aabB` → `aabb`.
    So (b) is wrong. We have
    `S` → `aB` → `ab`
    So (c) is wrong.
    A careful observation of the productions will reveal a similarity. Change `A` to `B`, `B` to `A`, `a` to `b` and `b` to `a`. The new set of productions will be the same as the original set. So (d) is false and (a) is the correct answer.

32. Option (b) is wrong because it can't generate `aabb` (in fact any even power). Option (c) is wrong since it generates ε also. Options (a) and (d) are both correct.

34. Option (c) generates the set $\{a^n b^n,\ n=1,2,3,\ldots\}$ which is not regular. Options (a) and (b) being left linear and right linear respectively, should have equivalent regular expressions.

40. Manually parse the string from the left. If you see the prefix *ab* or *aa* or *baa*, delete it and reparse the remaining string. If the parsing eventually leaves you with an empty string, then given string is in L^*.

String 1) is $ab + aa + baa + ab + aa$ – parsing eventually leaves you with an empty string so *abaabaaabaa* is in L^*.

String 2) is $aa + aa + baa + aa$ – parsing eventually leaves you with an empty string so *aaaabaaaa* is in L^*.

String 3) is $baa + aa + ab + aa + aa + b$ – the last *b* cannot be consumed, so *baaaaabaaaab* is not in L^*.

String 4) is $baa + aa + ab + aa$ – parsing eventually leaves you with an empty string so *baaaaabaa* is in L^*.

42. Option (a) is not correct because, transition on input 1 from state 00 should lead to state 01.
Option (b) is not correct because, transition on input 0 from state 00 should lead to state *q*, else, 000 will be accepted.
Option (c) is not correct because, transition on input 1 from state 01 should lead to state 11.
Option (d) has all the missing transitions.

48.
$$L_2^* = a^*$$
$$L_1 L_2^* = a^*$$
$$L_1^* = \Phi$$
So, $L_1 L_2^* \cup L_1^* = a^*$

49. Turing-recognizable languages are also known as recursively enumerable languages. They are closed under union and intersection but not complementation. Turing-decidable languages are also known as recursive languages. They are closed under union, intersection and complementation.

53. Statement 1 is correct because there exists polynomial time algorithm to determine the presence of a cycle in an undirected graph. Statement 2 is correct because, by definition, $P \subseteq NP$. Regarding statement 3, the NP stands for Non-deterministic Polynomial.

54. L_1 is regular because it can be represented by the regular expression 0*1*0* Complement of a regular language is regular. A language that is regular is also context-free.

55. Any finite set has to be regular because you can construct a regular expression that is a union of all the words in the set.

56. Note that 0, 1, and 2 are alphabets. Let us assume 0^0 indicates the absence of 0. This means, when $i = 0$, $0^i 21^i$ indicates the string 2.

Here is the CFG that generates this language.

$$S \rightarrow 0S1 \rightarrow 2$$

This language can be recognized by a deterministic push-down automaton because a deterministic push-down automaton can use the presence of alphabet 2 to delimit the run of 0's.

57. We can use the pumping lemma to show that this language is neither regular nor context-free. We can show that this language is context-sensitive by constructing a linear bounded automaton. So there exists a Turing machine that accepts this language.

58. Consider statement 1. By changing the two non-final states to final states, and changing the final state to a non-final state, we end up with an FSA that accepts the complement language. Any language acceptable by an FSA is context-free. Consider statement 2. There are two routes to reach the final state from the start state – by 11*0 or 0. That is, (11*0 + 0). Once you reach the final state, you can consume any string of 0's and 1's without leaving the final state. So, $L(A) = (11^*0 + 0)(0 + 1)^*$. Appending 0*1*, though redundant, doesn't change this language. Statement 3 is false because there is no need for a 3-state FSA to accept this language. The following 2-state FSA is good enough.

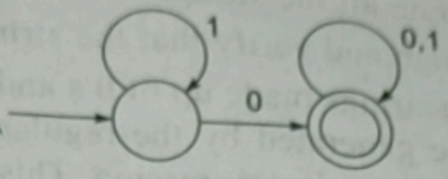

Fig. 11.9

Statement 4 is false because strings made up of only 1's are not accepted.

60. If L is recursively enumerable then its complement is recursively enumerable if and only if L is recursive.

61. Let us consider, as an example, the string 00101 of length 5. What does the following non-deterministic finite state automaton accept?

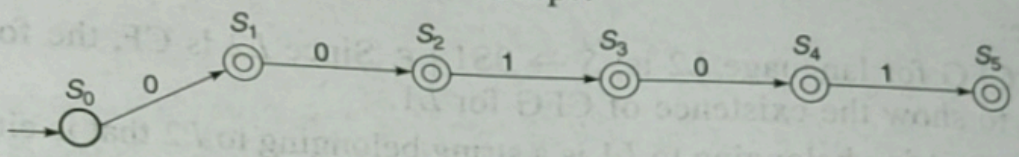

Fig. 11.10

This accepts substrings of 00101 that start with the first 0 (that is, 0, 00, 001, 0010, 00101).

How to modify this non-deterministic finite state automaton so that it accepts (in addition to what is currently accepted) substrings starting with the second 0 as well? By introducing one more transition (edge).

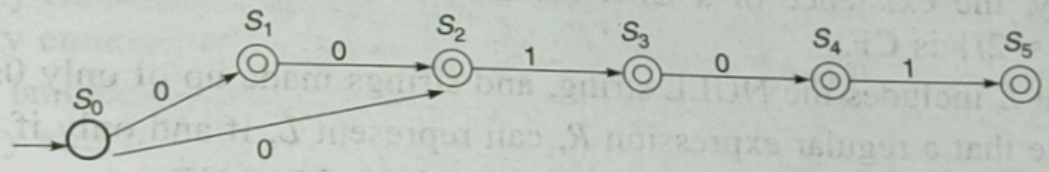

Fig. 11.11

Proceeding this way, the final non-deterministic finite state automaton that accepts al substrings of 00101 (and only those) is,

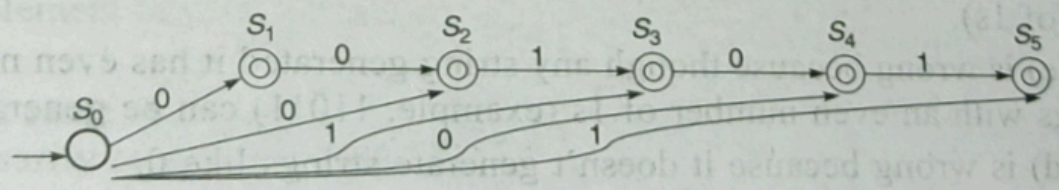

Fig. 11.12

The number of states is 6. That is 1 more than the string length. If the string length is n, the number of states will be $n + 1$.

62. For example, $P = a^*$; $Q = a^*b^*b^*$; $R = PQ = a^*b^*$

64. The first two options can be proved to be correct using De Morgan's laws. Option (c) can be disproved by the following counter-example. Let the universal set U be {a, b, c, d,}. L A = {{a}, {d}, {a, d}, {b, d}, {a, b, d}, {}}. A is closed under union and intersection but is not closed under complementation. For example complement of {a, d} is {b, c, which is not a member of A.

66. II generates strings like xxyyy, which are not supposed to be.
III generates strings like xyy, which are not supposed to be.
I can be verified to generate all the strings in L and only those.

67. Draw the transition diagram and verify that the string 10 from A, leads to C.

68. L is the set of all possible strings made up of 0's and 1's (including the null string). So $L \cup R$ is L, which can be generated by the regular expression $(a+b)^*$, and hence a regular language. R is not a regular expression. This can be proved by using Pumping Lemma or simply by the fact that finite state automata, that recognizes regular expressions, has no memory to record the number of 0's or 1's it has scanned. Without this information 0^n1^n cannot be recognized.

69. Verify by writing CFG for these languages.
Or, verify by constructing push-down automata that can accept these languages.
Note:
The CFG for language $L2$ is: $S \rightarrow 0S1 \mid \varepsilon$. Since $L2$ is CF, the following facts can be used to show the existence of CFG for $L1$.

- Any string belonging to $L1$ is a string belonging to $L2$ that is either prefixed with the regular expression 0^+ or suffixed with the regular expression 1^+
- Any regular grammar is a CFG
- CFG is closed under concatenation

Similarly, the existence of a CFG for L4 can be established because the language $\{0^i1^j \mid i = 2j\}$ is CF (it is generated by the CFG: $S \rightarrow 00S1 \mid \varepsilon$).
Similarly, the existence of a CFG for $L3$ can be established because the language $\{0^i1^j \mid i = 2j\}$ is CF.

70. Note that L includes the NULL string, and strings made up of only 0s.
Also note that a regular expression R, can represent L, if and only if

- R generates all bit strings with even number of 1s AND
- R does not generate any bit string that does not have even number of 1s.

Option (a) is wrong because it doesn't generate strings made up of only 0s (that is number of 1s).
Option (c) is wrong because though any string generated it has even number of 1s, not all strings with an even number of 1s (example, 11011) can be generated by it.
Option (d) is wrong because it doesn't generate strings like 0.

71. This can be proved using induction.

72. Option (d) is wrong because this grammar generates the palindrome *aba* which is not an even length palindrome.
Option (a) is wrong because this grammar doesn't generate palindromes like *abba*.
Option (c) is wrong. Though the strings generated by this grammar start and end with the same symbol, this grammar doesn't generate all strings that start and end with the same symbol.

73. P is the set of all problems that can be solved in polynomial time by a deterministic Turing machine.
NP is the set of all problems that can be solved in polynomial time by a non-deterministic Turing machine.
Since a non-deterministic Turing machine is powerful than a deterministic Turing machine, P is a subset of NP. Whether P = NP is still a open question but if one NP-class problem can be solved in polynomial time by a deterministic Turing machine then any NP-class problem can be solved so.

74. Option (a) is wrong because it generates strings like 010.
Option (b) is wrong because it generates strings like 000.
Option (d) is wrong because it generates strings like 001.

75. The strings common to both the languages will be of the form $a^t b^t c$, where $t \geq 0$. Since finite state automata cannot count and match the number of *a*'s and *b*'s, this cannot be regular. It can be generated by the following CFG.
$S \rightarrow UV$
$U \rightarrow aUb \mid \varepsilon$
$V \rightarrow c$

76. Option (a) is wrong because it doesn't accept all strings that begin with a 0 or 1 (for example, 111 is not accepted)
Option (b) is wrong because it doesn't accept all strings that end with a 0 (for example, 10 is not accepted).
Option (d) is wrong because it doesn't accept all strings that contain the substring 00 (for example, 1001 is not accepted).

77. A recursive language is also recursively enumerable. So, L1 is recursively enumerable.
Recursively enumerable languages are closed under intersection and union. So, L2 ∩ L3 is recursively enumerable, and L2 ∪ L3 is recursively enumerable.
Recursive languages are closed under complementation. So L1′ is recursive, and so also is recursively enumerable.
L2 – L1 is same as L2 ∩ L1′. Since L2 and L1′ are both recursively enumerable, and recursively enumerable languages are closed under intersection, L2 ∩ L1′ is recursively enumerable.
The complement of a recursively enumerable language need not be recursively enumerable.
L1 – L3 is same as L1 ∩ L3′. L3′ need not be recursively enumerable. So, L1 ∩ L3′ is not necessarily recursively enumerable.