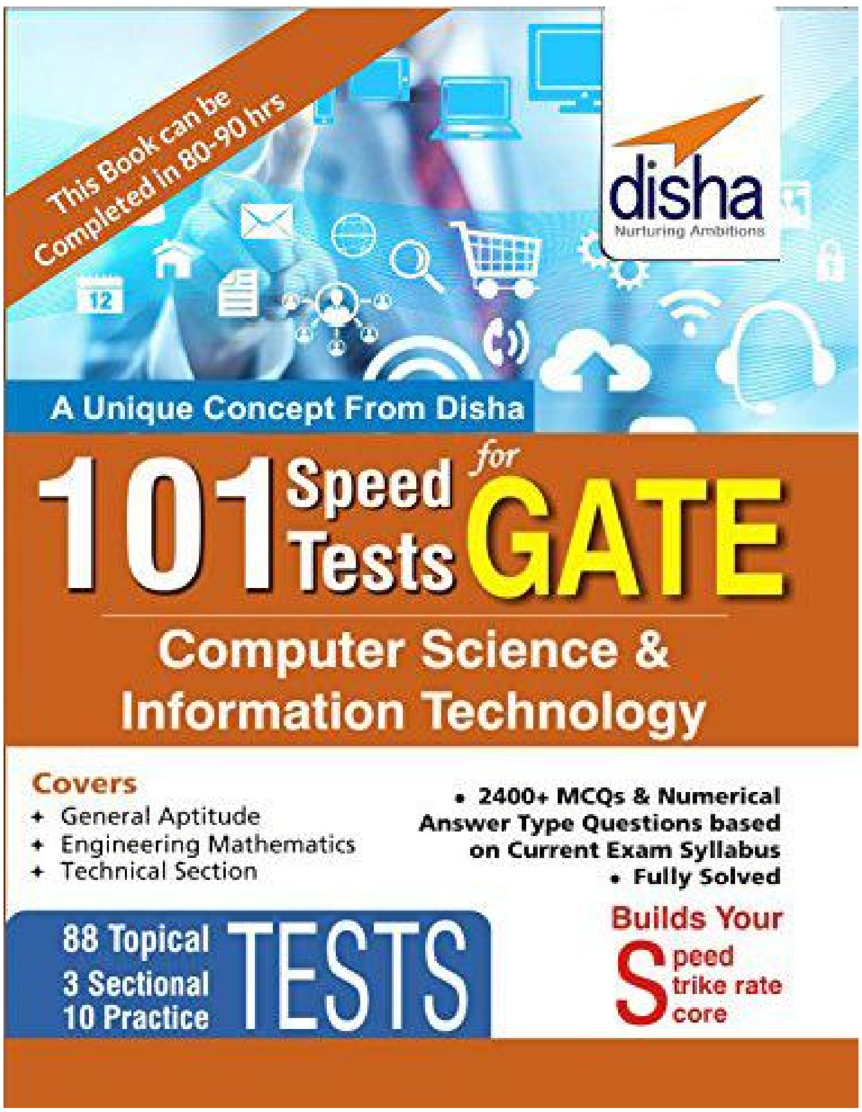
Regular Expressions and Finite Automata

This Chapter " Regular Expressions and Finite Automata" is taken from our Book:

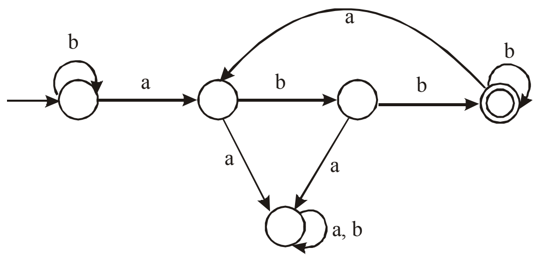


ISBN : 9788193288979

101 SPEED TEST
56 Regular Expressions and Finite Automata

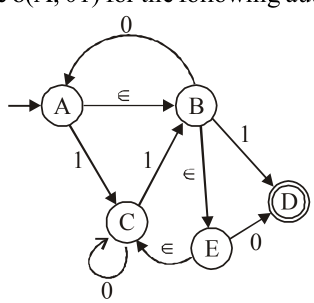
Max. Marks : 20 **No. of Qs. 20** **Time : 30 min.** **Date : ..../......../................**

1. Let L denotes the language generated by the grammar $S \rightarrow 0S0/00$. Which of the following is true?
   (a) $L = 0^+$
   (b) L is regular but not 0^+
   (c) L is context-free but not regular
   (d) L is not context-free
2. Consider the machine M:

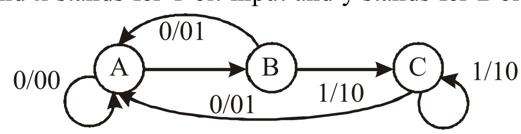


The language recognized by M is
   (a) {W ∈ (a, b)* | every a in W is followed by exactly two b's}
   (b) {W ∈ (a, b)* | every a in W is followed by at least two b's}
   (c) {W ∈ (a, b)* | W contains the substring abb}
   (d) {W ∈ (a, b)* | W does not contain aa as a substring}
3. Which of the following are regular sets?
   1. $\{a^n b^{2m} \mid n \geq 0, m \geq 0\}$
   2. $\{a^n b^m \mid n = 2m\}$
   3. $\{a^n b^m \mid n \neq m\}$
   4. $\{xcy \mid x, y, \in \{a, b\}^*\}$

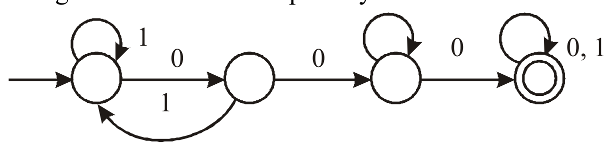
   (a) 1 and 4 (b) 1 and 3
   (c) 1 only (d) 4 only
4. What will be δ(A, 01) for the following automata ?



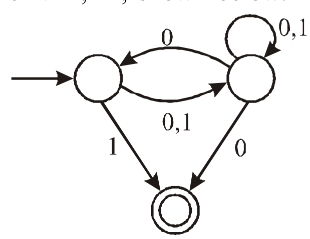
   (a) {D} (b) {B, D}
   (c) {B, C, D} (d) {B, C, D, E}
5. Which one of the following is **TRUE** ?
   (a) The language $L = \{a^n b^n | \geq 0\}$ is regular.
   (b) The language $L = \{a^n | \ n \text{ is prime}\}$ is regular.
   (c) The language $L = \{w \mid w \text{ has } 3k + 1\text{b's for some } k \in N$ with $\Sigma = \{a, b\}\}$ is regular.
   (d) The language $L = \{ww | w \in \Sigma^* \text{ with } \Sigma = \{0, 1\}\}$ is regular.
6. Given an arbitrary Non-deterministic Finite Automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least
   (a) N^2 (b) 2^N
   (c) 2N (d) N!
7. 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—.
8. 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



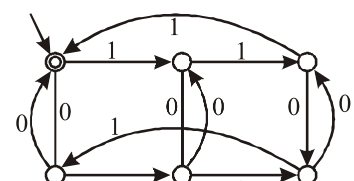
   (a) outputs the sum of the present and the previous bits of the input
   (b) outputs 01 whenever the input sequence contains 11
   (c) outputs 00 whenever the input sequence contains 10
   (d) None of the above
9. Consider the following deterministic finite state automation M: Let S denotes the set of 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



   (a) 1 (b) 5
   (c) 7 (d) 8
10. Consider the NFA, M, shown below:



Let the language accepted by M be L. Let L_1 be the language accepted by the NFA M_1 obtained by changing the accepting state of M to a non-accepting state and by changing the non-accepting state of M to accepting states. Which of the following statements is true?
   (a) $L_1 = \{0, 1\}^* - L$ (b) $L_1 = \{0, 1\}^*$
   (c) $L_1 \subseteq 1$ (d) $L_1 = L$
11. The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively.



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

12. 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) $\{W|N_a(W) > 3N_b(W)\}$
(b) $\{W|N_b(W) > 3N_a(W)\}$
(c) $\{W|N_a(W) = 3k, k \in \{0, 1, 2, ...\}\}$
(d) $\{W|N_b\ (W) = 3k,\ k\ \in \{0,\ 1,\ 2, ...\}\}$

13. A minimum state deterministic finite automation accepting the language $L = \{W \mid W \in \{0, 1\}^*$, number of 0's and 1's in W are divisible by 3 and 5, respectively as states

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

Y:

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

Z:

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

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

(a)

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

(b)

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

(c)

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

(d)

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

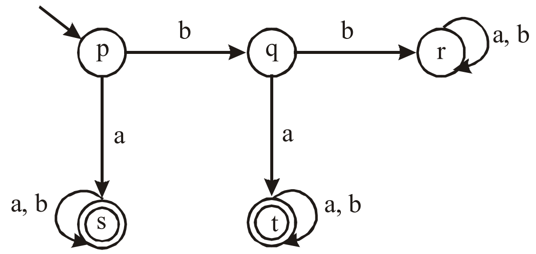
15. Given the following state table of an FSM with two states A and B, one input and one output

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

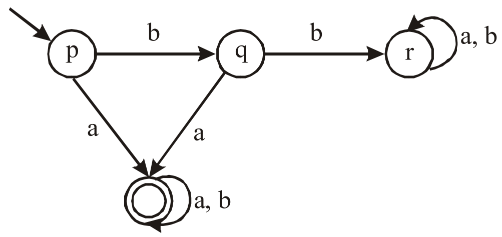
If the initial state is A = 0, B = 0, what is the minimum length of an input string, which will take the machine to the state A = 0, B = 1 with output = 1?

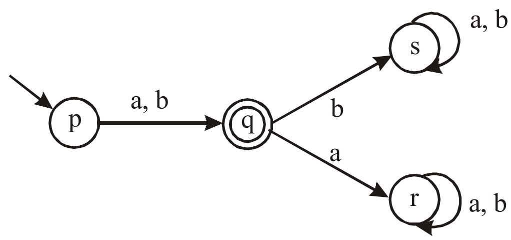
(a) 3 (b) 4
(c) 5 (d) 6

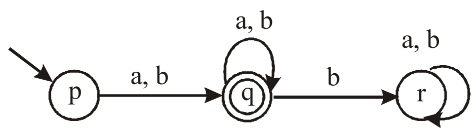
16. A deterministic finite automation (DFA)D with alphabet $\Sigma = \{a, b\}$ is given below.

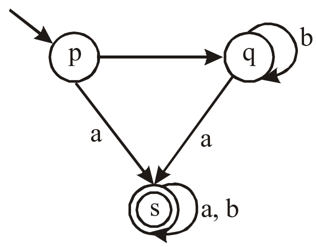


Which of the following finite state machines is a valid minimal DFA which accepts the same language as D?

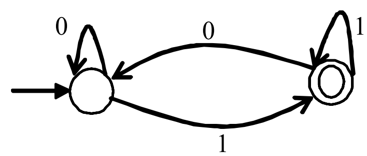
(a)


(b)


(c)


(d)


17. Which of the regular expressions given below represent the following DFA?



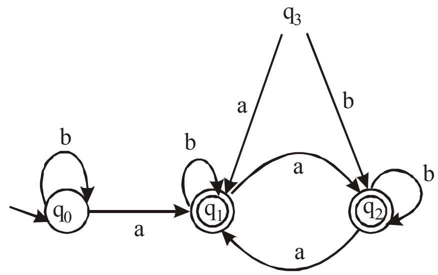
I $0^*1(1+00^*1)^*$ II $0^*1^*1+11^*0^*1$
III $(0+1)^*1$

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

18. Let $L_1 = \{w \in \{0, 1\}^* | w$ 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?

(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

Consider the following Finite State Automation:



19. The language accepted by this automation is given by the regular expression

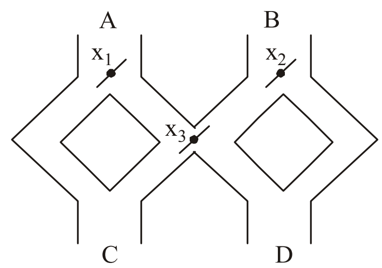
(a) b * ab * ab * ab (b) (a + b)*

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

20. Consider the toy shown in the figure. A marble is dropped in at A or B. Levers x_1, x_2 and x_3 cause the marble to fall eithe to the left or right. Whenever a marble encounters a lever, it causes the lever to change state, so that the next marble to encounter the lever will take the opposite branch.

(a) Model this toy by a Mealy machine. Denote a marble in at A by a O-input and a marble in at B by a 1-input. A sequence of inputs is accepted if the last marble comes at D.

(b) The output of the Mealy machine is the sequences of C's and D's out of which successive marbles fall.



(c) Describe the set accepted by the finite automation.

(d) Suppose that instead the levers switched before allowing the Marble to Pass. Answer for this situation are (a) and (c).

Hints and Solutions

1. **(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.

2. **(b)** a is followed by two or more than 2b's so the language recognized by M is .

3. **(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 , where L_r is language corresponding to the regular expression r. For a recursive definition of .
   4. The (r^+) is used as a regular expression to represent .

   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.

4. **(d)**

5. **(c)** We know that a language L is regular if an equivalent finite Automaton 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 $3k + 1$ number of b's. (i.e. 4, 7, 10 ... number of b's).

6. **(c)** The relation between the NDFA and DFA can be seen in the image below:

It shows that the Non-Deterministic Finite Automaton is superset of Deterministic Finite Automata.
Since, the value of arbitrary finite automata is N, so value of DFA would be the one directly related to it (multiple in most cases).
This implies that value of DFA is 2 N.

7. **16** Since all a's are divisible by 6.

Total states = 7 → a
And all b's are divisible by 8.

Total number of states = 9 → b
So, the minimum number of states in DFA is a + b
= 9 + 7 = 16

8. **(a)** As per the given diagram.

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

9. **(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

10. **(b)** The given machine M is

Now, the complementary machine is

In the case of DFA, but in the case of nfa this is not true.
Infact and L (M) have no connection.
To find L_1 = , we have to look at and directly find its language.
Clearly, , since q_0 is accepting it (0 + 1) (0 + 1)* , since q_2 is accepting it.
∴ = $L_1 = \lambda + (0 + 1)(0 + 1)^*$
∴ $L_1 = (0 + 1)^* = \{0, 1\}^*$

11. **(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.

12. **(b)** DFA of the given grammar is

Clearly, the machine is accepting $N_a(W) = 3k$, where k ∈{0, 1, 2, 3,...}

13. **15** 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:

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

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

Table for Z × Y is

15. **(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.

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

17. **(d)** I, II and III.
Given DFA :

(I) 0*1 (1 + 00 * 1)* it represents the given DFA using following transition:
O* :– 1 $q_0 \to q_0$
1 :– $q_0 \to q_1$
(1 + 00 * 1) * :– $q_1 \to q_0 \to q_0 \to q_1$
(1 +) (0) (0*) (1)
So, the string is accepted.

(II) 0 * 1 * 1 + 11 * 0 * 1 :-
Following transition occurs :–
0 * :– $q_0 \to q_0$
1 * :– $q_0 \to q_0$ (if 1 is absent) or 1 * : $q_0 \to q_1$ (if 1 is present)
(1 + 11*) : $q_0 \to q_1$ $q_1 \to q_1$
0 * :– $q_1 \to q_1$ (if 0 is absent) $q_1 \to q_0$ (if 0 is present)
1 :– $q_1 \to q_1$ $q_0 \to q_1$
final state is q_1, this string is also accepted.

(III) (0 + 1) * 1 :– Following transition occurs :–
(0 + 1) * :– $q_0 \to q_0 \to q_1$
(0 +) (1)
1 :– $q_1 \to q_1$
Ans final state is q_1, so this string is also accepted. Hence (d) is true.

18. **(a)** L_1 is regular but L_2 is not

19. **(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).

20. **(a)** The states in the machine are made up from the positions of the 3 levers. Each lever can be in one of the two possible positions, and .
Let us mark as 0 and as 1.
There are 3 levers, thus the total number of states= 2^3 that may represented by $x_1x_2x_3$ where each of x_1, x_2, x_3 may be 0 or 1.
∴ States in machine 000, 001 ....., 111.
Please do not confuse these 0/1 with 0 or 1 inputs.

Now, suppose a marble arrives at A in the initial condition i.e. all 3 levers in position. The lever x_1 flips and the marble takes the route to the lever x_3 which again flips and the marble finally comes out of D.
Thus the state changes from 000 to 101 on input 0 and D is the output. Similarly, the other transitions may be worked out.