

Name # Hamza Saeed
 ID # 13804
 subject # Compiler construction.

Question#1 Construct regular expression defining each of the following language over the alphabet $\Sigma = \{a, b\}$.

i. All words have even length.

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

ii. All words having at least three a and three b.

Answers

$(a+b)^*(a^3a)^*(b^3b)^*(a+b)^*$

iii. All words having at least double a or triple b.

Answers

$(a+b)^*(a^2)^*(a+b)^* + (a+b)^*(b^3)^*(a+b)^*$

Name # Hamza Saad
 ID # 13804
 Subject # compiler Construction

Page - 2

iv. All words starts with four
 a or triple b.

Answer:-

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

x ——— v ——— x

Question #2

For figure 3 if q_0
 is the initial state, the
 draw a transition
 table for it.

Answer:-

	0	1
q_0	q_1	q_2
q_1	q_2	q_3
q_2	q_3	q_4
q_3	q_3	q_3
q_4	q_4	q_4

x ——— x ——— x

Question 3

Define what is Finite
 Automaton. What can be the
 regular expression of the
 diagram given in figure 1.

Name # Hamza Sreed
ID # 13804
subject# computer construction

Page #3

Answers Finite Automata:-

Finite automata are used to recognize patterns. It takes the string accordingly. When the designed symbol is found then the transition occurs. At the time of transition the automata can be either move to next state or stay in same state.

Finite automata have two more states: Accept state or reject state. When input string is processed successfully, and automata reached final state then it will accept.

A finite Automata is a collection of 5 Tuple.

$(Q, \Sigma, q_0, F, \delta)$

x ————— x ————— x

Name # Hamza Saeed
 ID # 13804
 subject # compiler construction

Page #4

Question #4

Draw a transition table for the diagram given in figure 2. (a) is the starting state and (dotted lines) are the dead transition state's which can be ignored? (10)

Answers:-

	a	b	c	d	e	f	g	h
0	1	-	-	-	2	-	-	-
1	1	3	-	-	2	-	-	-
2	1	-	-	-	2	4	-	-
3	5	-	6	-	7	-	-	-
4	8	-	-	-	9	-	10	-
5	5	-	6	-	7	-	-	-
6	-	-	6	11	7	-	-	-
7	-	-	6	-	7	12	-	-
8	8	-	-	-	9	-	10	-
9	8	-	-	-	9	-	10	-
10	-	-	-	-	9	-	10	12
11	5	-	6	-	7	-	-	-
12	14	-	15	-	16	-	-	-
13	8	-	-	-	9	-	10	-
14	14	-	15	-	16	-	17	-
15	-	-	15	18	-	-	17	-
16	14	-	15	-	16	-	17	-
17	14	-	15	-	16	-	17	19
18	14	-	15	-	16	-	17	-
19	14	-	15	-	16	-	17	-

X ————— X ————— X