



**PES UNIVERSITY, Bangalore**  
(Established under Karnataka Act No. 16 of 2013)  
**Department of Computer Science & Engineering**

**Automata Formal Languages & Logic**

**Question Bank - Unit 1**

**Questions from the Prescribed Textbook**

Topic	Exercise No.	Question No's
Regular expression	3.1 3.2	12 to 21, 27, 28 6,7, 15
Regular expression to Finite Automata	3.2	1-9
Finite Automata to regular expression	3.2	12
Regular expression in practice	3.2	17

**Extra Questions**

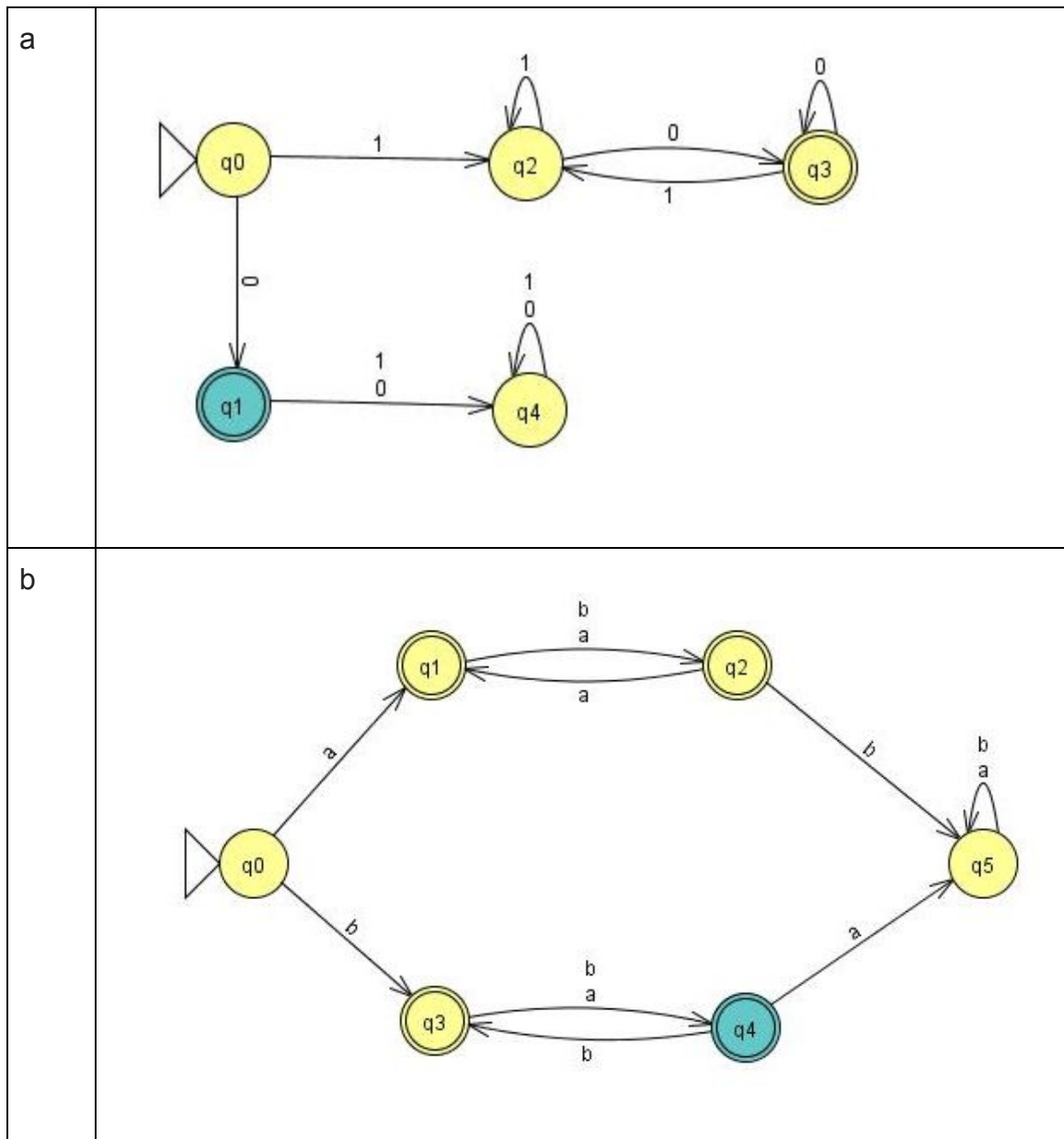
- 1) Construct regular expression for each of the following
  - a) Binary strings containing at least one 00 and at least one 11
  - b) Binary strings with at least two occurrences of at least two consecutive 1 s, the two occurrences not being adjacent (i.e., 011011 is acceptable but 011111 is not).
  - c) Strings over  $\{a, b, c\}$  in which the fourth symbol from the beginning is a  $c$ .
  - d)  $\{w \mid w \text{ begins with a } 1 \text{ and ends with a } 0\}$  over the alphabet  $\Sigma \{0,1\}$
  - e)  $\{w \mid w \text{ has length at least } 3 \text{ and its third symbol is a } 0 \text{ over the alphabet } \Sigma \{0,1\}$

2)convert the following RegEx to an equivalent NFA

- a)  $(0+\lambda)(1+\lambda)(1+2)^*0(2+1)^*$
- b)  $(0+11+10(1+00)^*01)^*$
- c)  $(a+b+c)^*c(a+b)c(a+b)(c+\lambda)^*$
- d)  $0^*1^*0^*$  with three states

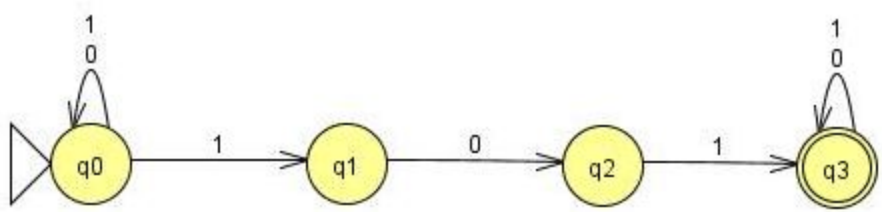
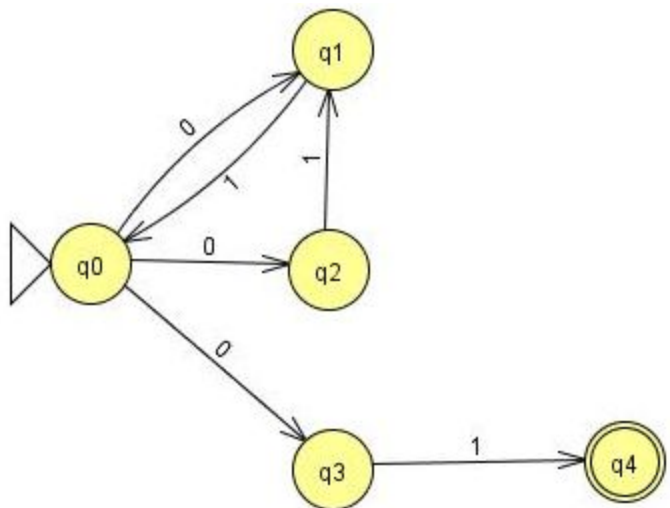
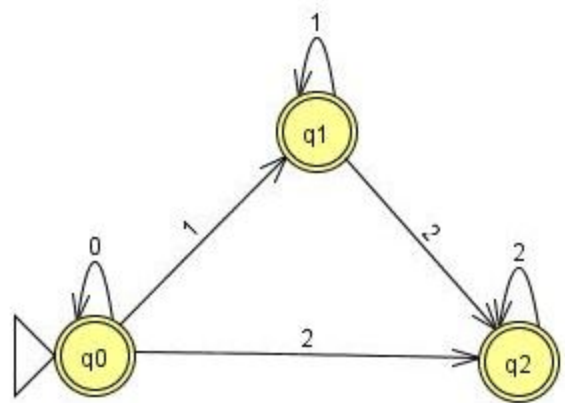
**Automata Formal Languages & Logic**

- e)  $1^*(001^*)^*$  with three states
- 3) Convert the following DFA/NFA to an equivalent RegEx



**PES UNIVERSITY, Bangalore**  
(Established under Karnataka Act No. 16 of 2013)  
**Department of Computer Science & Engineering**

**Automata Formal Languages & Logic**

c	 <pre> graph LR     start(( )) --&gt; q0((q0))     q0 -- 1 --&gt; q1((q1))     q1 -- 0 --&gt; q2((q2))     q2 -- 1 --&gt; q3(((q3)))     q0 -- 1 --&gt; q0     q0 -- 0 --&gt; q0     q3 -- 1 --&gt; q3     q3 -- 0 --&gt; q3         </pre> <p>Finite Automaton for c: States q0, q1, q2, q3. q0 is the start state, q3 is the final state. Transitions: q0 to q1 on 1, q1 to q2 on 0, q2 to q3 on 1. Self-loops on q0 and q3 for both 0 and 1.</p>
d	 <pre> graph LR     start(( )) --&gt; q0(((q0)))     q0 -- 0 --&gt; q1((q1))     q0 -- 1 --&gt; q1     q0 -- 0 --&gt; q2((q2))     q0 -- 0 --&gt; q3((q3))     q2 -- 1 --&gt; q1     q3 -- 1 --&gt; q4(((q4)))         </pre> <p>Finite Automaton for d: States q0, q1, q2, q3, q4. q0 is the start state, q4 is the final state. Transitions: q0 to q1 on 0 and 1, q0 to q2 on 0, q0 to q3 on 0, q2 to q1 on 1, q3 to q4 on 1.</p>
e	 <pre> graph LR     start(( )) --&gt; q0(((q0)))     q0 -- 0 --&gt; q0     q0 -- 1 --&gt; q1(((q1)))     q0 -- 2 --&gt; q2(((q2)))     q1 -- 1 --&gt; q1     q1 -- 2 --&gt; q2     q2 -- 2 --&gt; q2         </pre> <p>Finite Automaton for e: States q0, q1, q2. q0 is the start state, q1 and q2 are final states. Transitions: q0 to q0 on 0, q0 to q1 on 1, q0 to q2 on 2, q1 to q1 on 1, q1 to q2 on 2, q2 to q2 on 2.</p>



# **PES UNIVERSITY, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

## **Department of Computer Science & Engineering**

### **Automata Formal Languages & Logic**

4) Are the following pairs of RegEx's equivalent (do they represent same set of strings)

a)  $(0+\lambda)(11^*0)^*(1+\lambda)$  and  $(1+\lambda)(011^*)^*(0+\lambda)$

b)  $(0^*1^*)^*$  and  $(0+1)^*$

c)  $(0+1)^*(0+\lambda)$  and  $(1+\lambda)(1+0)^*(0+1+\lambda)$

5) Construct regular expression for each of the following

a) All valid dates in the year 2012 expressed in the dd/mm format.

b) All valid names of people: a first name and an optional last name and any number of middle names or middle initials (e.g., James Bond or James H. H. E. Bond); or any number of initials followed by a single name (e.g., J. H. H. E. Bond). First name, middle name and last name are all in init-caps (i.e., only the first letter is capitalized); initials are a capital letter followed by a period.

c) 20. All valid monetary amounts: starting with the Rupee symbol with an optional decimal point and a two-digit fractional amount; if it is larger than three digits, then commas separate billions, millions and thousands (e.g., Rs. 101.99 or Rs. 10,000 or Rs. 1,670,439,444.49).