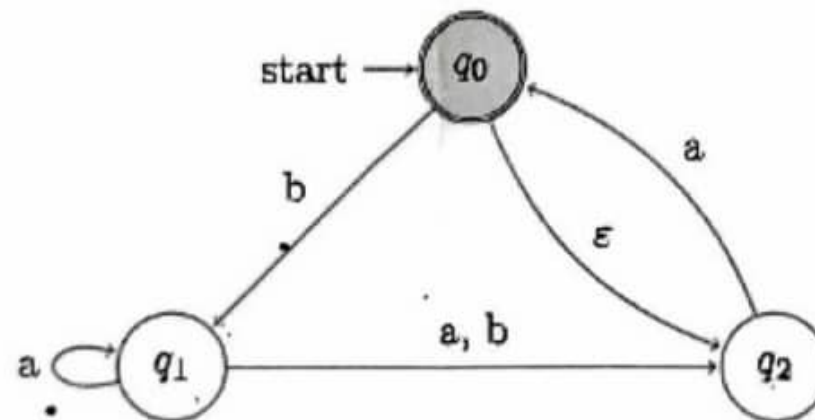


Fall 2024  
CSE228 – Theory of Computation  
Quiz-1

Section: 64\_H

Time: 30 Minutes

- Q1. Draw a DFA for the following language defined over  $\Sigma = \{0,1\}$  [2.5]  
 $L = \{w \mid w \text{ contains either } 010 \text{ or } 101 \text{ as substrings}\}$
- Q2. Draw a NFA for the following language defined over  $\Sigma = \{0,1\}$  [2.5]  
 $L = \{w \mid w \text{ begins or ends with } 00 \text{ or } 11\}$
- Q3. Convert the following NFA to equivalent DFA [10]



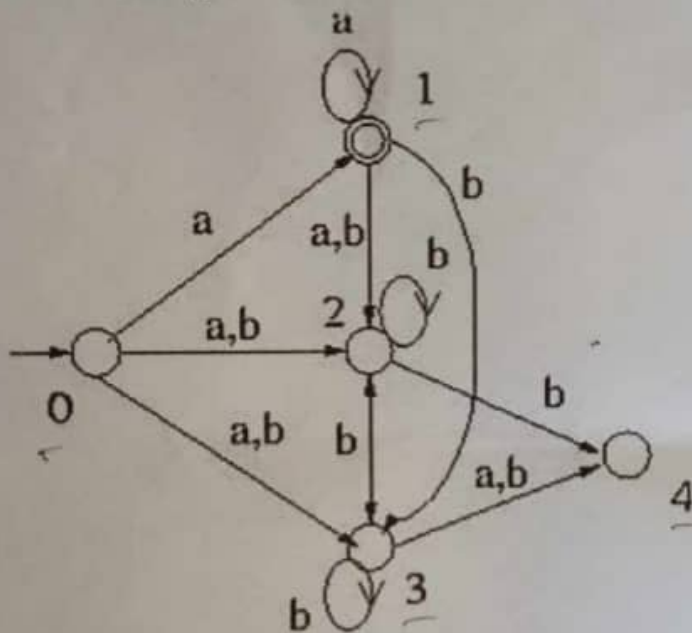
Daffodil International University  
Computer Science & Engineering  
Theory of Computation

Time: 30 minutes

Marks: 15

Quiz 1

1. Construct a DFA that accepts strings over  $\{a, b\}$  that contain exactly two 'a's.
2. Create an NFA that accepts strings over  $\{0, 1\}$  that can have either the pattern "00" or "11" anywhere in the string.
3. Convert the following NFA to DFA


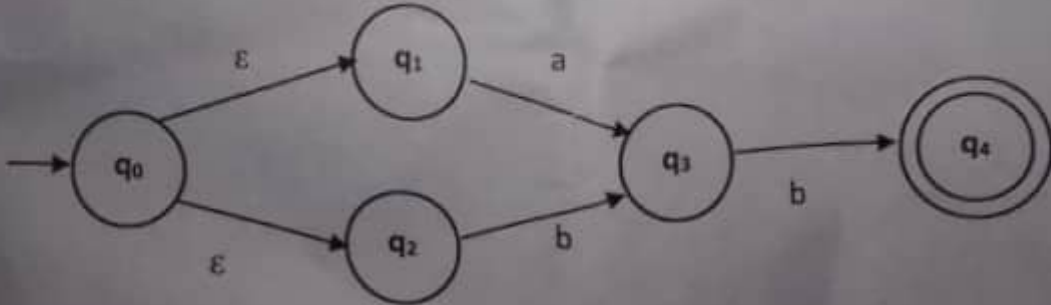


## Quiz – 01

Course Code: CSE228  
Full marks: 15

Course Title: Theory of Computation  
Time: 40 Minutes

Answer each of the following questions

1.	<p>a) What is theory of computation, and what are its primary goals in computer science and mathematics?</p> <p>b) Suppose two languages <math>L_1 = \{a, ab, bb\}</math> and <math>L_2 = \{aa, b\}</math>. Perform intersection, and concatenation operation on the languages.</p>	1+2
2.	<p>a) Let M be the Deterministic Finite Automata (DFA) shown below</p>  <pre> graph LR     start(( )) --&gt; q0((q0))     q0 -- 1 --&gt; q0     q0 -- 0 --&gt; q1((q1))     q1 -- 1 --&gt; q2(((q2)))     q2 -- 1 --&gt; q0     q2 -- 0 --&gt; q2     style start fill:none,stroke:none     </pre> <p>Provide a formal description of M.</p> <p>b) Construct deterministic finite automata to recognize odd number of 1's and even number of 0's?</p>	3+3
3.	<p>a) Construct an NFA accepting binary strings with three consecutive 0's.</p> <p>b) Convert the <math>\epsilon</math> - NFA into its equivalent DFA.</p>  <pre> graph LR     start(( )) --&gt; q0((q0))     q0 -- epsilon --&gt; q1((q1))     q0 -- epsilon --&gt; q2((q2))     q1 -- a --&gt; q3((q3))     q2 -- b --&gt; q3     q3 -- b --&gt; q4(((q4)))     style start fill:none,stroke:none     </pre>	2+4