CS154

Name: ______________________ Date: ___________

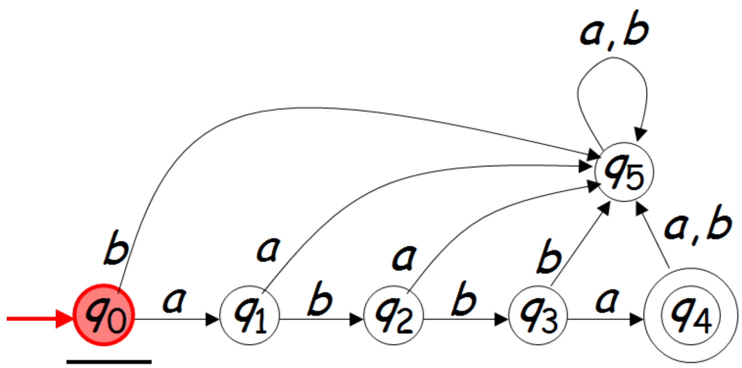
Note: The purpose of the following questions is:

• *Enhance learning*	• *Summarized points*	• *Analyze abstract ideas*

Class 2: Finite Automata

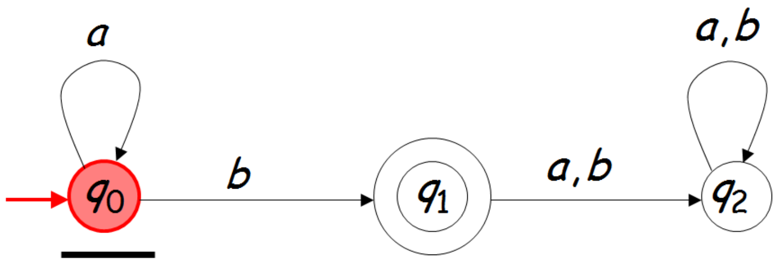
1. [Slide 2-3] What is the Deterministic Finite Accepters (dfa)?
   In common with all automata, a deterministic accepter has internal states, rules for transitions from one state to another, some input, and ways of making decisions.

2. [Slide 4] Consider the following dfa:



   Does it accepts or rejects the following input strings
   a) abba [Slide 5-10]
   b) aba [slide 11-15]
   c) λ [Slide 16-17]

3. Consider the following dfa:



   Does it accepts or rejects the following input strings
   a) aab [Slide 18-22]
   b) bab [Slide 23-27]

4. [Slide 28-36] For the dfa defined in question #2, all of these are incorporated in the following definition.

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

Define the following:

Q =

Σ =

δ =

q_o =

F =

$\delta(q_0, a) =$ $\delta(q_0, b) =$ $\delta(q_2, b) =$

5. [Slide 37] For the dfa in question #2, the transition Function δ, complete the following table

δ	a	b
q_0		
q_1		
q_2		
q_3		
q_4		
q_5		

6. [Slide 38-41] Extended Transition Function δ^* is defined as

$$\delta^* : Q \times \Sigma^* \to Q$$

For the dfa in question #2:

$\delta^*(q_0, ab) =$

$\delta^*(q_0, abba) =$

$\delta^*(q_0, abbbaa) =$

7. [Slide 44-45] In question #2, show how δ^* could be used recursively to prove that

$$\delta^*(q_0, ab) = q_2$$

8. [Slide 46] L(M) = { strings that drive M to a final state}
Define L(M) for the Machine in question #2.

9. Find dfa's for the following languages on $\Sigma = \{a, b\}$.
   a) $L(M) = \{\lambda, ab, abba\}$ [Slide 47-48]
   b) $L(M) = \{a^n b : n \geq 0\}$ [Slide 51]

c) $L(M) = \{\text{all strings with prefix } ab\}$ [Slide 52]
d) $L(M) = \{awa: w \in \{a, b\}^*\}$ [Slide 56]

10. [Slide 53] Find dfa's for the following languages on $\Sigma = \{0, 1\}$.
    $L(M) = \{\text{all strings without substring } 001\}$

11. [Slide 55] Define regular languages, and give few examples for some regular languages.

Reading:
An Introduction to Formal Language and Automata, Peter Linz, 5th edition, Sec 2.1

Selected Exercises:
Section 2.1: 2(c), 5(a), 7(a, d), 9(a, d), 13, 14, 23(a, b), 25.