MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY (MUST), MIRPUR
DEPARTMENT OF SOFTWARE ENGINEERING

Formal Methods in Software Engineering

Lecture [9]: Deterministic Finite Automata

Engr. Samiullah Khan

(Lecturer)

Topics discussed in Today's Lectures

- Deterministic Finite Automata
- DFA with Transition Table
- How a DFA Processes Strings

Deterministic Finite Automata

Definition: A deterministic finite automaton (DFA) consists of

1. a finite set of *states* (often denoted Q)
2. a finite set Σ of *symbols* (alphabet)
3. a *transition function* that takes as argument a state and a symbol and returns a state (often denoted δ)
4. a *start state* often denoted q_0
5. a set of *final* or *accepting* states (often denoted F)

We have $q_0 \in Q$ and $F \subseteq Q$

Deterministic Finite Automata

So a DFA is mathematically represented as a 5-uple

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

The transition function δ is a function in

$Q \times \Sigma \rightarrow Q$

$Q \times \Sigma$ is the set of 2-tuples (q, a) with $q \in Q$ and $a \in \Sigma$

DFA with Transition Table

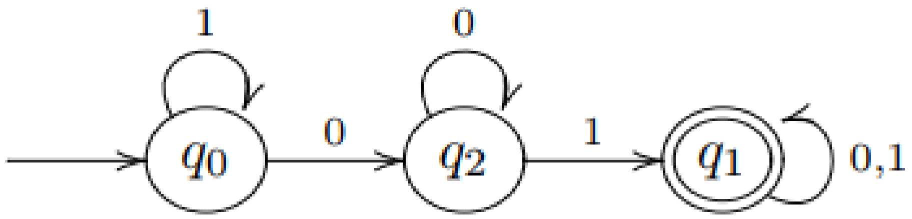
How to present a DFA? With a *transition table*

	0	1
$\rightarrow q_0$	q_2	q_0
$*q_1$	q_1	q_1
q_2	q_2	q_1

The $\rightarrow$ indicates the *start* state: here q_0

The $*$ indicates the final state(s) (here only one final state q_1)

This defines the following *transition diagram*



Deterministic Finite Automata

For this example

$Q = \{q_0, q_1, q_2\}$

start state q_0

$F = \{q_1\}$

$\Sigma = \{0, 1\}$

δ is a *function* from $Q \times \Sigma$ to Q

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

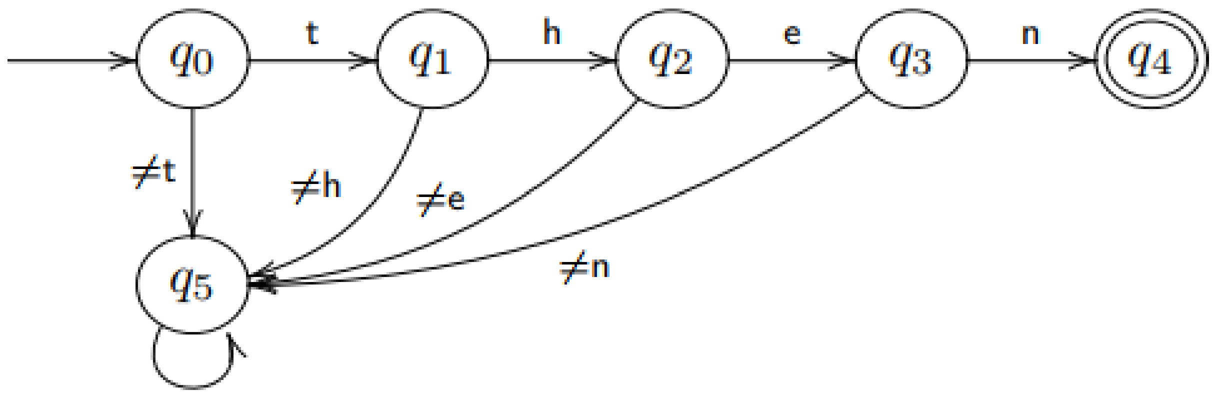
$\delta(q_0, 1) = q_0$

$\delta(q_0, 0) = q_2$

DFA Example: password

When does the automaton accepts a word??

It reads the word and accepts it if it stops in an accepting state



Only the word then is accepted

Here $Q = \{q_0, q_1, q_2, q_3, q_4\}$

Σ is the set of all characters

$F = \{q_4\}$

We have a “stop” or “dead” state q_5, *not* accepting

How a DFA Processes Strings

Let us build an automaton that accepts the words that contain 01 as a subword

$\Sigma = \{0, 1\}$

$L = \{x01y \mid x, y \in \Sigma^*\}$

We use the following states

A: start

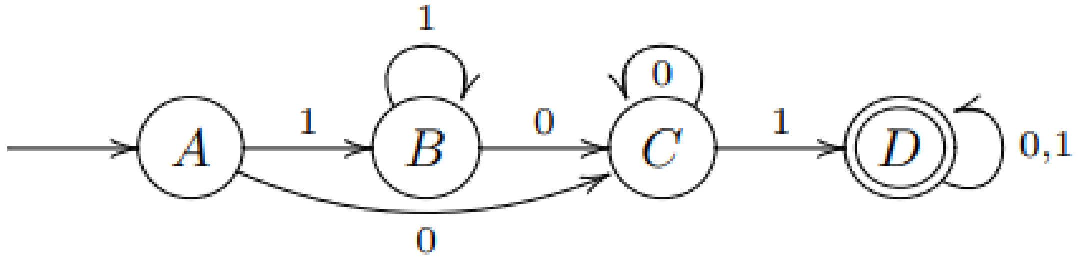
B: the most recent input was 1 (but not 01 yet)

C: the most recent input was 0 (so if we get a 1 next we should go to the accepting state D)

D: we have encountered 01 (accepting state)

How a DFA Processes Strings (Contd…)

We get the following automaton

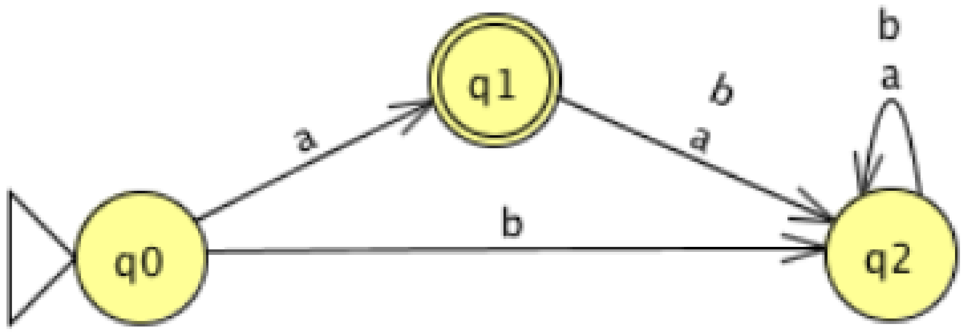


Transition table

	0	1
→A	C	B
B	C	B
C	C	D
*D	D	D

$Q = \{A,B,C,D\}$, $\Sigma = \{0,1\}$, start state A, final state(s) $\{D\}$

DFA Example 2



Formally, a DFA is described by a 5-tuple (Q, Σ, δ, q0, F) in which

Q is a finite set of states,

Σ is a finite set of input symbols also known as the alphabet,

δ is a state transition function ($\delta : Q \times \Sigma \rightarrow Q$),

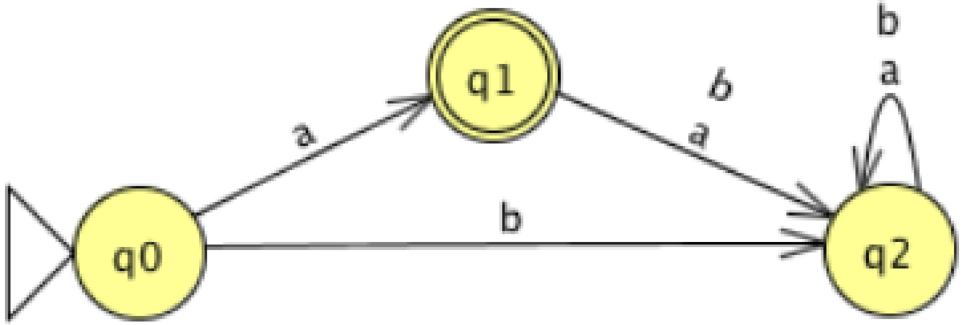
q0 is the start state and

F is a set of accept states

For example, the DFA whose state diagram is shown above is represented by the 5-tuple

({q0, q1, q2}, {a, b}, δ as given by state diagram above, q0, {q1})

DFA Example 2 – Contd…

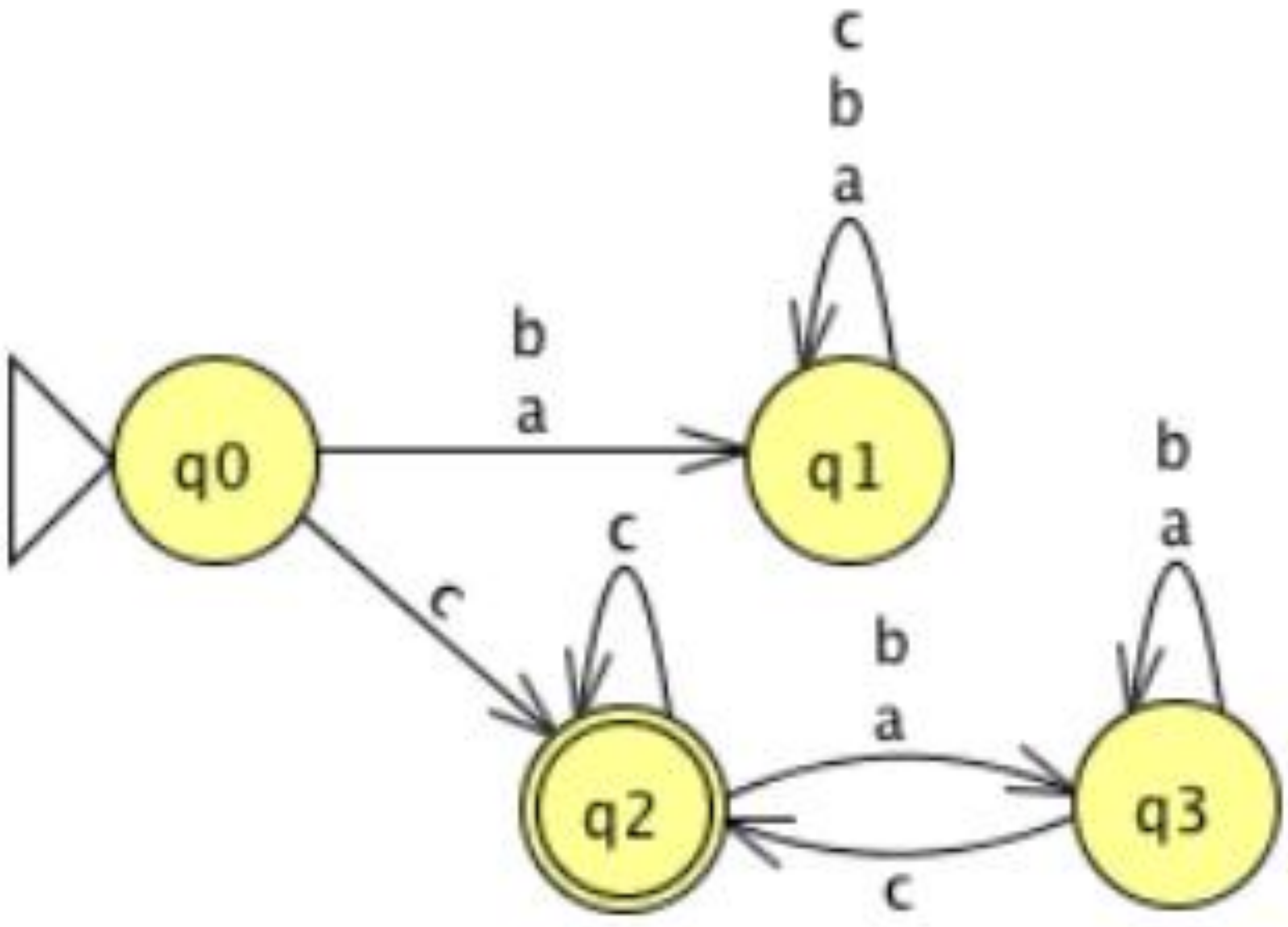


The transition function given by state diagram above is also described by the following table:

	a	**b**
q0	q1	q2
q1	q2	q2
q2	q2	q2

In the transition function of a DFA, every state has exactly one transition associated with each symbol of the alphabet. Because the DFA determines the unique next state for each next input symbol, this is a deterministic finite automaton.

DFA Task - To be done by Students



THANKS