CSE 2233 Theory of Computing

Shekh. Md. Saifur Rahman

Lecturer, CSE Department

UIU

Terminology - Strings

- An alphabet is a finite, nonempty set of symbols. Conventionally, we use the symbol Σ for an alphabet. For example,
  - i. Σ = 0, 1 , the binary alphabet.
  - ii. Σ = { a, b, c, … … … , z}, the set of all lower-case letters.
- A string over an alphabet Σ is a finite sequence of symbols, where each symbol is an element of Σ –
  - The string w where w = $w_1 w_2 w_3 \ldots w_n$ represents a string of length n where each $w_i \in \sum$
  - The reverse of w is represented as w_R
  - The length of w denoted by |w|, is the number of symbols contained in w.
  - The empty string, denoted by ε, is the string having length zero.
- Ex: if the alphabet Σ is equal to {0, 1}, then 10, 1000, 0, 101, and ε are strings over Σ, having lengths 2, 4, 1, 3, and 0 respectively.

Continued...

- Power of an Alphabet:

  Σ^1 -> Length is 1

Σ^k = set of string of length k, where each and every symbol of the strings belongs to Σ.

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

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

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 = \{000, 001, 011, 100, 101, 010, 110, 111\}$

$\Sigma^0 = \{\varepsilon\}$

Continued…

$\Sigma^0, \Sigma^1, \Sigma^2, \dots, \Sigma^k$

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \dots$

$\Sigma^* = \{\varepsilon\} \cup \Sigma^+$

Language:

$\Sigma = \{a, b\}$

L = {any string that starts with a}
= {a, aa, ab, aaa, a, ….}

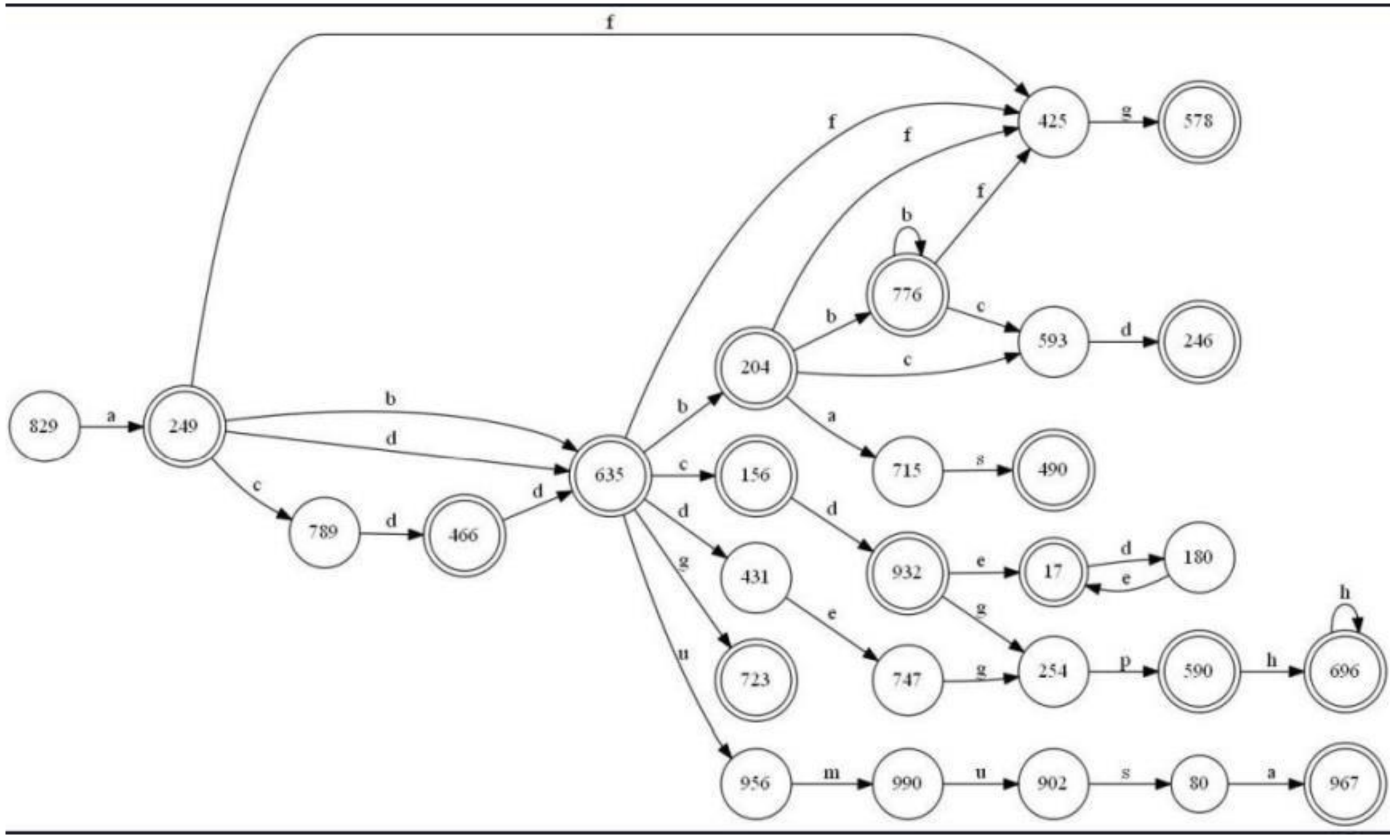
L = {any strings that starts with a and it’s length is 2}
= {aa, ab}

Terminology - Language

- Language: A set of strings all of which are chosen from some $\Sigma*$, where Σ is a particular alphabet, is called a language.
- Regular Language: These languages are exactly the ones that can be described/recognized by Finite Automata. This language can be processed by computers having a very small amount of memory.
- Here,

- Computer is meant by a computational model that varies depending on the feature we want to focus on.

Finite Automata



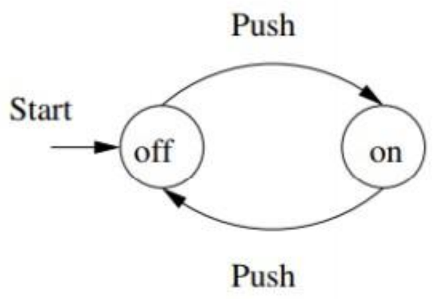
Finite Automata

Finite automata are good models for computers with an extremely limited amount of memory.

What can a computer do with such a small memory?

On-off switch Finite Automaton?

Example 1.1: Perhaps the simplest nontrivial finite automaton is an on/off switch. The device remembers whether it is in the "on" state or the "off" state, and it allows the user to press a button whose effect is different, depending on the state of the switch. That is, if the switch is in the off state, then pressing the button changes it to the on state, and if the switch is in the on state, then pressing the same button turns it to the off state.



Heading towards the definition

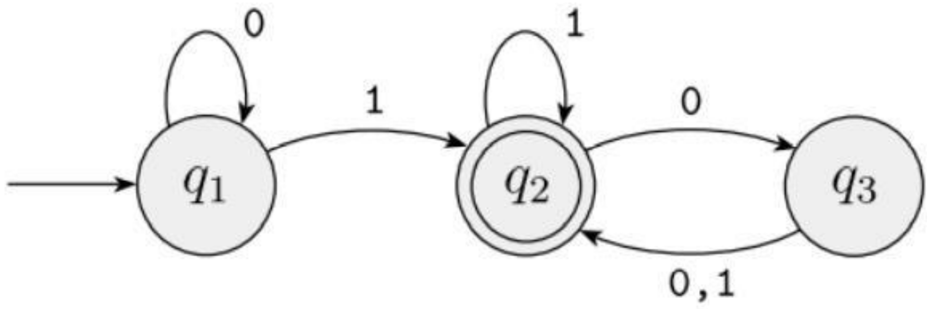


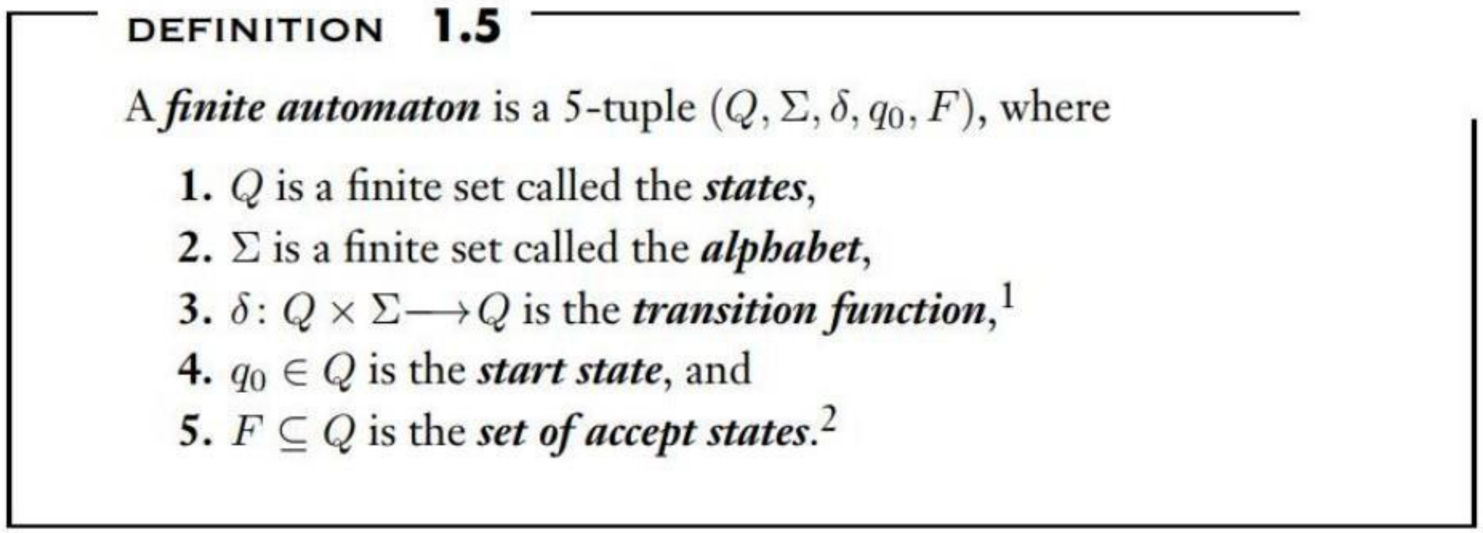
FIGURE 1.4
A finite automaton called M_1 that has three states

Finite Automata – Still not formal definition

- Finite Automata: A finite automaton (FA) is a simple idealized machine used to recognize patterns within input taken from some character set (or alphabet) ∑ . The job of an FA is to accept or reject an input depending on whether the pattern defined by the FA occurs in the input.

- Graphical Representation:
  - Node (for states)
  - Arcs (for transitions)

Formal definition of Finite Automaton – Finally.



DEFINITION 1.5

A ***finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set called the ***states***,
2. Σ is a finite set called the ***alphabet***,
3. $\delta \colon Q \times \Sigma \longrightarrow Q$ is the ***transition function***,[1]
4. $q_0 \in Q$ is the ***start state***, and
5. $F \subseteq Q$ is the ***set of accept states***.[2]

Example

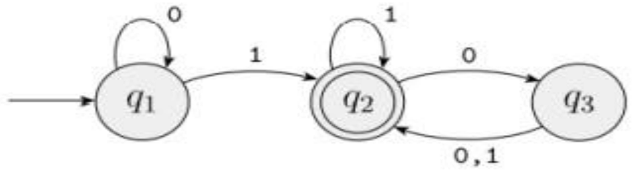


FIGURE 1.6
The finite automaton M_1

We can describe M_1 formally by writing $M_1 = (Q, \Sigma, \delta, q_1, F)$, where

1. $Q = \{q_1, q_2, q_3\}$,
2. $\Sigma = \{0,1\}$,
3. δ is described as

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2,

4. q_1 is the start state, and
5. $F = \{q_2\}$.

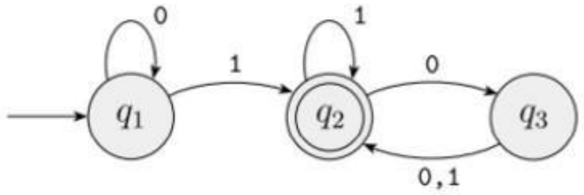
FA – Machine M

- If A is the set of all strings that machine M accepts
- We say that A is the language of machine M
- We write L(M) = A.
- We say that M recognizes A or that M accepts A.

FA- Machine M

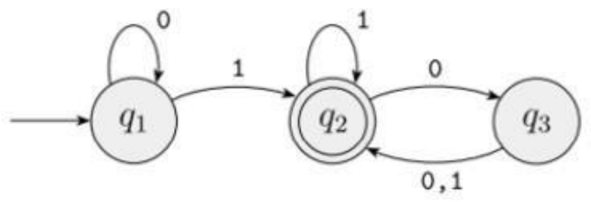
- A machine may accept several strings, but it always recognizes only one language.
- If the machine accepts no strings, it still recognizes one language - the empty language Ø.
- For this example, we say if L(M) = A , then

  A = {w| w contains at least one 1 and an even number of 0s follow the last 1}



Deterministic Finite Automaton (DFA)

The term “deterministic finite automata” refers to the fact that on each input there is one and only one state to which the automaton can transition from its current state.



Example:

Σ = { a, b}

L : set of string of length 2

{aa, ab, ba, bb}

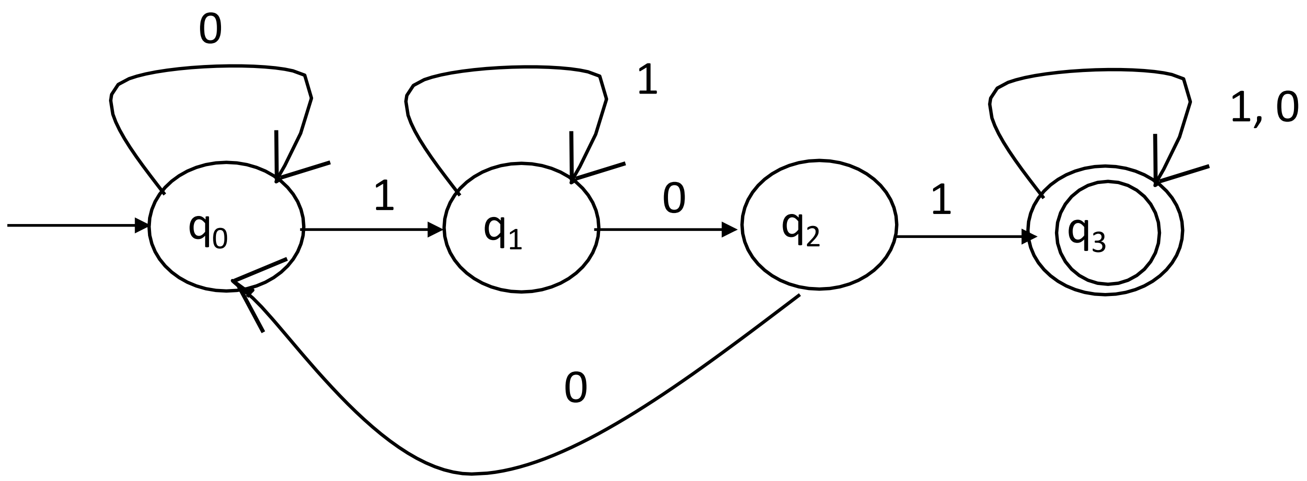
$$\begin{tikzcd}
{} \arrow[r] & q_0 \arrow[r, "{a, b}"] & q_1 \arrow[r, "{a, b}"] & q_2 \arrow[r, "{a, b}"] & q_3 \arrow[loop above, "{a, b}"]
\end{tikzcd}$$

Example:

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

L : {a set of string that has 101 as substring}

{101, 0101, ….}



Continued…

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

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

δ is described as

	0	**1**
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3	q_3	q_3

Thank You