CSE 2001:
Introduction to Theory of Computation

Fall 2012

Suprakash Datta

datta@cse.yorku.ca

Office: CSEB 3043

Phone: 416-736-2100 ext 77875

Course page: http://www.cs.yorku.ca/course/2001

Next: Finite automata

Ch. 1.1: Deterministic finite automata (DFA)

We will :

• Design automata for simple problems

• Study languages recognized by finite automata.

Recognizing finite languages

- Just need a lookup table and a search algorithm
- Problem – cannot express infinite sets, e.g. odd integers

Finite Automata

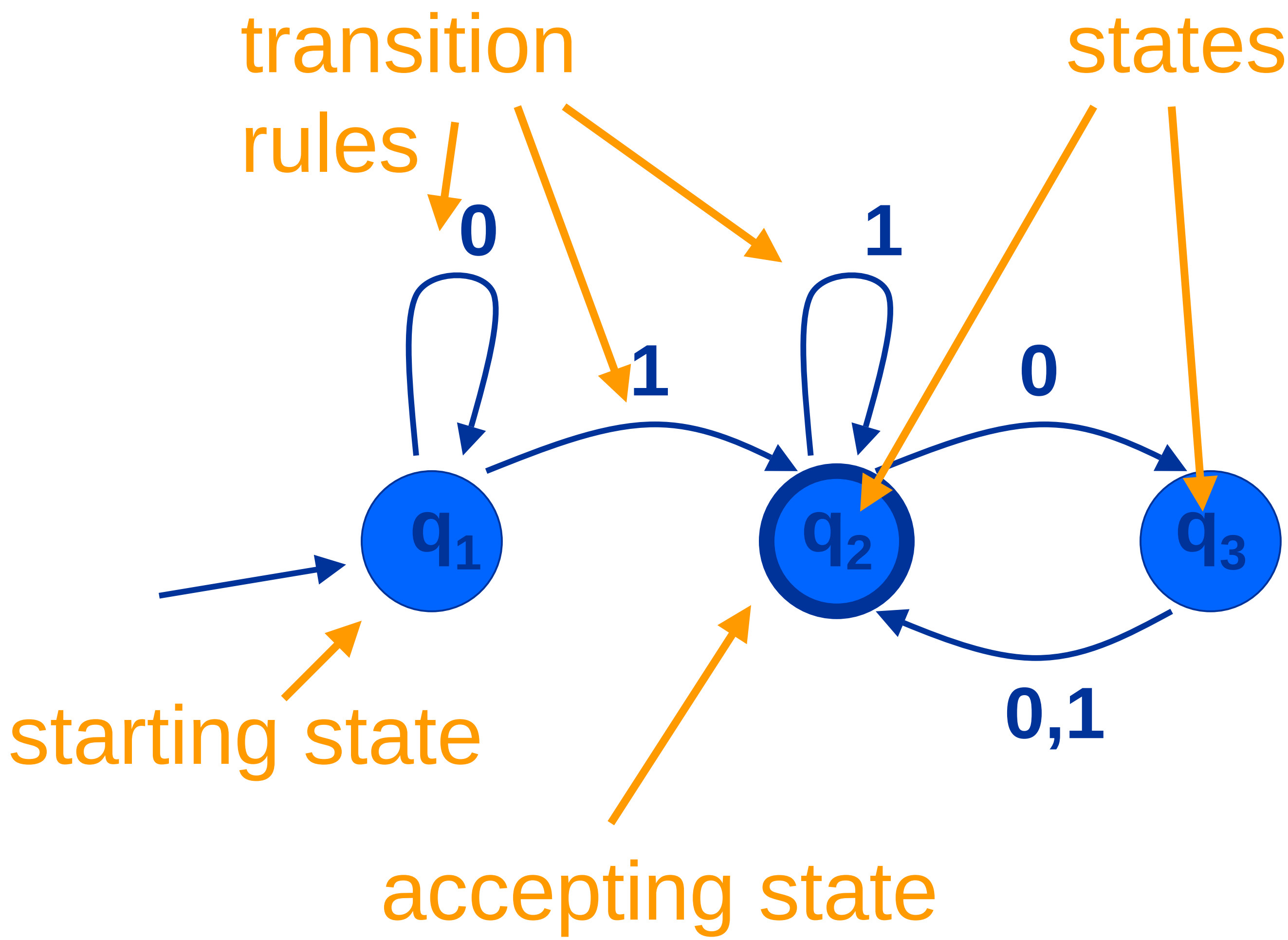
The simplest machine that can recognize an infinite language.

"Read once", "no write" procedure.

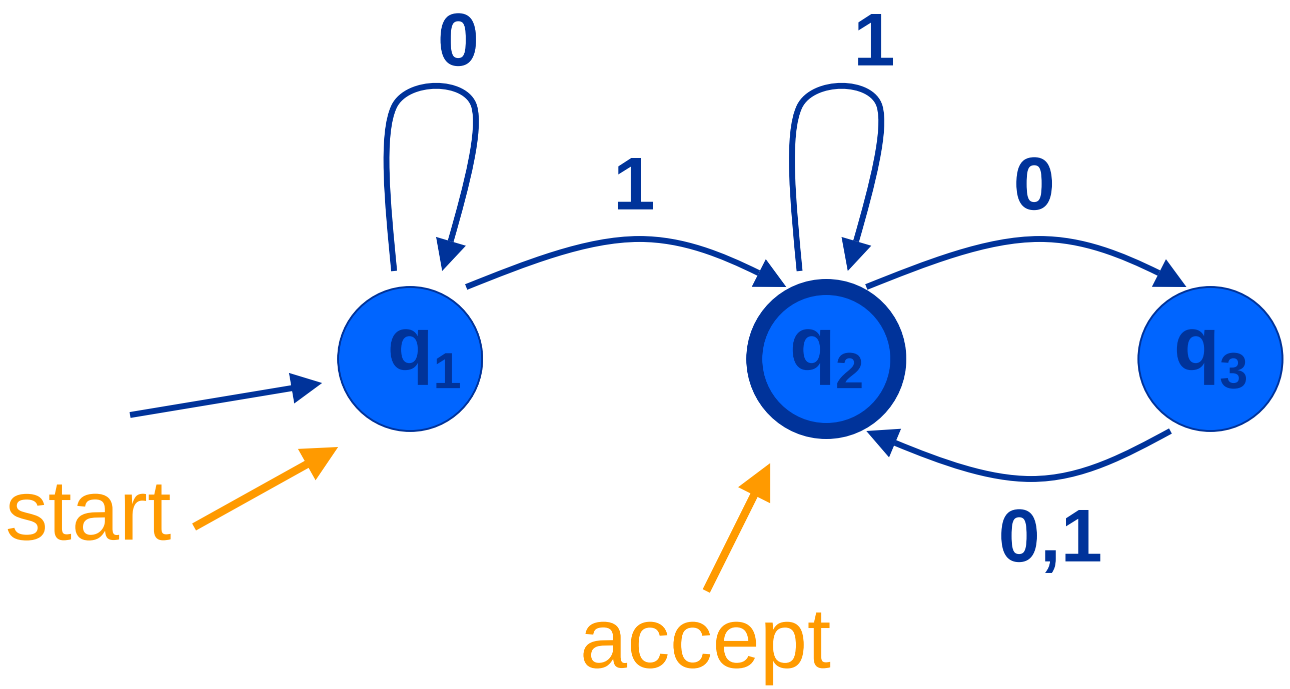
Useful for describing algorithms also.
Used a lot in network protocol description.

Remember: DFA's can accept finite languages as well.

A Simple Automaton (0)



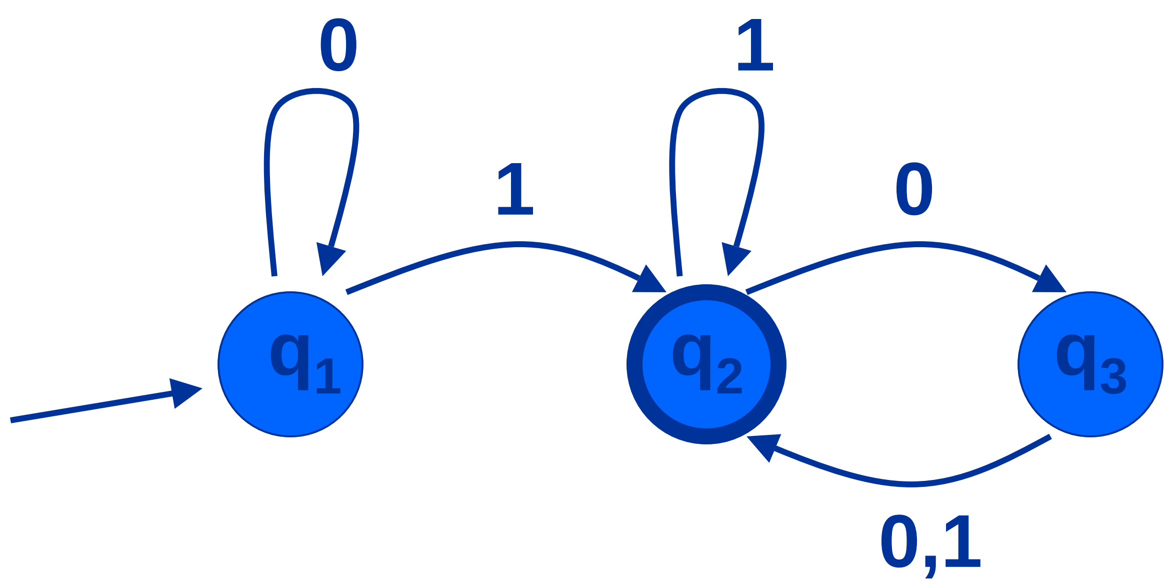
A Simple Automaton (1)



on input “0110”, the machine goes:

$q_1 \rightarrow q_1 \rightarrow q_2 \rightarrow q_2 \rightarrow q_3$ = “reject”

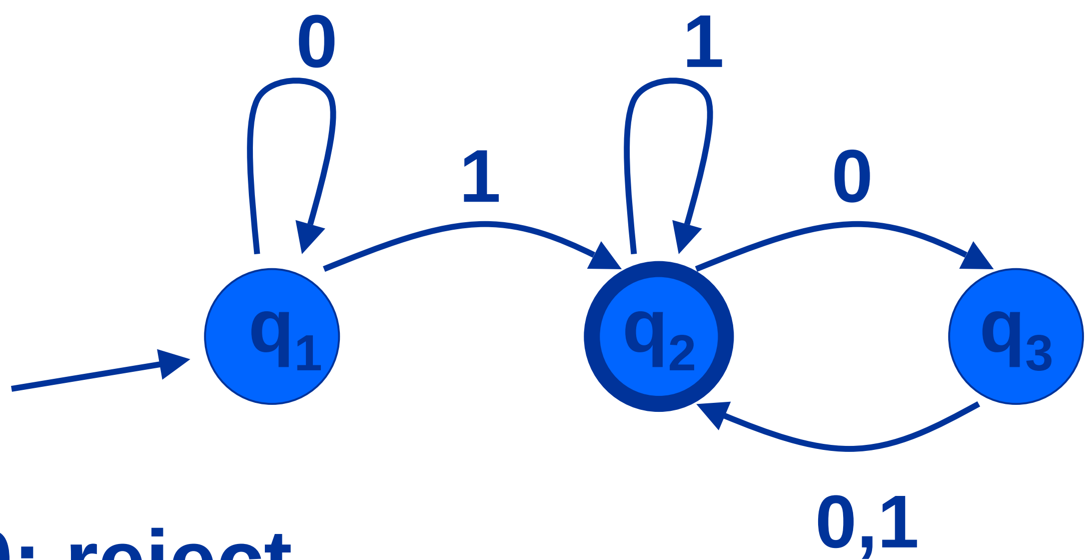
A Simple Automaton (2)



on input “101”, the machine goes:

$q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_2$ = “accept”

A Simple Automaton (3)



010: reject

11: accept

010100100100100: accept

010000010010: reject

ε: reject

Examples of languages accepted by DFA

- L = { w | w ends with 1}
- L = { w | w contains sub-string 00}
- L = { w | |w| is divisible by 3}
- L = { w | |w| is odd or w ends with 1}
- L = { w | |w| is divisible by 10^6}

Note: Σ = {0,1} in each case

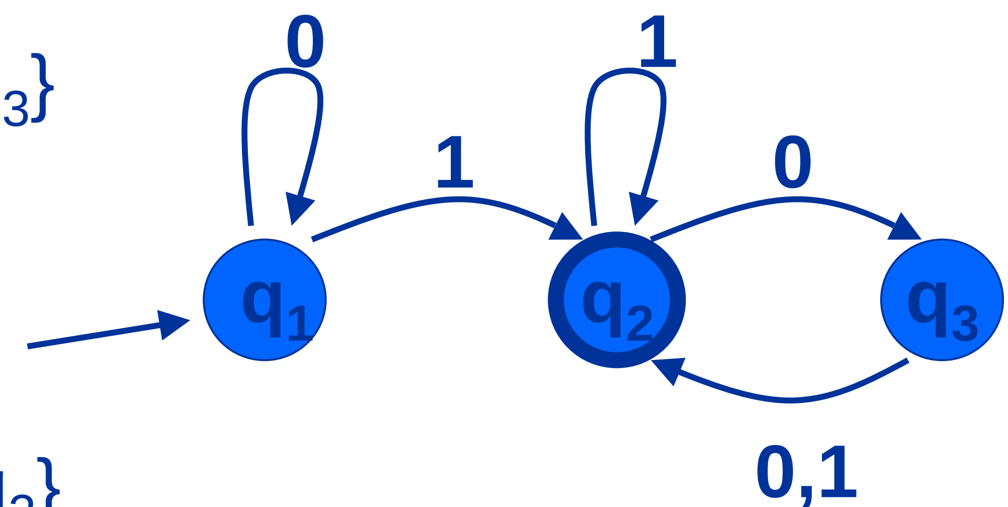
DFA design

- Design DFA for language
  - $L = \{w \in \{0,1\}^* \mid w \text{ contains substring } 01\}$
- Three states to remember:
  - Have seen the substring 01
  - Not seen substring 01 and last symbol was 0
  - Not seen substring 01 and last symbol was not 0
- General principles?

DFA : Formal definition

- A deterministic finite automaton (DFA) M is defined by a 5-tuple $M=(Q,\Sigma,\delta,q_0,F)$
  - Q: finite set of states
  - Σ: finite alphabet
  - δ: transition function $\delta:Q\times\Sigma\rightarrow Q$
  - $q_0 \in Q$: start state
  - $F \subseteq Q$: set of accepting states

M = (Q,Σ,δ,q,F)



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

alphabet Σ = {0,1}

start state q_1

accept states $F=\{q_2\}$

transition function δ:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Recognizing Languages (defn)

A finite automaton **M = (Q,Σ,δ,q,F)** accepts a string/word **w** = $w_1 \ldots w_n$ if and only if there is a sequence $r_0 \ldots r_n$ of states in Q such that:

1) $r_0 = q_0$

2) $\delta(r_i, w_{i+1}) = r_{i+1}$ for all $i = 0, \ldots, n-1$

3) $r_n \in F$

Regular Languages

The language recognized by a finite automaton M is denoted by **L(M).**

A <u>regular language</u> is a language for which there exists a recognizing finite automaton.

Two DFA Questions

Given the description of a finite automaton **M = (Q,**Σ**,**δ**,q,F),** what is the language **L(M)** that it recognizes?

In general, what kind of languages can be recognized by finite automata? (What are the regular languages?)