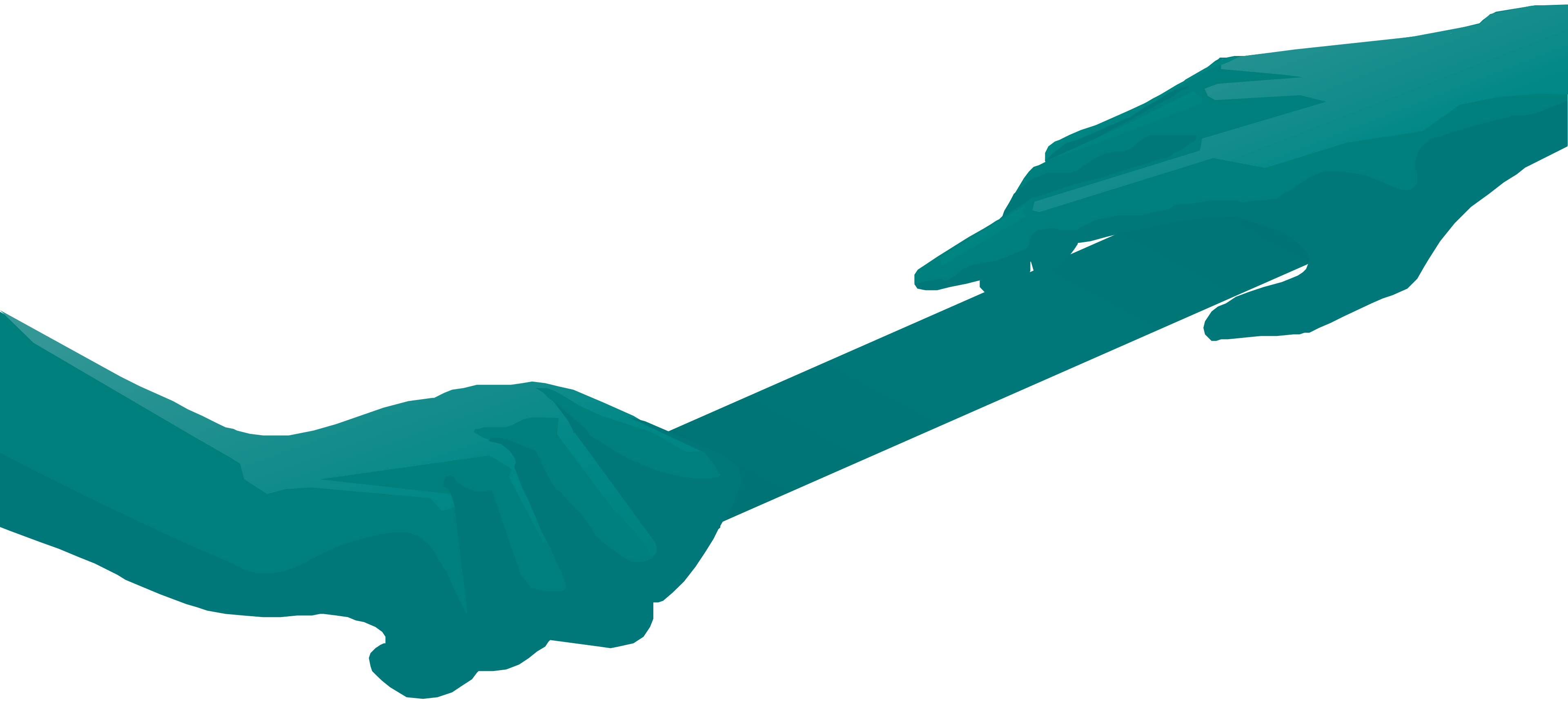
Automata and Languages

Prof. Mohamed Hamada

Software Engineering Lab.
The University of Aizu
Japan

FINITE STATE MACHINES (AUTOMATA)

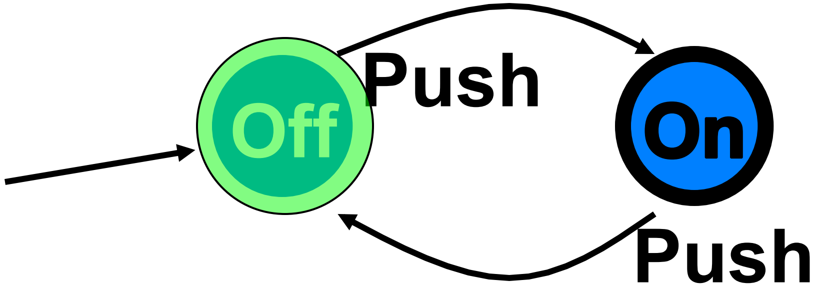


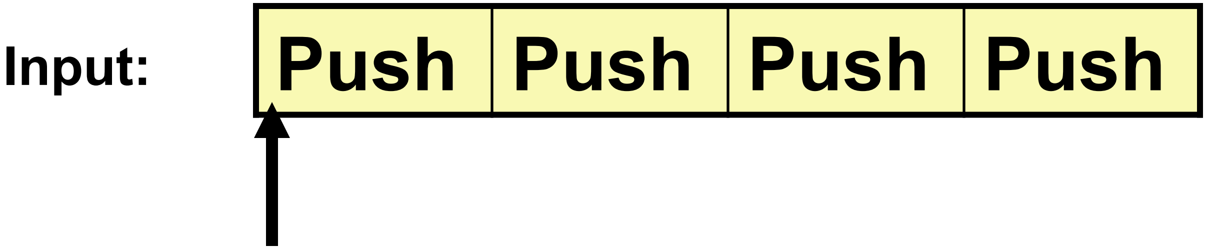
Switch Example 1

Think about the On/Off button

Switch Example 1

The corresponding Automaton

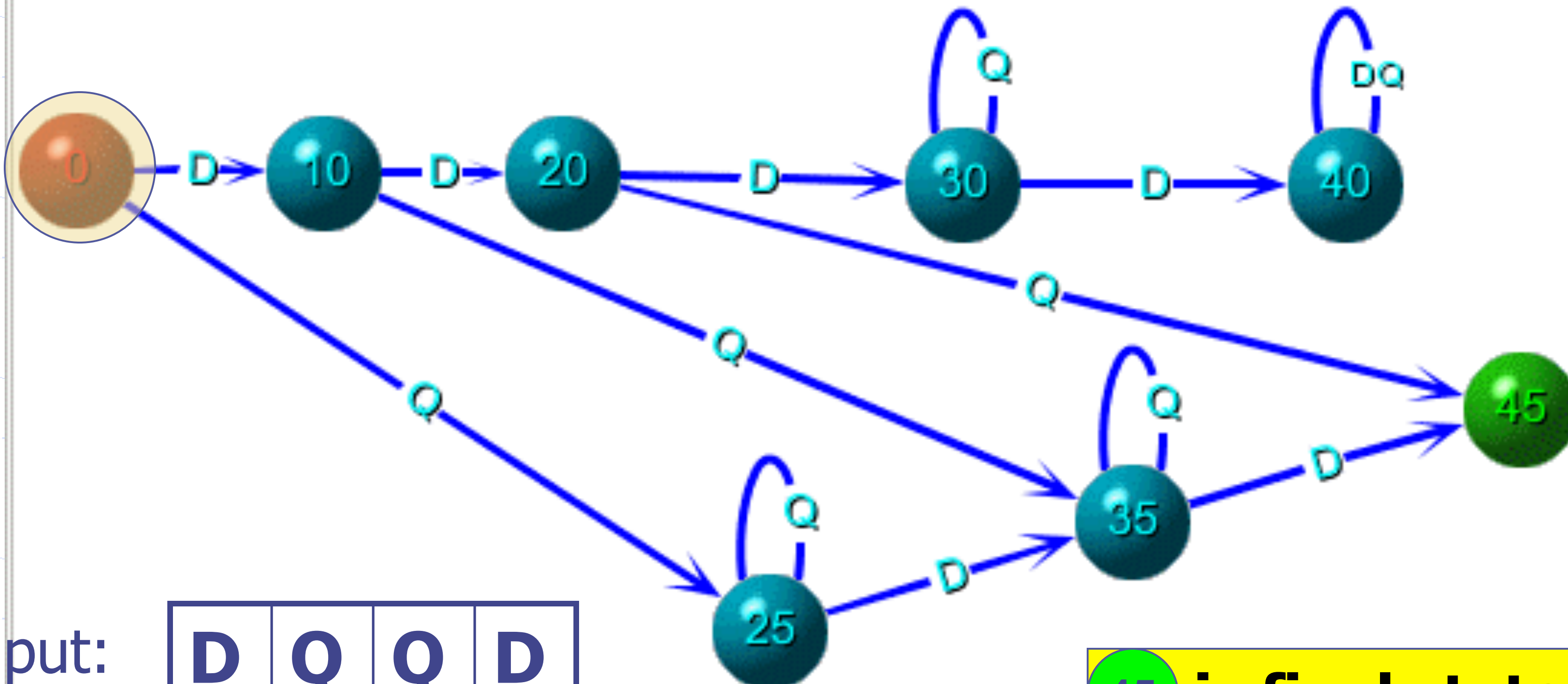




Vending Machine Example 2

- Vending machine dispenses Cola for $0.45

Vending Machine Example 2

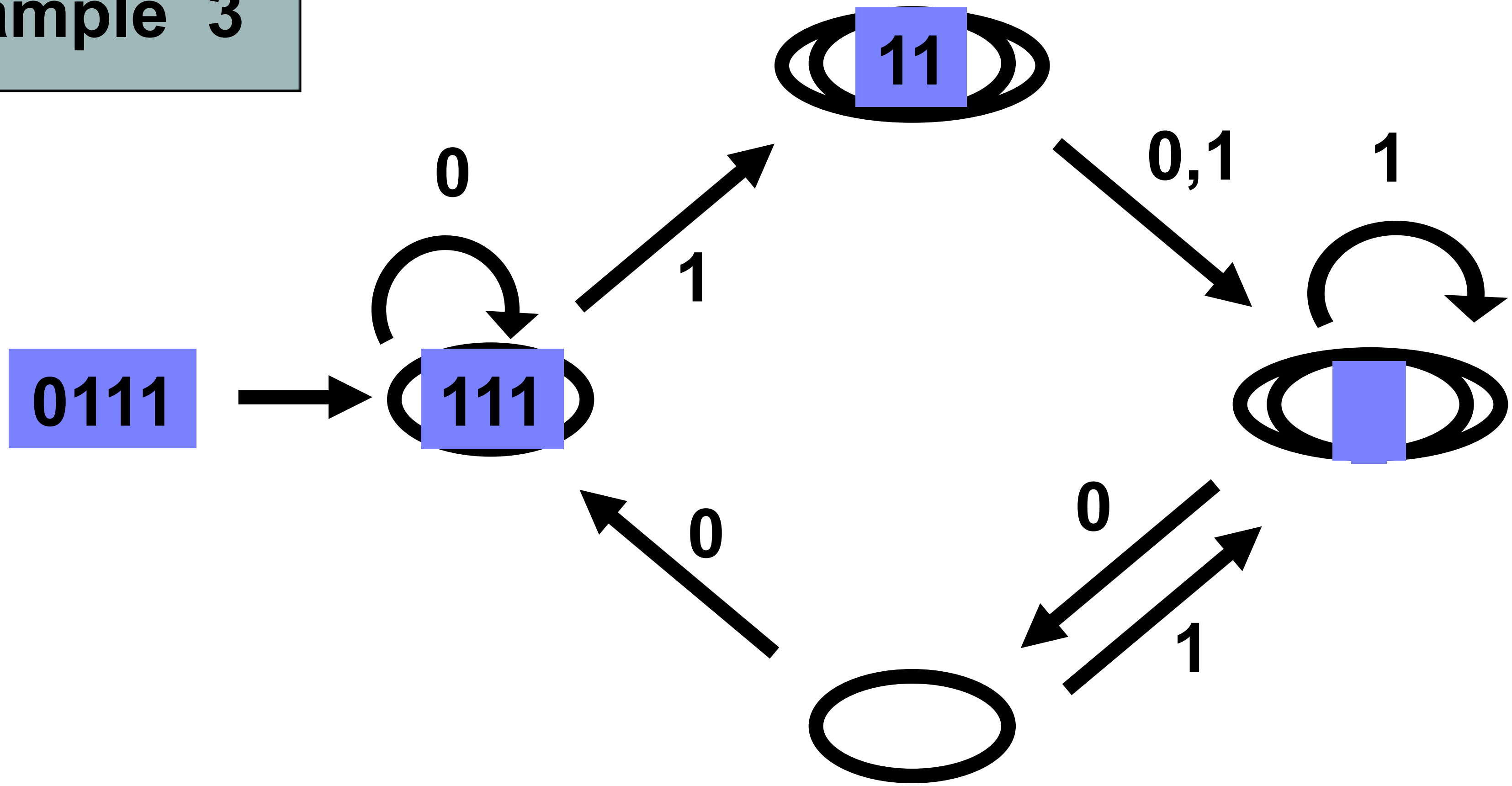


Input: D | Q | Q | D

45 is final state

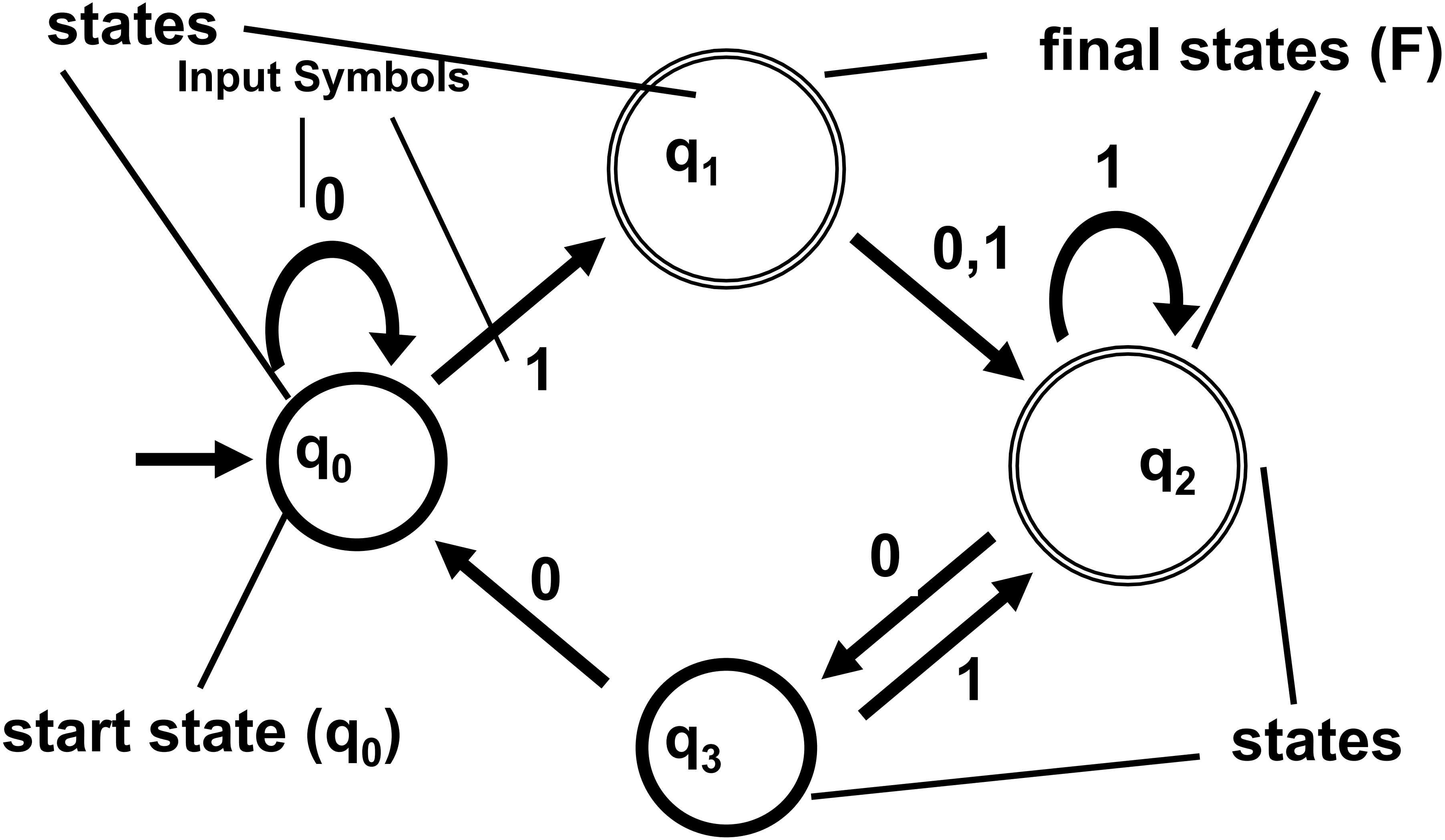
Input Accepted

Example 3



The machine accepts a string if the process ends in a *final state*

Example 3


states
Input Symbols
0
1
q1
final states (F)
1
0,1
q0
q2
0
0
1
start state (q0)
q3
states

Definitions

An *alphabet* Σ is a finite set of symbols (in Ex3, Σ = {0,1})

A *string* over Σ is a finite sequence of elements of Σ (e.g. 0111)

For a string s, |s| is the *length* of s

The unique string of length 0 will be denoted by λ and will be called the *empty string*

The ***reversal*** of a string *u* is denoted by u^R.
Example: $(\text{banana})^R$ = ananab

Definitions

The *concatenation* of two strings is the string resulting from putting them together from left to right. Given strings *u* and *v*, denote the concatenation by *u* .*v*, or just *uv.*

Example:

jap . an = japan, QQ . DD = QQDD

Q1: What's the Java equivalent of concatenation?

The + operator on strings

Q2: Find a formula for $|u\,.v\,|$?

$|u\,.v\,| = |u\,|+|v\,|$

Definitions

If Σ is an alphabet,
Σ * denotes the set of all strings over Σ.

A *language* over Σ is a subset of Σ *

i.e. a set of strings *each* consisting of sequences of symbols in Σ.

Examples

Example1: in our vending machine we have

$\Sigma = \{$ **D, Q** $\}$

$\Sigma^* = \{\lambda,$

D, Q,

DD, DQ, QD, QQ,

DDD, DDQ, DQD, DQQ, QDD, QDQ, QQD, QQQ,

DDDD, DDDQ, … $\}$

$L = \{ u \in \Sigma^* \mid u$ **successfully vends** $\}$

Example2: in our switch example we have

$\Sigma = \{$ **Push** $\}$

$\Sigma^* = \{\lambda,$

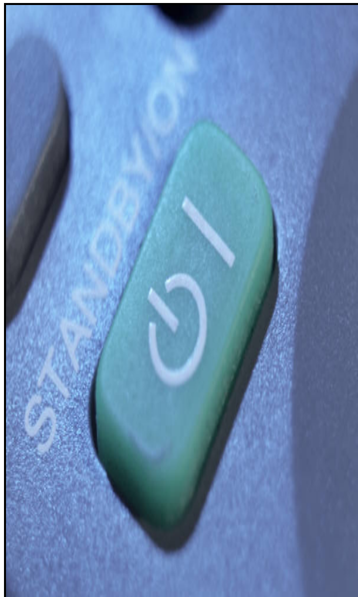
Push,

Push Push,

Push Push Push,

Push Push Push Push, … $\}$

$L = \{ \text{Push}^n \mid n \text{ is odd} \}$



Definitions

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

Q is the set of states

Σ is the alphabet

δ is the transition function

$q_0 \in Q$ is the start state

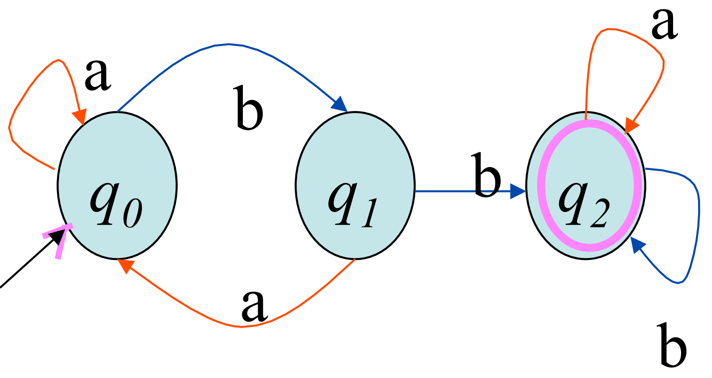
$F \subseteq Q$ is the set of final states

$L(M)$ = the language of machine M

= set of all strings machine M accepts

Definitions

State Diagram and Table



δ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2

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

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

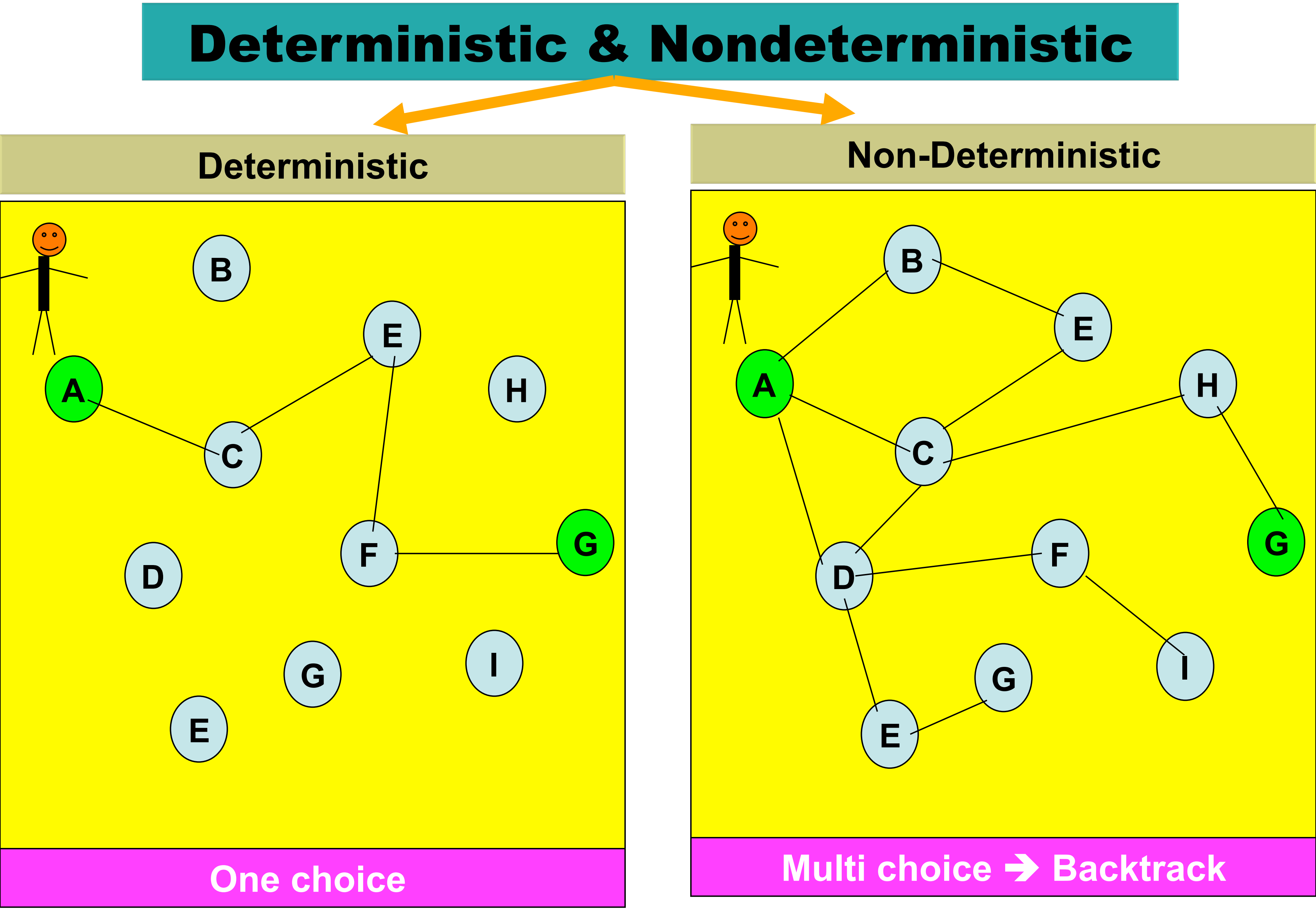
$$F = \{q_2\}$$

FINITE STATE MACHINES (AUTOMATA)

Deterministic Finite Automata (DFA)

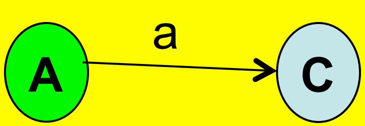
Non-Deterministic Finite Automata (NFA)

Non-Deterministic Finite Automata with empty move (λ-NFA)


Deterministic & Nondeterministic
Deterministic
Non-Deterministic
B
E
A
C
H
D
F
G
G
I
E
One choice
B
E
A
H
C
G
D
F
G
I
E
Multi choice → Backtrack

Deterministic & Nondeterministic

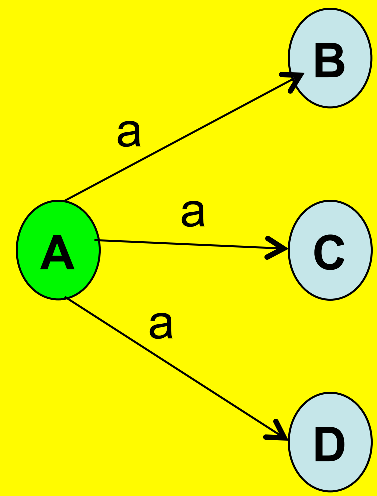
Deterministic



From ONE state machine can go to another ONE state on one input

One choice

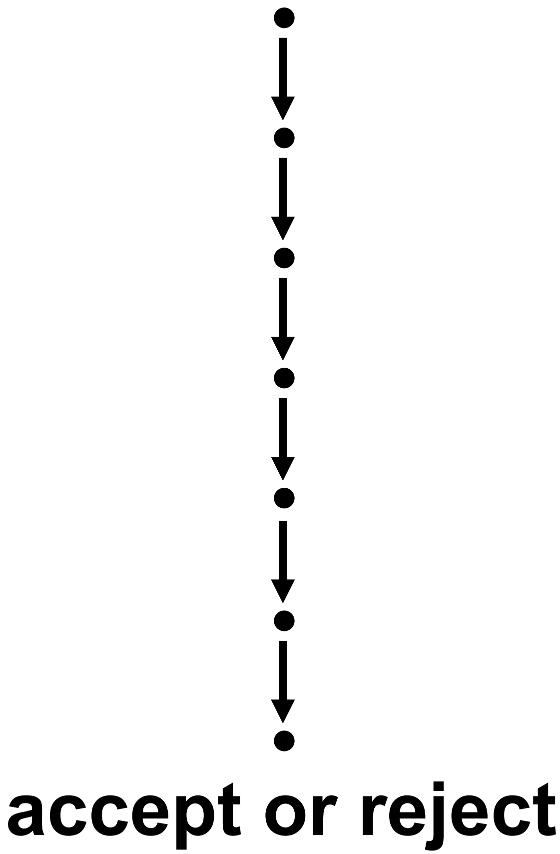
Non-Deterministic



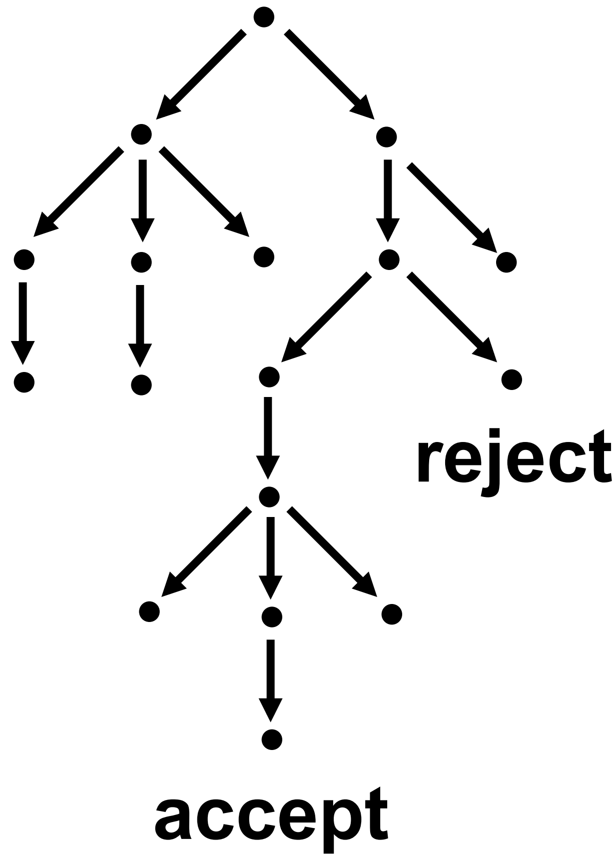
From ONE state machine can go to MANY states on one input

Multi choice

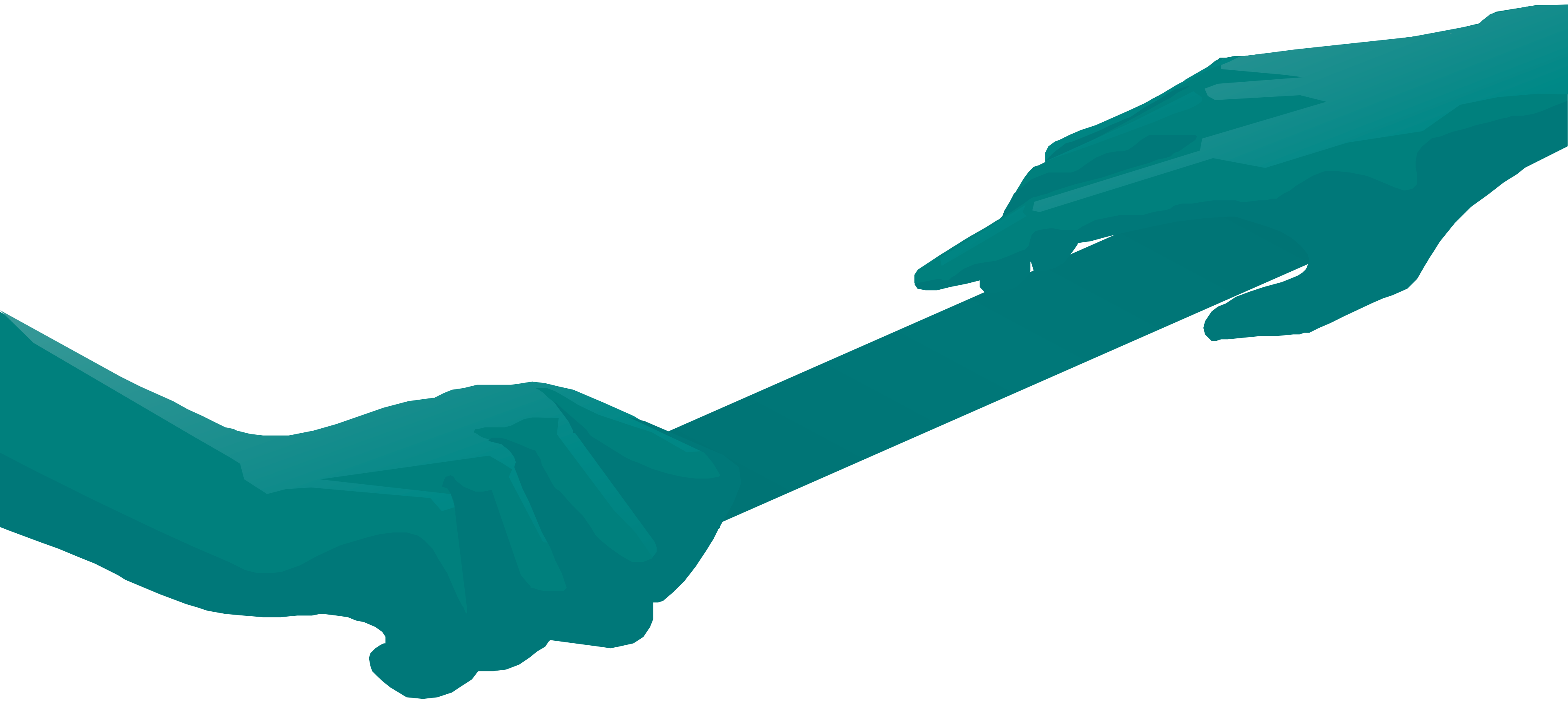
Deterministic Computation



Non-Deterministic Computation



DETERMINISTIC FINITE AUTOMATA (DFA)



Definitions

A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$

Q is the set of states

Σ is the alphabet

$\delta : Q \times \Sigma \rightarrow Q$ is the transition function

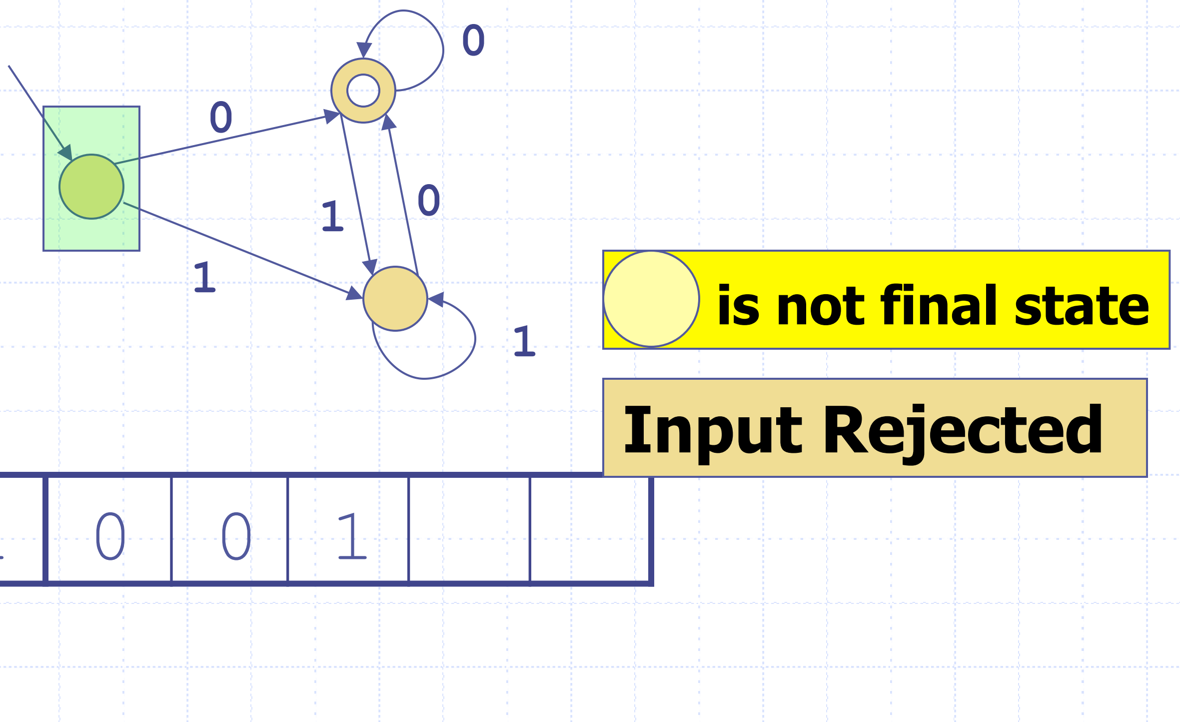
$q_0 \in Q$ is the start state

$F \subseteq Q$ is the set of accept states

$L(M)$ = the language of machine M
= set of all strings machine M accepts

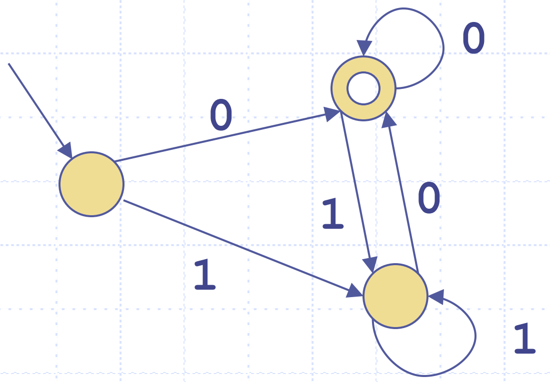
Deterministic Finite Automata (DFA)

Example 1



Deterministic Finite Automata (DFA)

Example 1

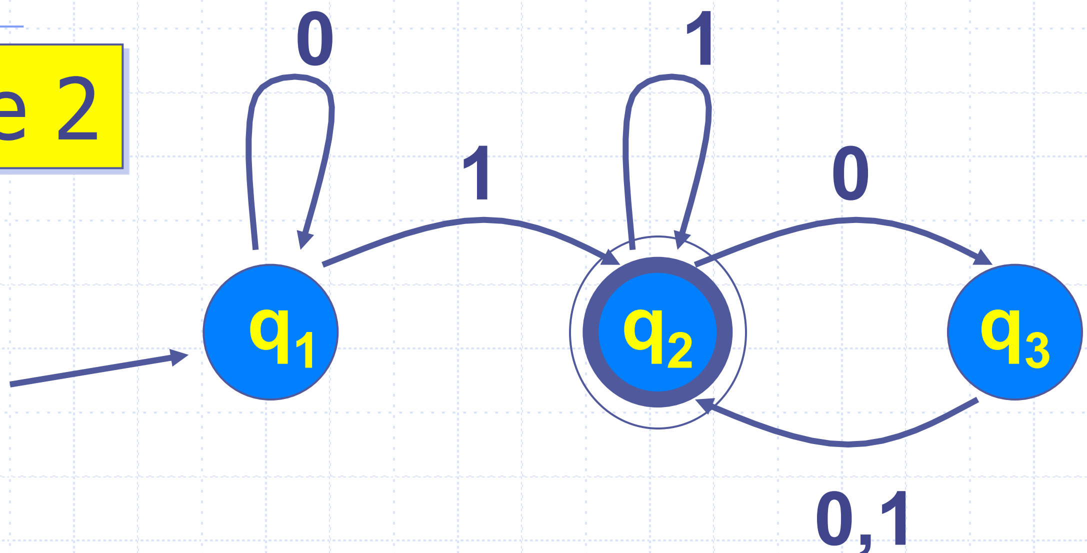


Q: **What kinds of bit-strings are accepted?**

A: Bit-strings that represent binary even numbers.

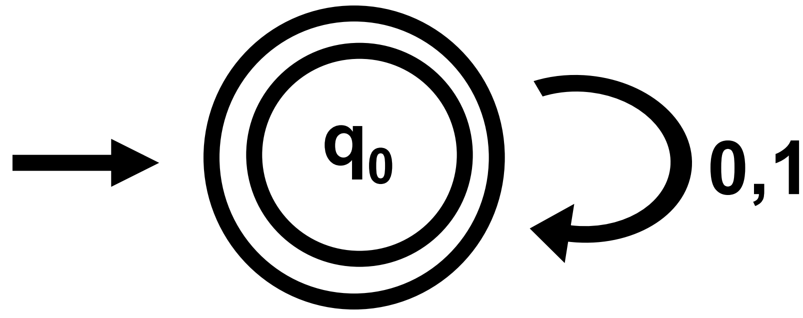
Deterministic Finite Automata (DFA)

Example 2



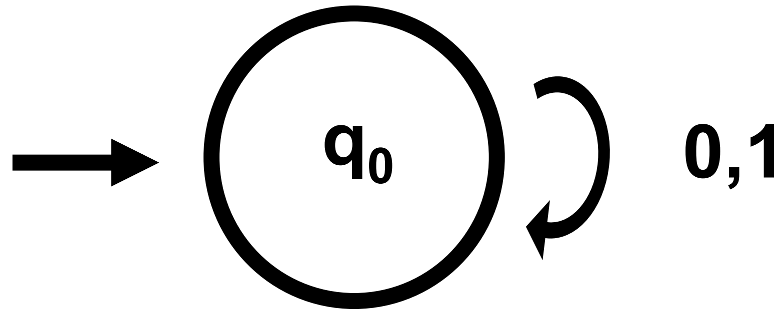
010	reject
11	accept
010100100100100	accept
010000010010	reject
λ	reject

Exercise



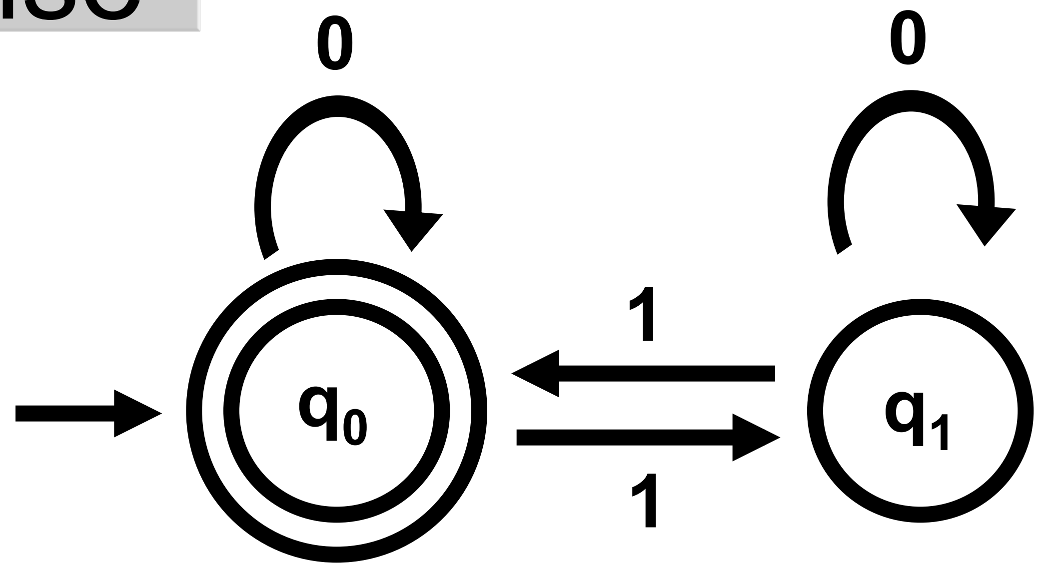
L(M) = {0,1}*

Exercise



L(M) = ∅

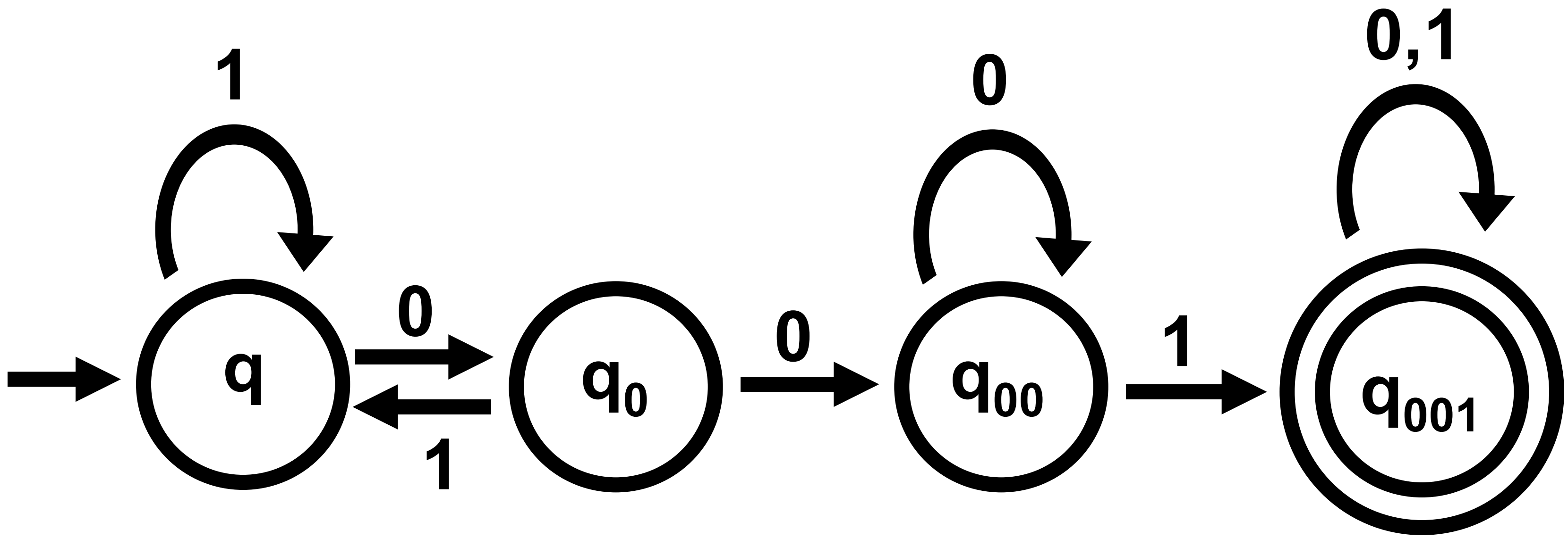
Exercise



L(M) = { w | w has an even number of 1s}

Exercise

Build an automaton that accepts all and only those strings that contain 001



Exercise

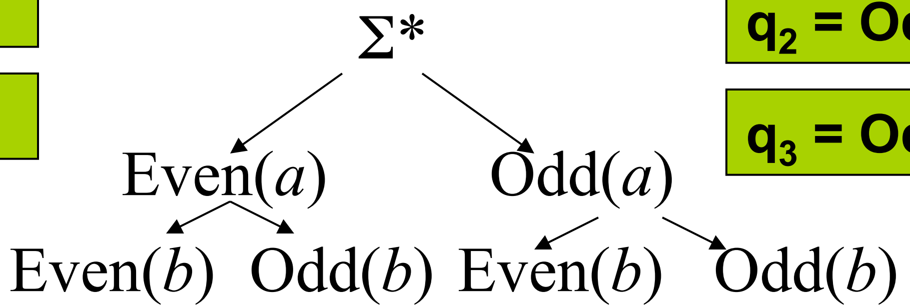
Strings over {*a*,*b*} containing even number of *a*' s and odd number of *b*' s.

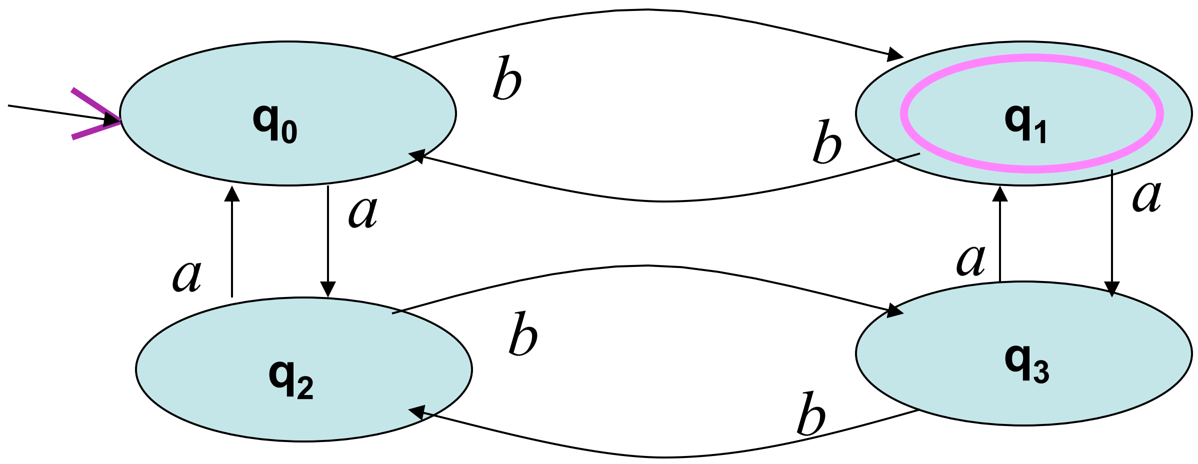
q_0 = Even(*a*).Even(*b*)

q_1 = Even(*a*).Odd(*b*)

q_2 = Odd(*a*).Even(*b*)

q_3 = Odd(*a*).Odd(*b*)





Exercise

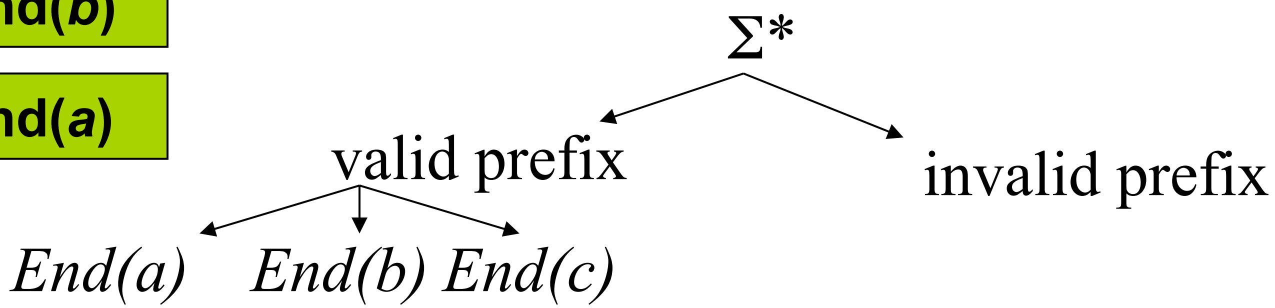
Strings over {*a*,*b*,*c*} that has the form (ab)*c

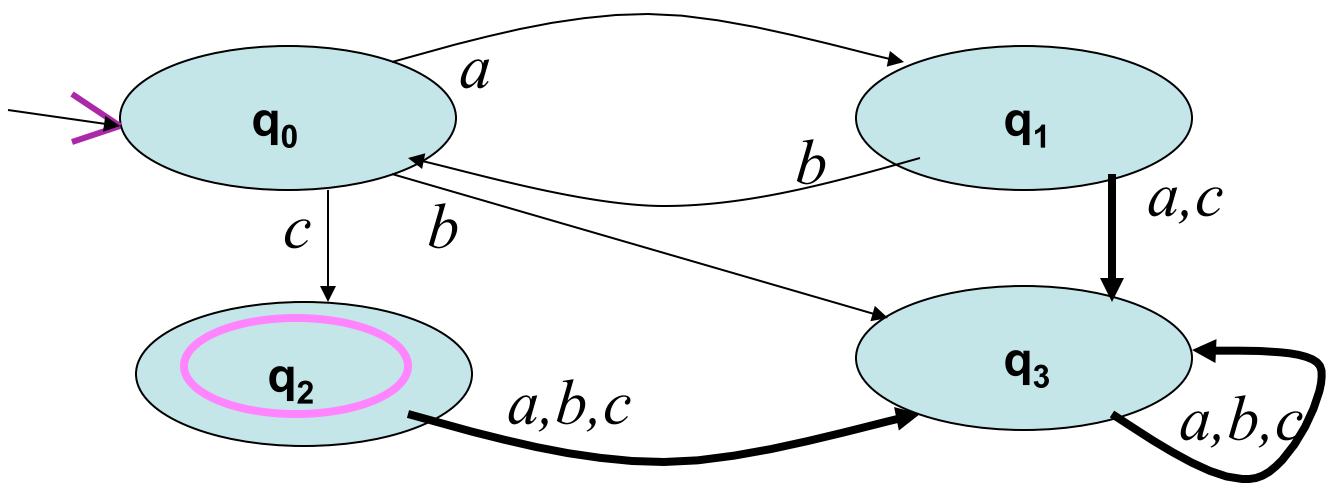
q_0 = End(*b*)

q_1 = End(*a*)

q_2 = End(*c*)

q_3 = Error





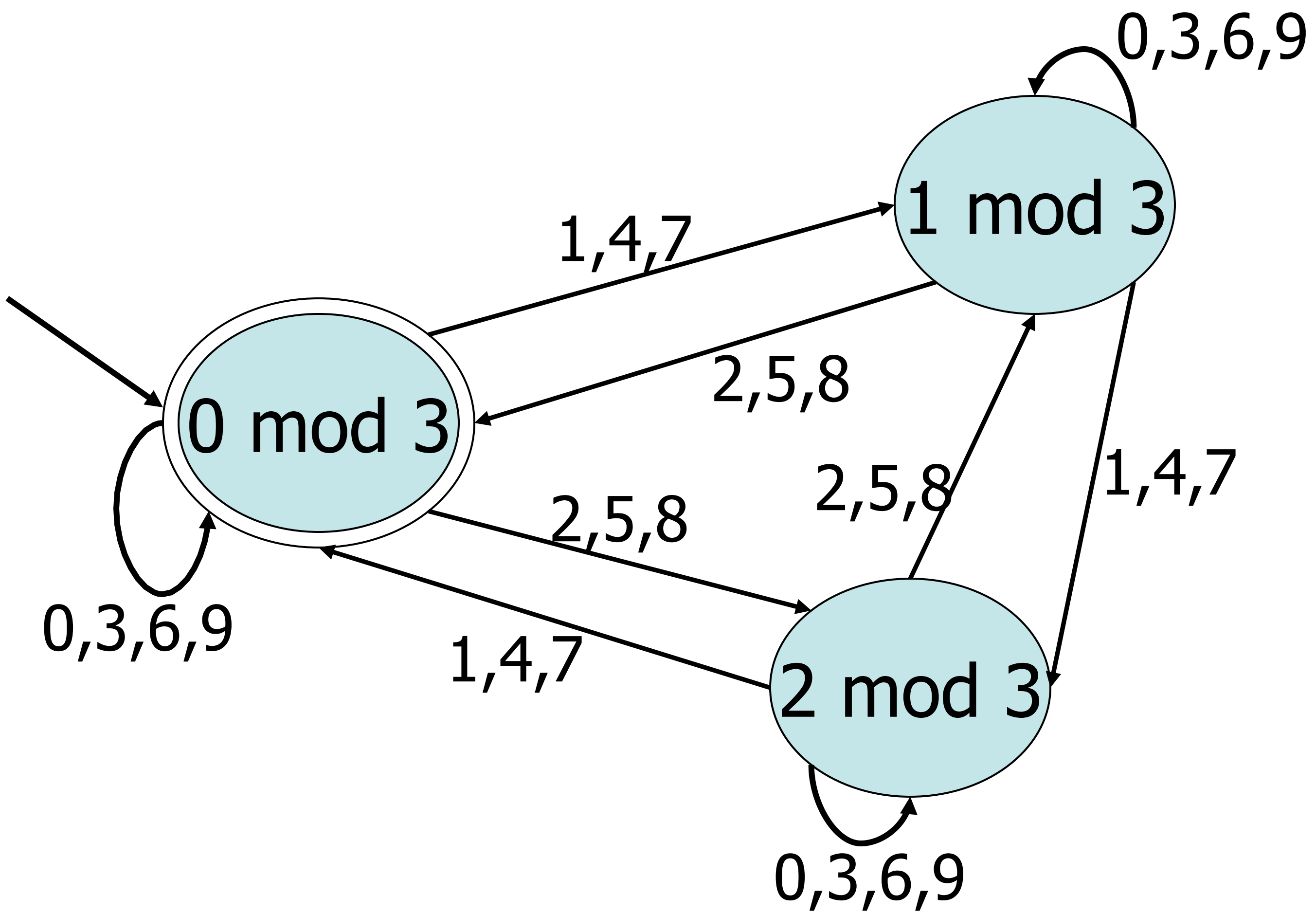
Exercise

Design with a friend a machine that tells us when a *base*-10 number is divisible by 3.

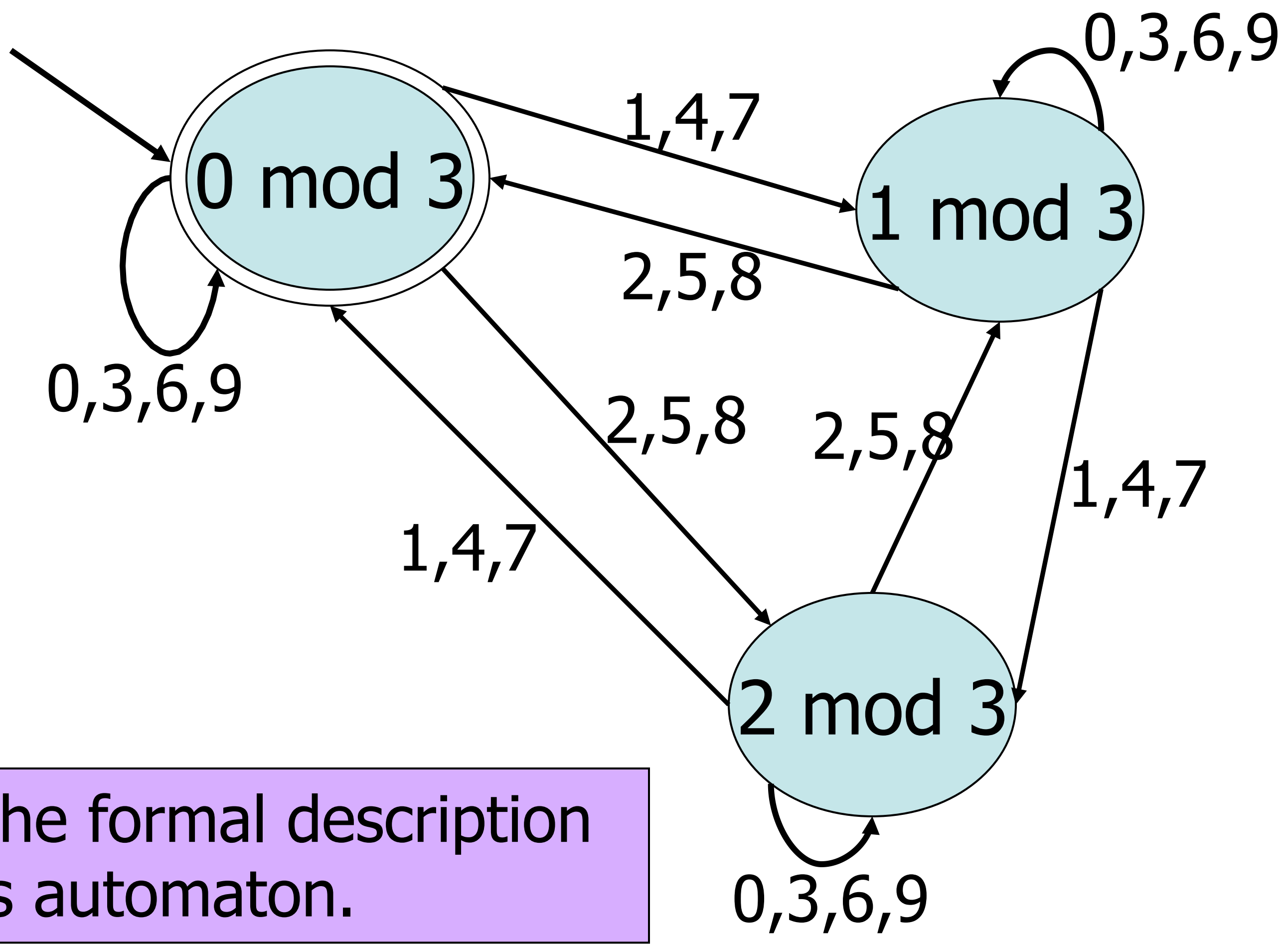
What should your alphabet be?

How can you tell when a number is divisible by 3?

Answer



Exercise



Find the formal description of this automaton.

Answer

Q = { 0 mod 3, 1 mod 3, 2 mod 3 } (rename: {q_0, q_1, q_2})

Σ = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

q_0 = 0 mod 3

F = { 0 mod 3 }

$$\delta : Q \times \Sigma \rightarrow Q$$

$$\delta(q_0,2) = q_2,\ \delta(q_0,9) = q_0, \delta(q_1,2) = q_0,$$

$$\delta(q_1,7) = q_2,\ \delta(q_2,3) = q_2, \delta(q_2,5) = q_1.$$

$$\text{Question} : \delta(q_i, j) = ?$$

$$\delta(q_i, j) = q_{(i+j) \bmod 3}$$