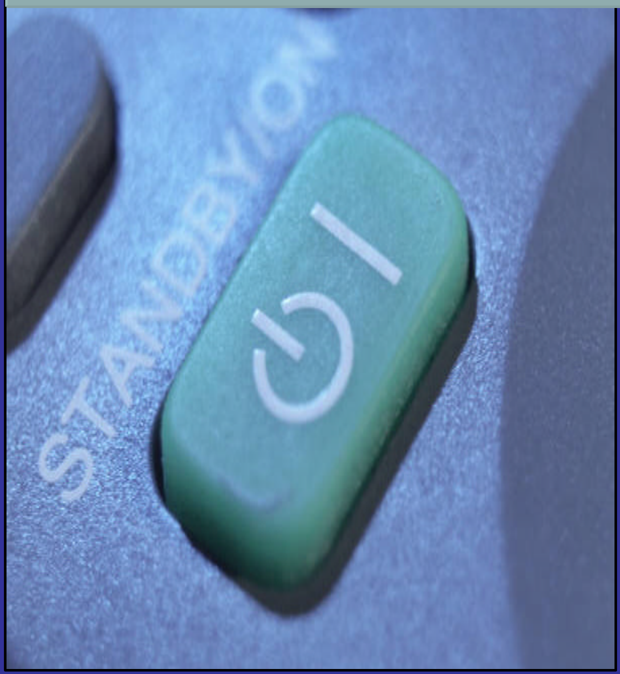
FINITE STATE MACHINES (AUTOMATA)

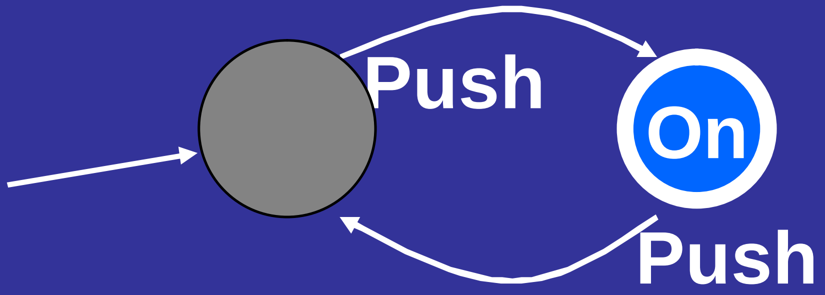
Switch Example 1

Think about the On/Off button

Switch Example 1



The corresponding Automaton

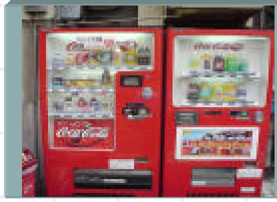


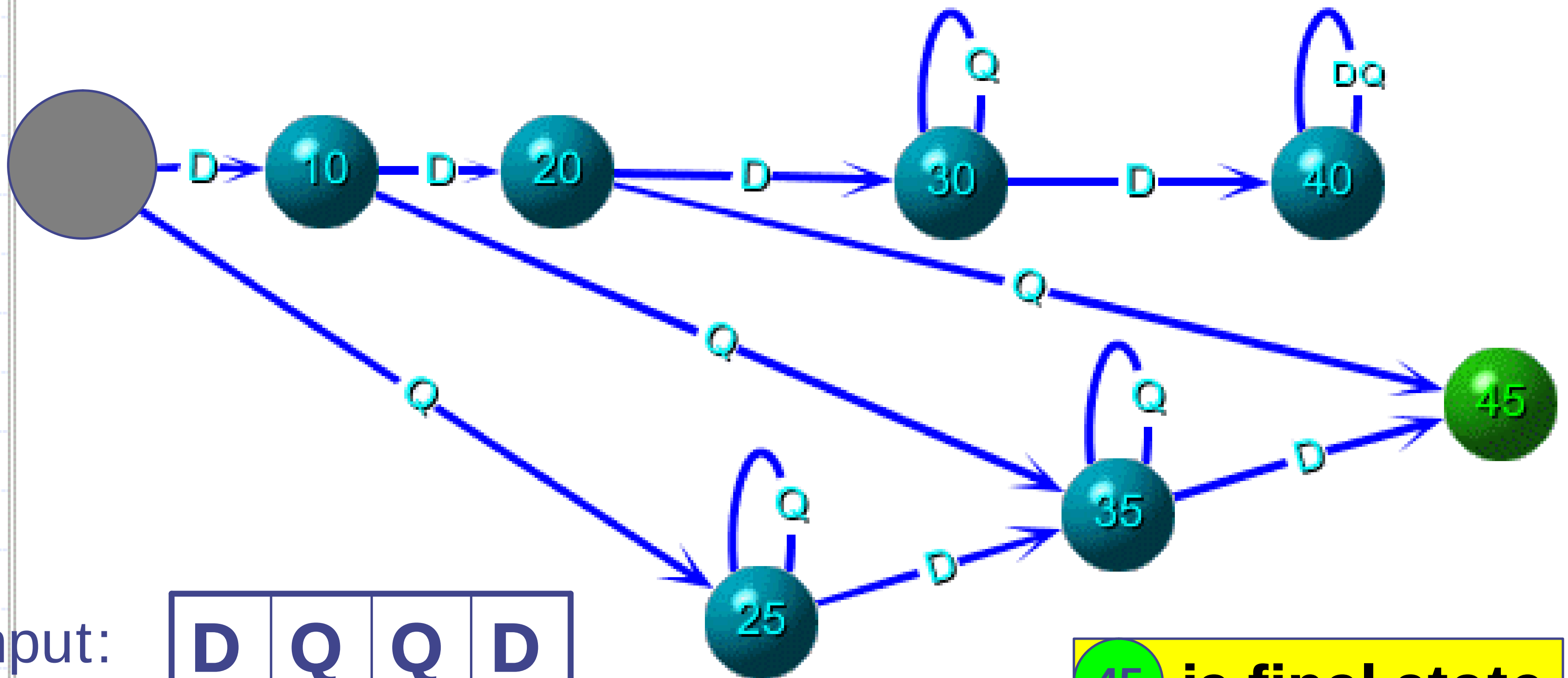
Input:

Push	Push	Push	Push

Vending Machine Example 2

- Vending machine dispenses Cola for $0.45

 Vending Machine Example 2

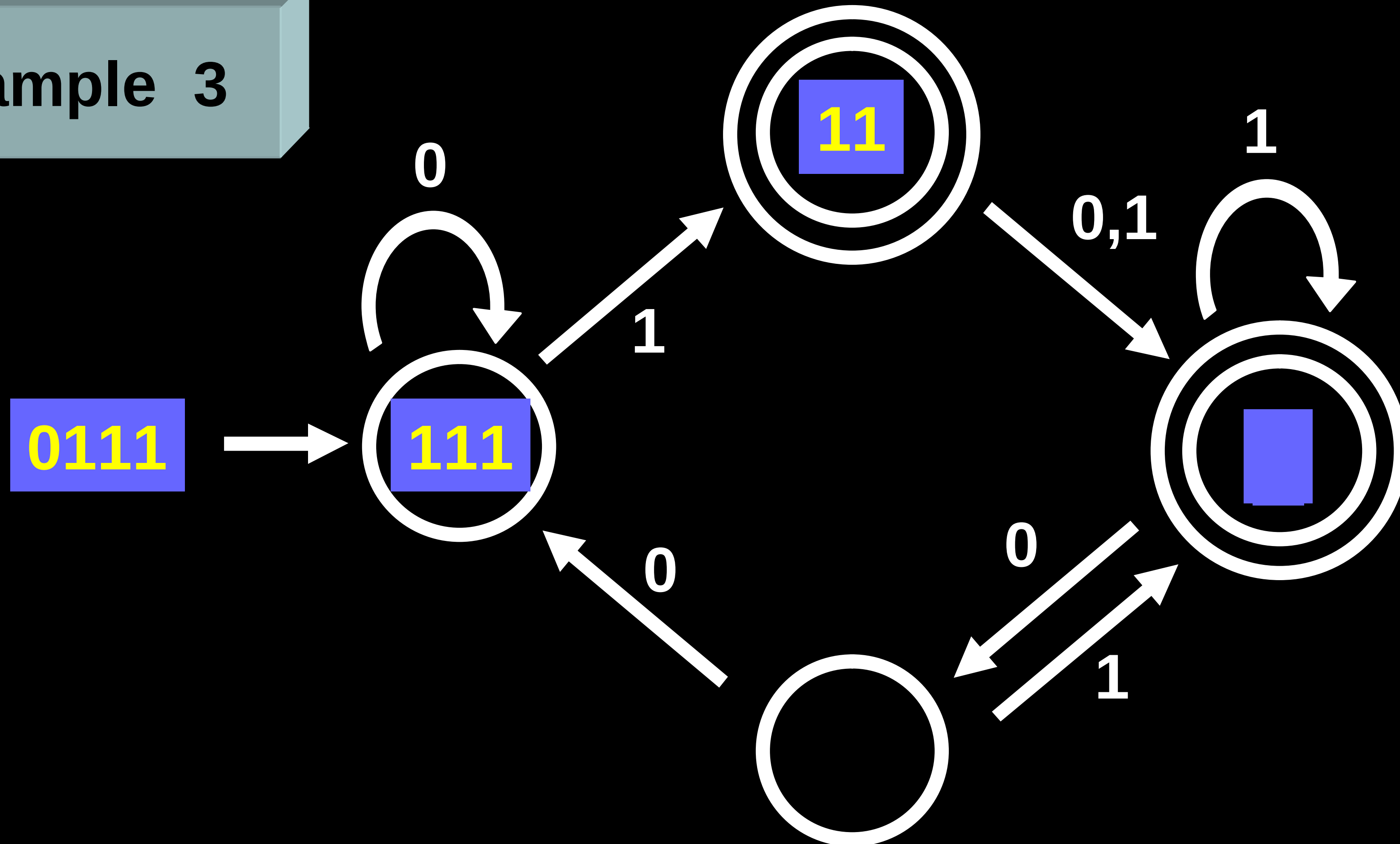


Input: | D | Q | Q | D |

45 **is final state**

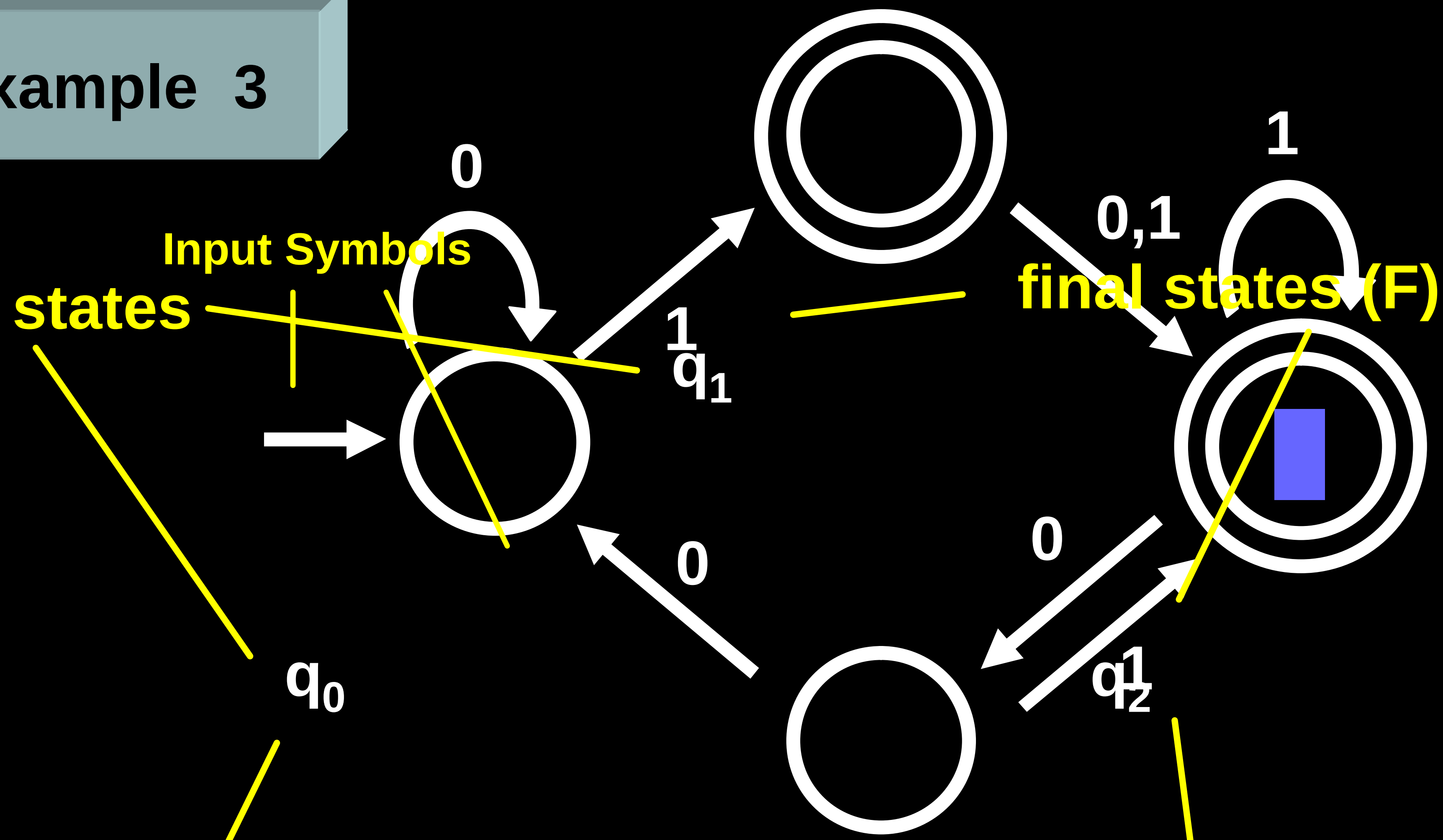
Input Accepted

Example 3



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

Example 3



The machine **accepts** a string if the process ends in a q_3 *final state*

start state (q_0)

states

Definitions

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

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

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

The unique string of length 0 will be denoted by ? 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 S is an alphabet,

S * denotes the set of all strings over S.

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

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

Examples

Example1: in our vending machine we have

S = { **D, Q** }

S* = {l**,**

D, Q,

DD, DQ, QD, QQ,

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

DDDD, DDDQ, … }

L* =** { ***u Î S* **|** ***u* successfully vends** }

Example2: in our switch example we have

S = { **Push**}

S* = {l,

Push,

Push Push,

Push Push Push,

Push Push Push Push, … }

***L* =** { Push^{n} | n is odd }

Definitions

A finite automaton is a 5-tuple $M = (Q, S, d, q_0, F)$

Q is the set of states

S is the alphabet

d is the transition function

q_0 Î Q is the start state

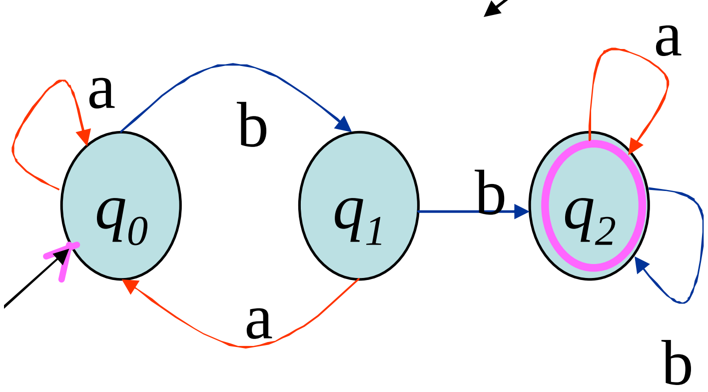
F Í Q is the set of final states

L(M) = the language of machine M

= set of all strings machine M accepts

Definitions

State Diagram and **Table**

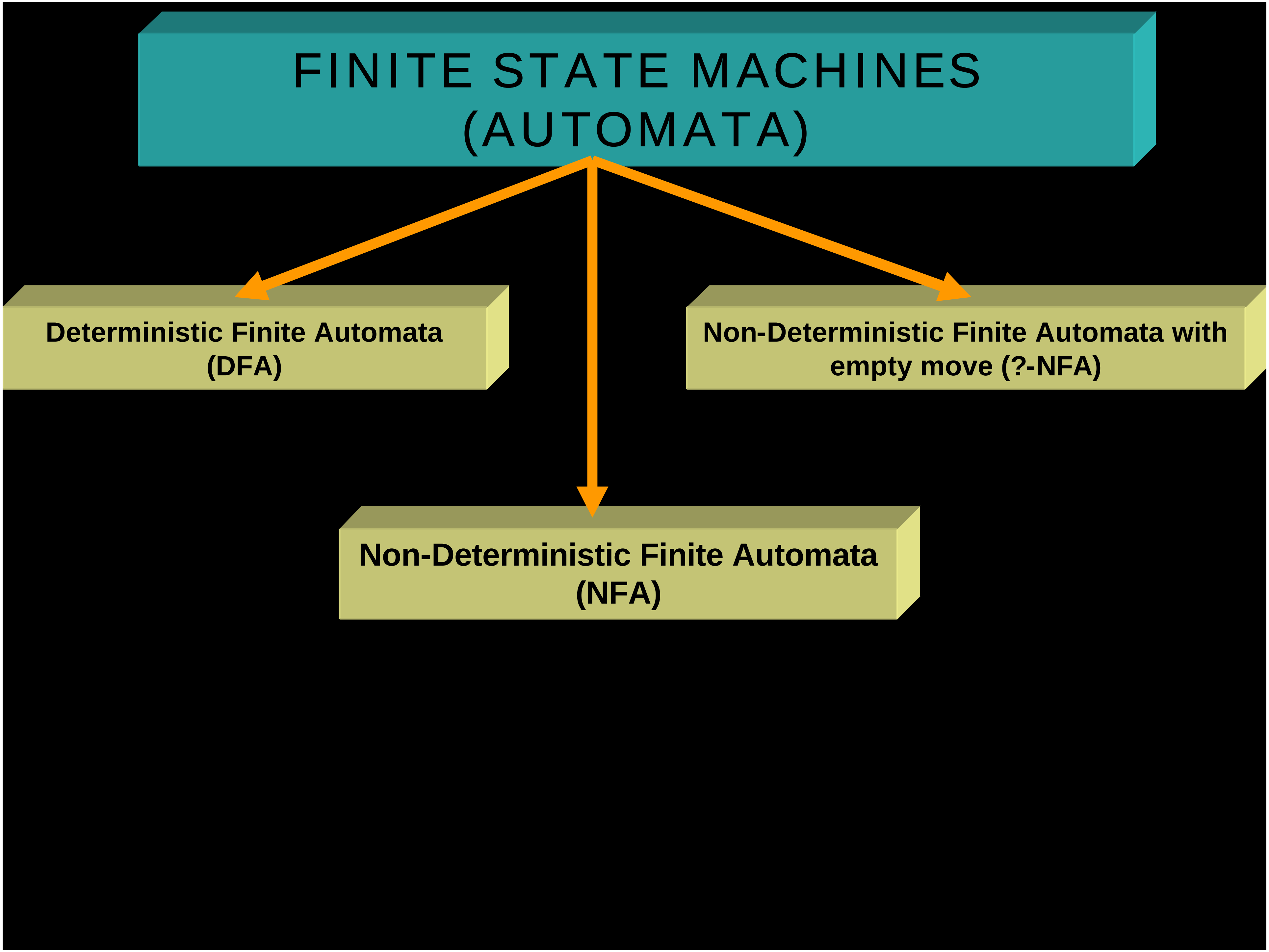


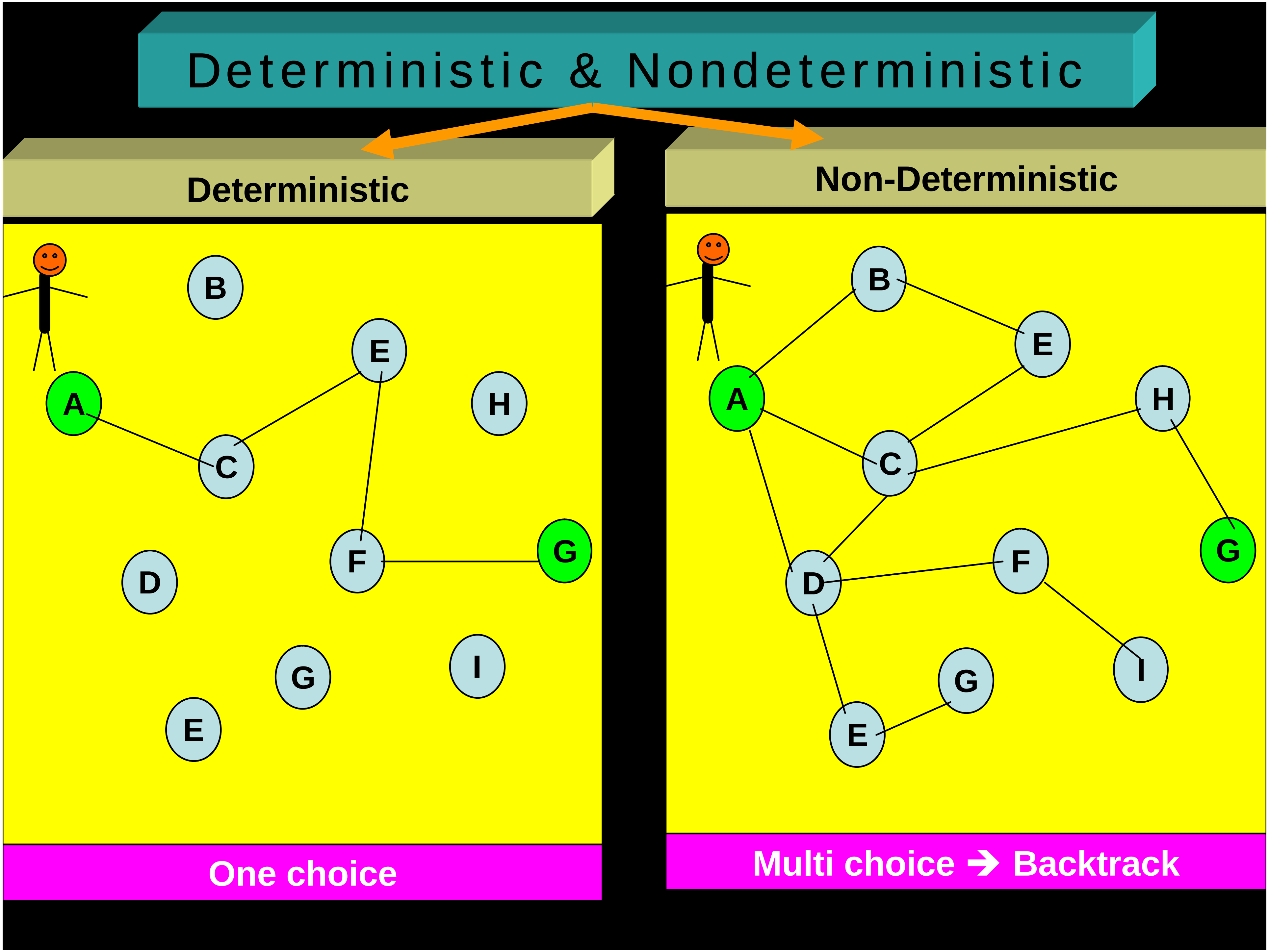
d	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2

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

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

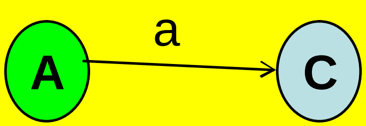
$$F = \{q_2\}$$


FINITE STATE MACHINES
(AUTOMATA)
Deterministic Finite Automata
(DFA)
Non-Deterministic Finite Automata with
empty move (?-NFA)
Non-Deterministic Finite Automata
(NFA)


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

Deterministic & Nondeterministic

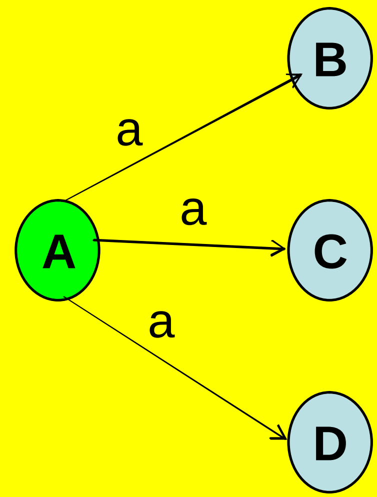
Deterministic



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

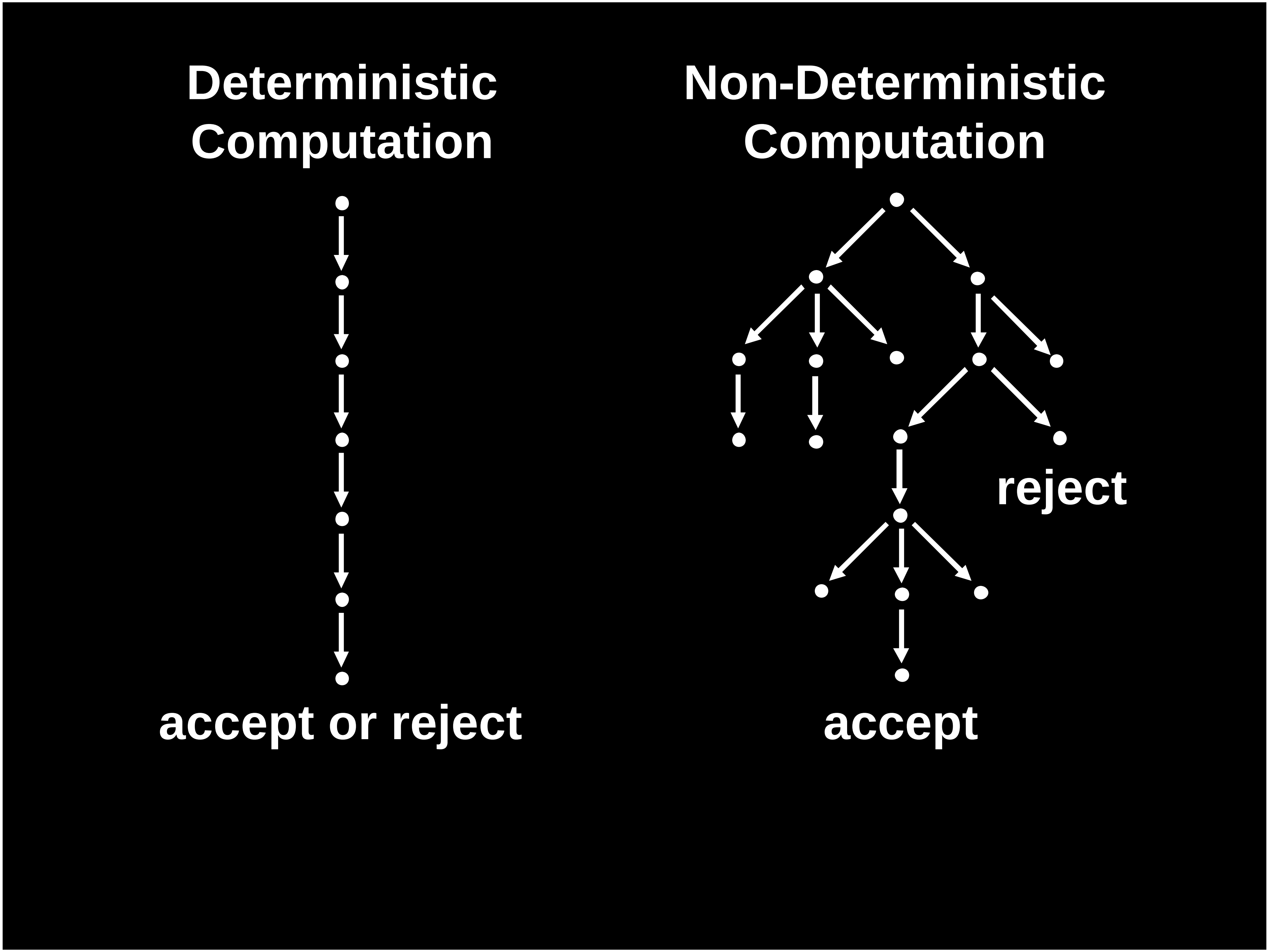
One choice

Non-Deterministic

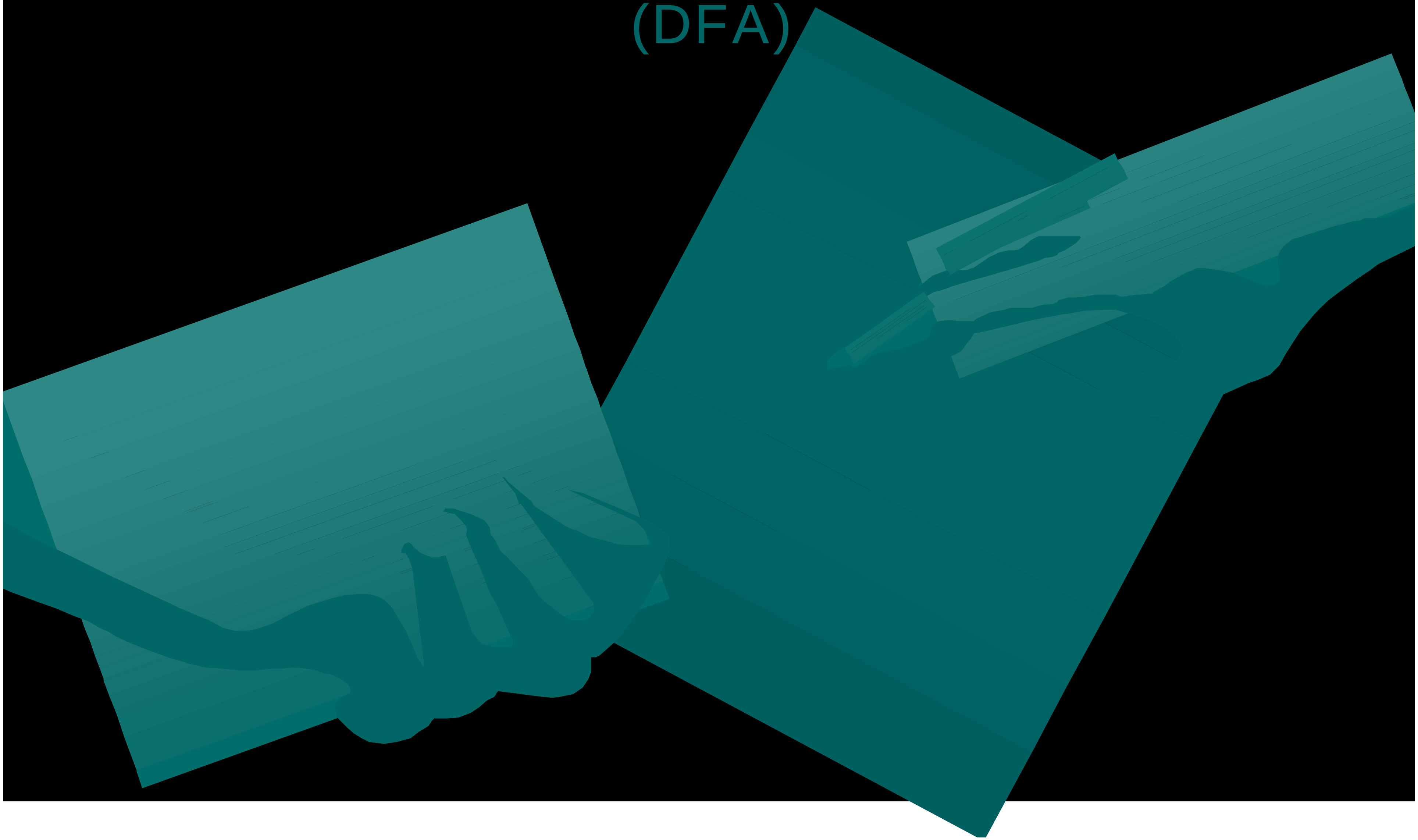


From ONE state machine can go to MANY states on one input

Multi choice


Deterministic Computation
Non-Deterministic Computation
reject
accept or reject
accept

DETERMINISTIC FINITE AUTOMATA (DFA)



Definitions

A DFA is a 5-tuple M = (Q, S, d, q_0, F)

Q is the set of states

S is the alphabet

d : Q ´ S ? Q is the transition function

q_0 Î Q is the start state

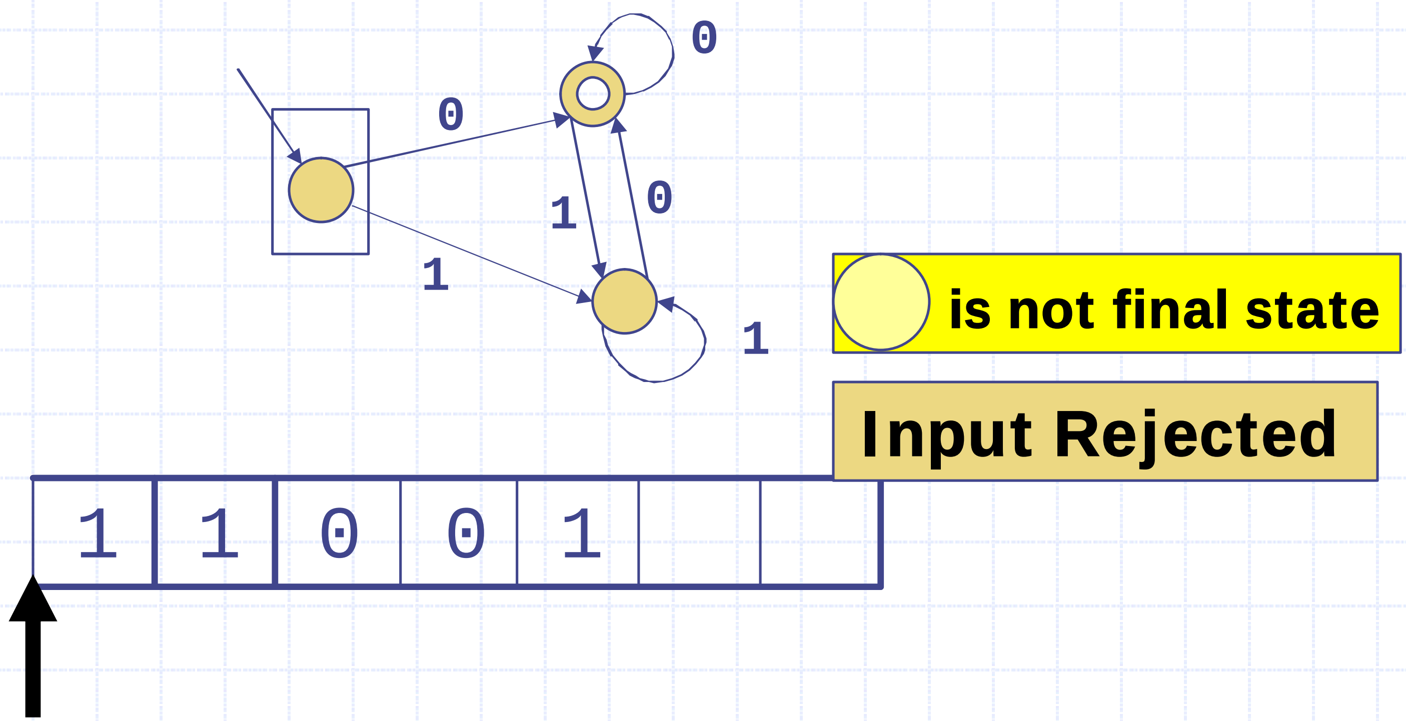
F Í Q is the set of accept states

L(M) = the language of machine M

= set of all strings machine M accepts

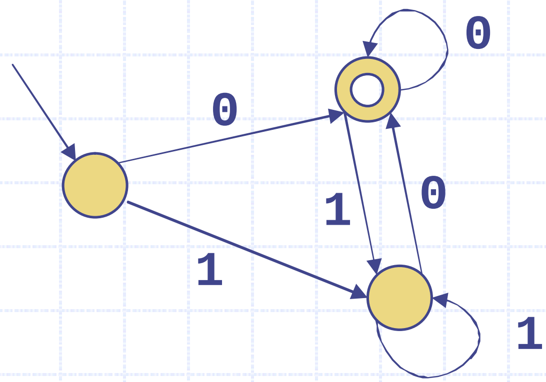
Deterministic Finite Automata (DFA)

Example 1



Deterministic Finite Automata (DFA)

Example 1

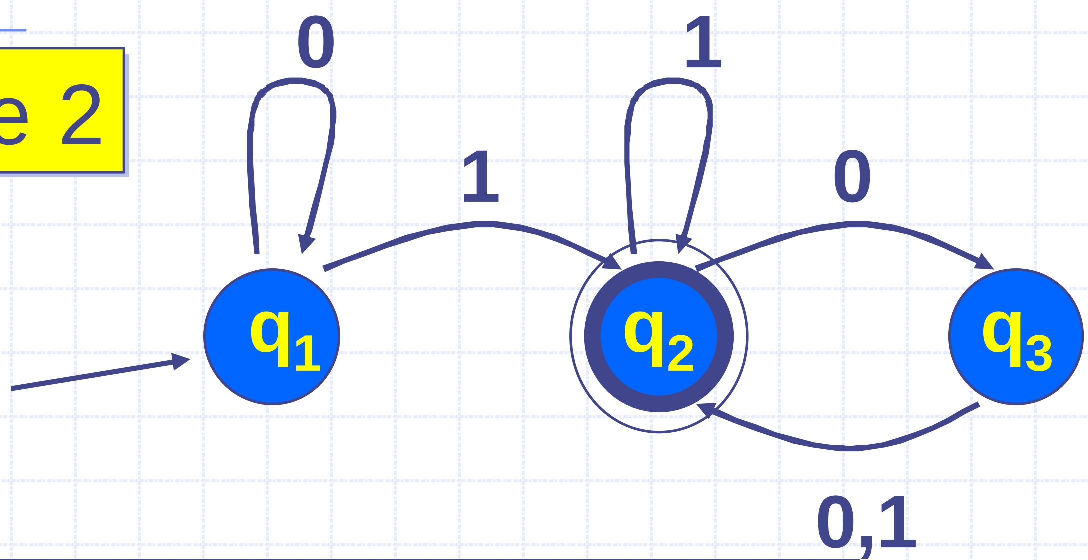


Q: **What kinds of bit-strings are accepted?**

A: Bit-strings that represent binary even numbers.

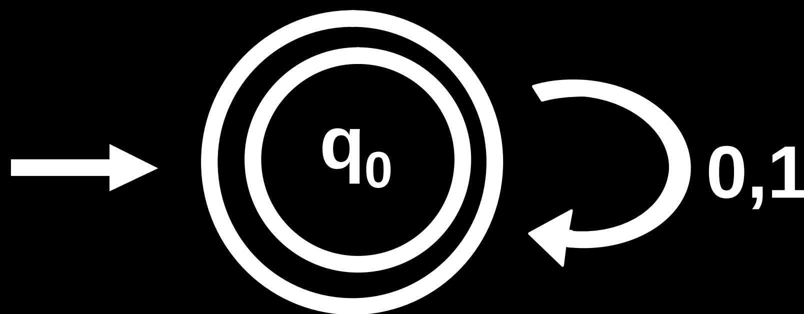
Deterministic Finite Automata (DFA)

Example 2



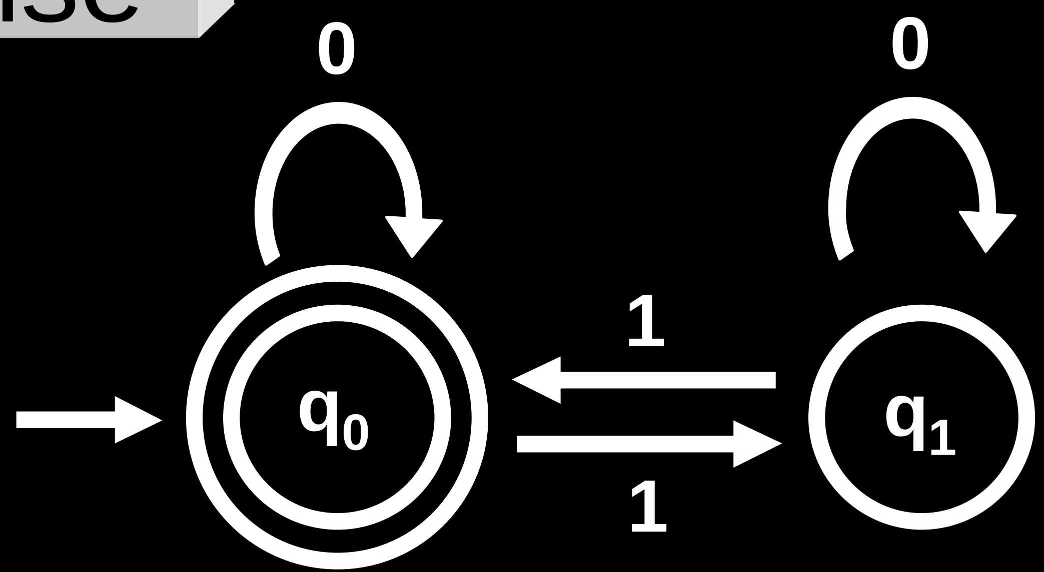
010	reject
11	accept
010100100100100	accept
010000010010	reject
]	reject

Exercise



$L(M) = \{0,1\}^*$

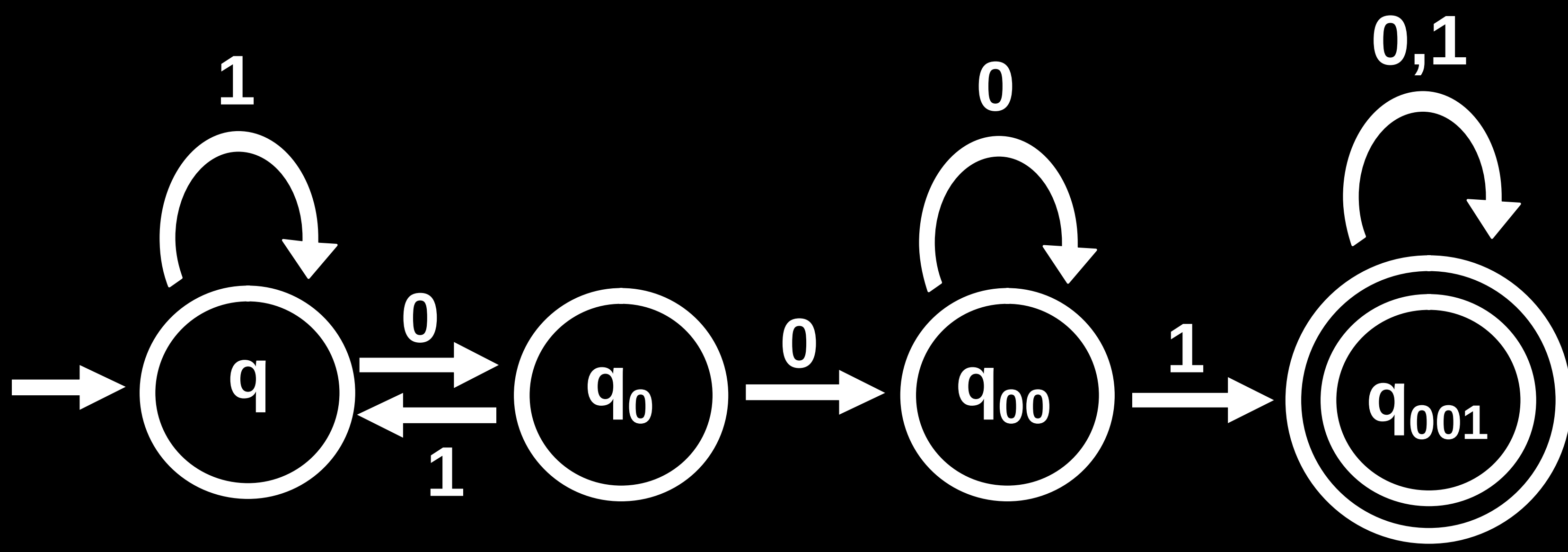
Exercise



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

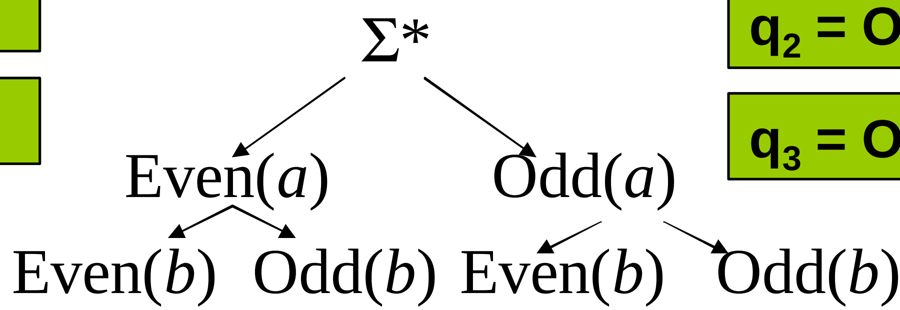
Strings over {*a*,*b*} containing even number of *a*'s and odd number of *b*'s.

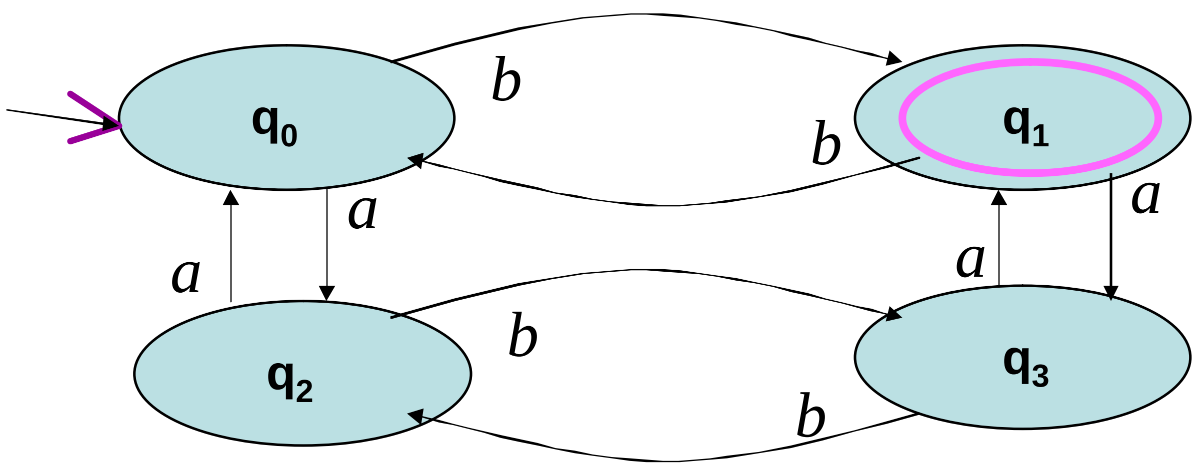
q_0 = Even(*a*).Even(*b*)

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

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

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





Exercise

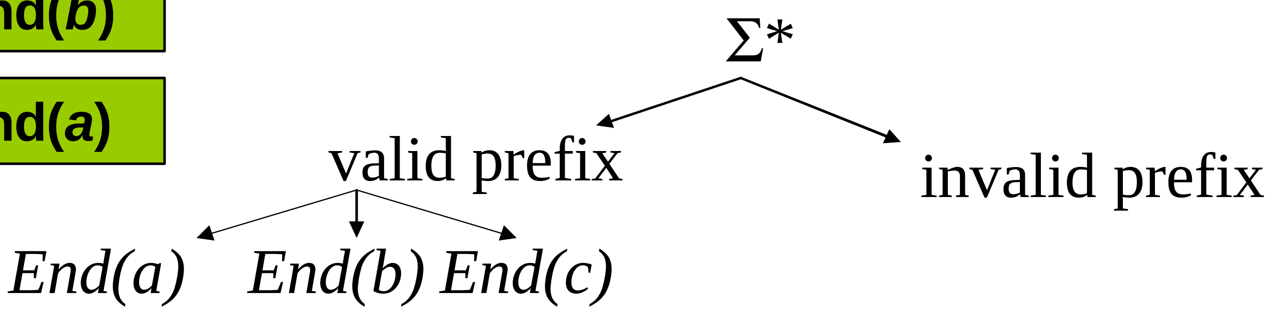
Strings over {*a*,*b*,*c*} that has the form (ab)*c

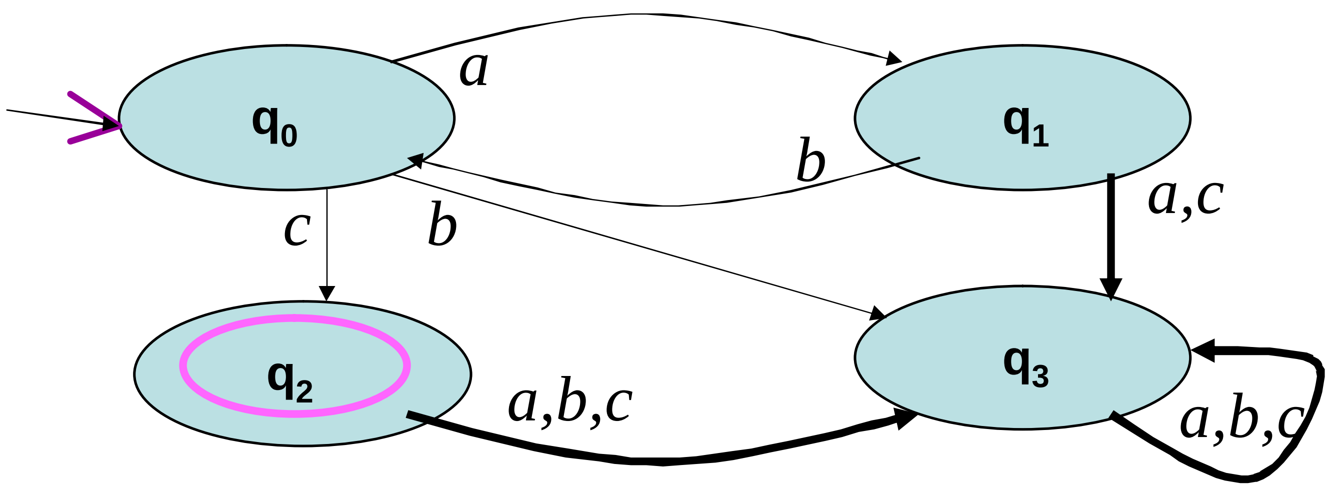
q_0 = End(*b*)

q_1 = End(*a*)

q_2 = End(*c*)

q_3 = Error





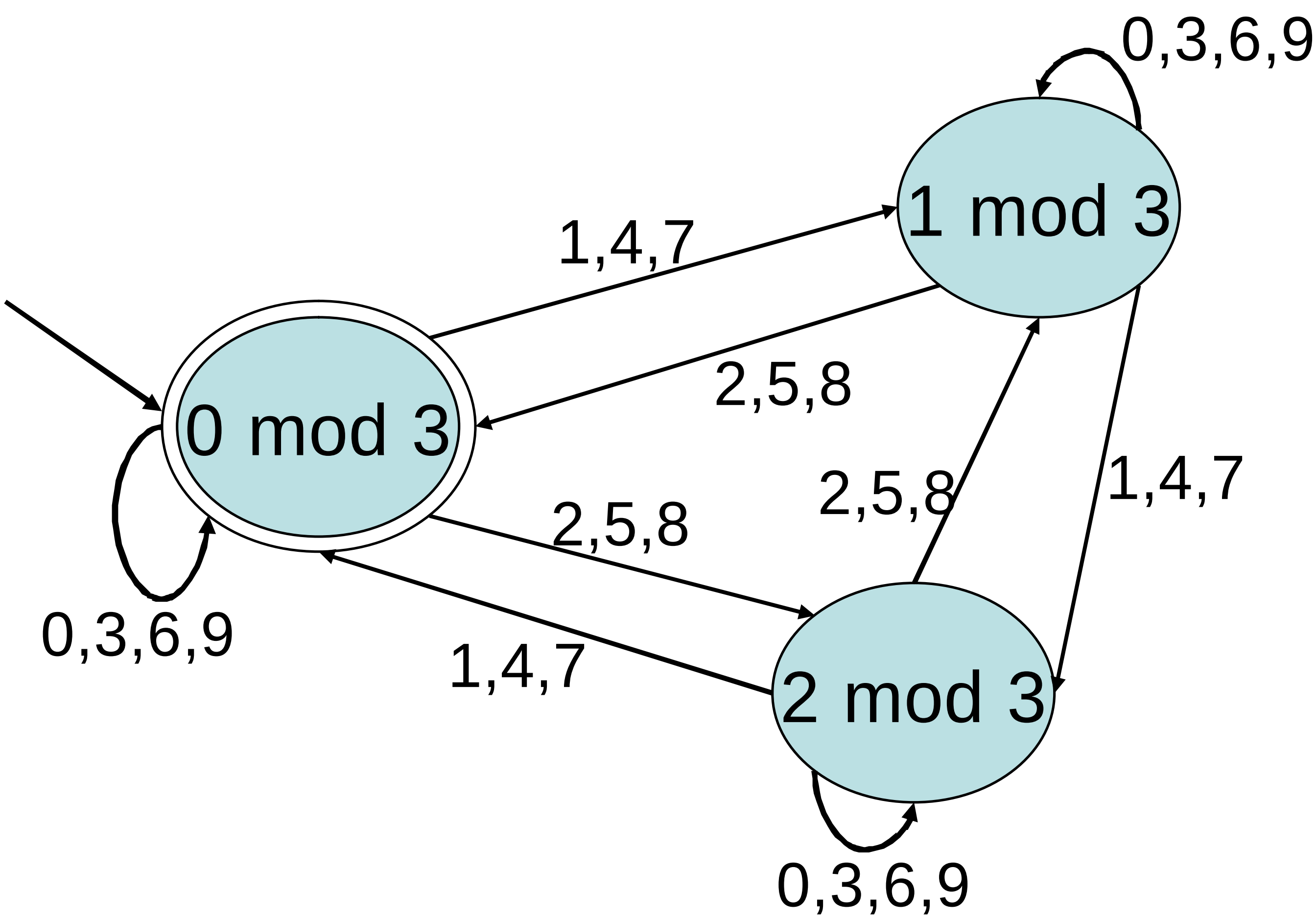
Exercise

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

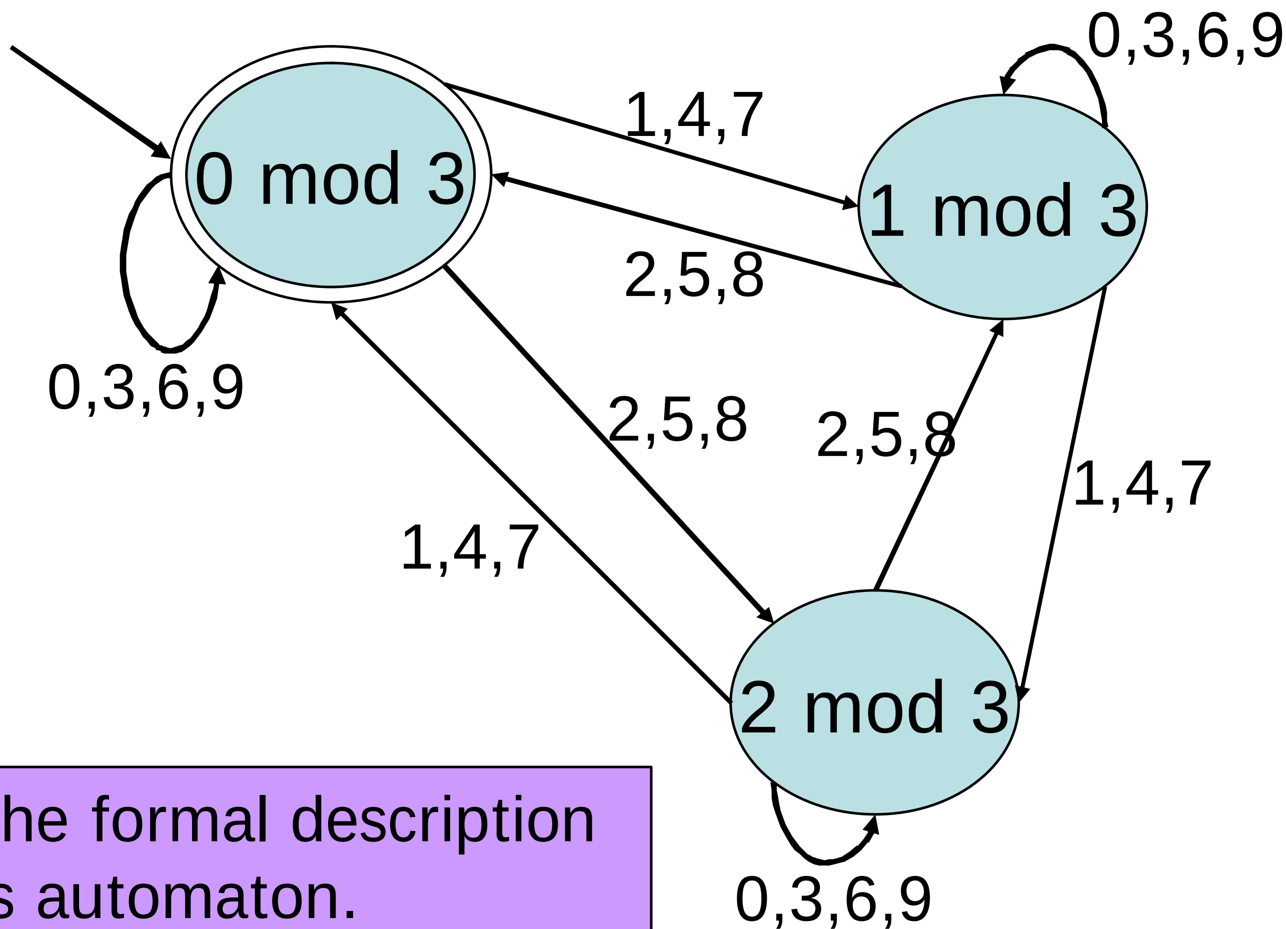
What should your alphabet be?

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 }

$$\mathrm{d} : Q \times \Sigma \rightarrow Q$$

$$\mathrm{d}(q_0, 2) = q_2, \quad \mathrm{d}(q_0, 9) = q_0, \mathrm{d}(q_1, 2) = q_0,$$

$$\mathrm{d}(q_1, 7) = q_2, \quad \mathrm{d}(q_2, 3) = q_2, \mathrm{d}(q_2, 5) = q_1.$$

Question : $\mathrm{d}(q_i, j) = ?$

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