FINITE AUTOMATA

COMP 4200 – Formal Language

TERMINOLOGIES

ALPHABETS AND STRINGS

- An alphabet is any finite set of characters.
  - Examples: {0,1}, {a,b,c}, {0,1,#}, {a,….z,A,…..Z}
  - Typically represented by Σ.

alph. → $\Sigma = \{a, b, c\}$

string. → abc

3

- A string over an alphabet Σ is a finite sequence of characters from Σ.
  - Examples: Σ = {a, b, c} some valid strings include
    - abc,
    - baba,
    - aaaabbbbccc.

- Empty string, denoted by ∈, with length 0.

- Length, number of characters in string, denoted by $|x|$

LANGUAGE

$\{a, aa, aaa\}$

- A Language is a set of strings.
- We say that L is a *language over Σ* if it is a set of strings formed from characters in Σ.
- Example: The language of palindromes over Σ = {a, b, c} is the set

  same on both sides

  {ε, a, b, c, aa, bb, cc, aaa, aba, aca, bab, … }
- One special language is Σ^*, which is the set of all possible strings generated over the alphabet Σ^*.
- Formally we can say, L is a language over Σ iff $L \subseteq \Sigma^*$.
- Example: Σ = {a, b, c} then Σ^* = {ε, a, b, c, aa, ab, ac, ba, . . . , aaaaaabbbaababa, . . .} .

- universal set = Σ^* same as sigma including empty string inclusive of everything

LANGUAGE EXAMPLE

- The following is a language L = {b, ba, baa, baaa, baaaa, …} .

  Now, is the following a language? {aa, ab, ba, ε} .

  True

LANGUAGE EXAMPLE

- The following is a language L = {b, ba, baa, baaa, baaaa, …}
- .

Now, is the following a language? {aa, ab, ba, ε} .

Yes! because its finite, and its definitely a language.

How about {aa, ab, ba, Ø}. Is this a language?

No! Because Ø is no a valid string

null string

Ø

You cannot have a null string in language, but you can have empty . ε

LANGUAGE NOTATION

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

$L_1 = \{ \varepsilon, aa, ab, bb, aaaa, aabb, \ldots \}$

- Example 1: Consider, set of strings $L_1 = \{x \mid x \in \{a, b\}^* \text{ and } |x| \text{ is even}\}$ In words, L_1, is the language of all strings made out of a, b that have even length.
- Example 2: Language $L_2 = \{x \mid \text{there is a } z \text{ where } xz = \text{apples}\}$

L_2 is the language made out of all prefixes of L_2 that is $\{\varepsilon$, a, ap, app, appl, apple, apples$\}$.

AUTOMATON

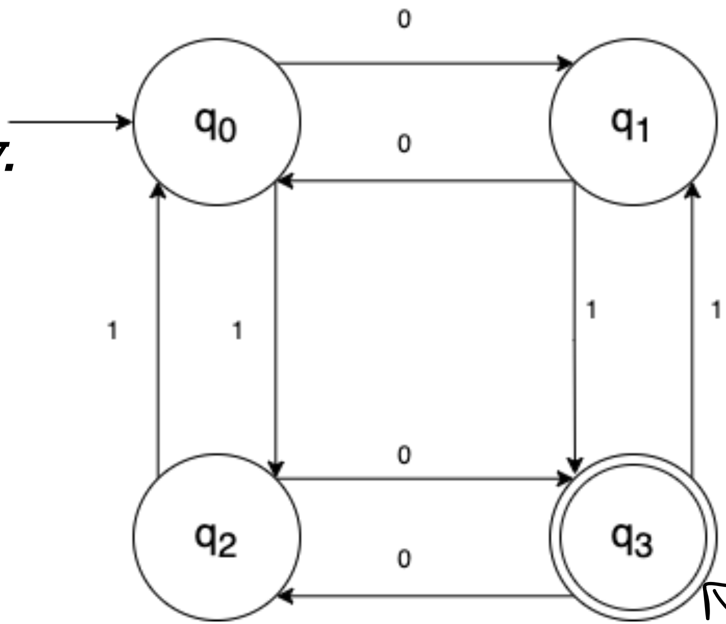
- Automaton is a simple, idealized mathematical computation machine that has limited memory.
- Automaton can also be said as a *simple state machine.*
- These machines are called as Finite State Automaton (FSM) or Finite State Automaton (FSA).
- In other words, a finite automaton is a mathematical machine for determining whether a string is contained within some language.

1 bit of memory.
Can only hold 1 state

STATE DIAGRAM

You always need a start state. You can never have a machine without a start state

Start state, always starts with an arrow.



Circles represent a state of Automaton.

input

final state, machine must have
or end state, accept

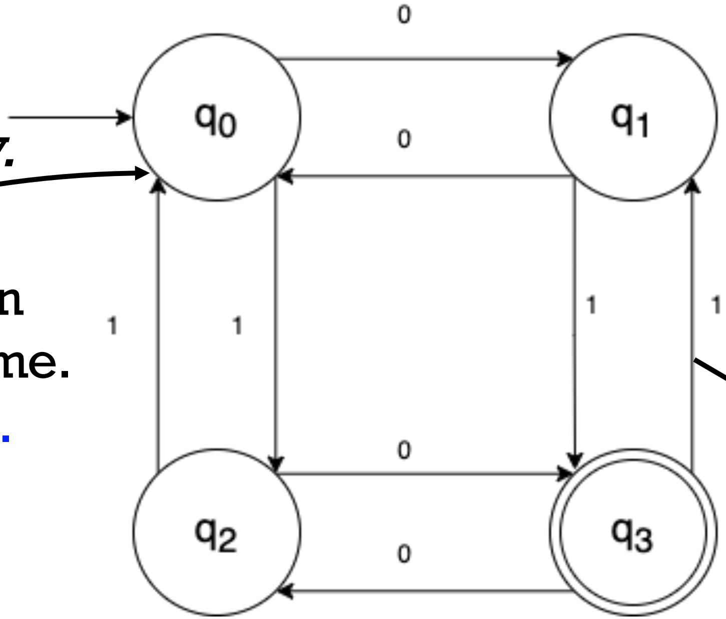
The automaton is run on an *input string* and answers "yes" or "no."

only have 1
Start state

STATE DIAGRAM

Start state, always starts with an arrow.

Automaton is always in one state at a given time. It begins in Start State.

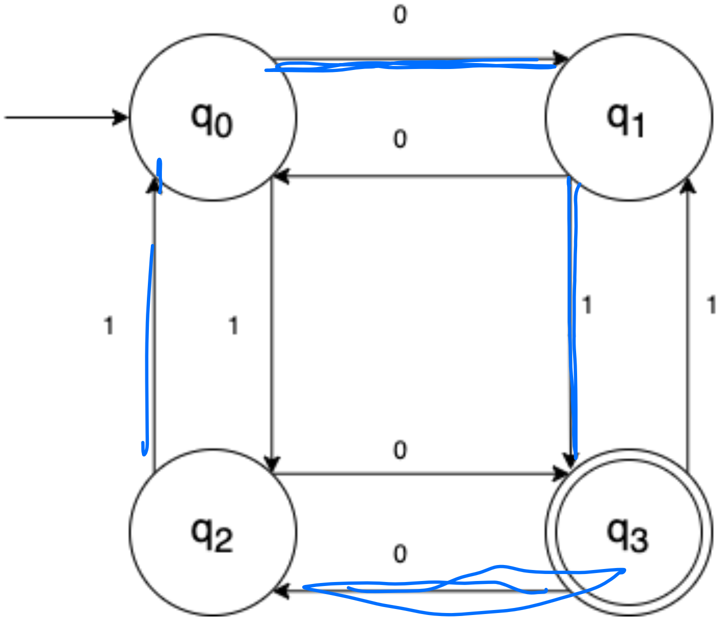


Circles represent a state of Automaton.

Each arrow is called the *transition*

STATE DIAGRAM

Always begin with start state which is “q_0” here.



Input: **0 1 0 1 1 0**

$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_3 \xrightarrow{0} q_2$

$\downarrow$ 1

q_0

$\downarrow$ 1

q_2

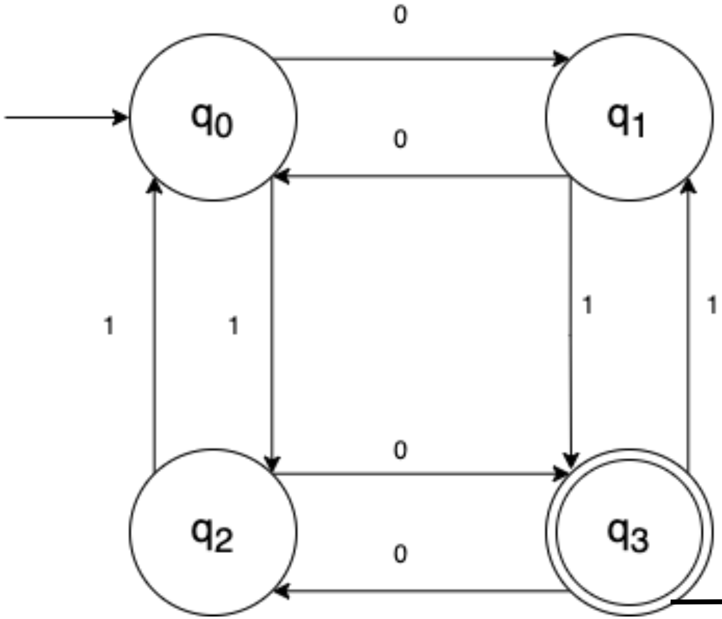
$\downarrow$ 0

q_3

Accepted

STATE DIAGRAM

Always begin with start state which is "q_0" here.



In q_0, → q_1 , on input 0.
q_1 → q_3 , on input 1.
Etc…..

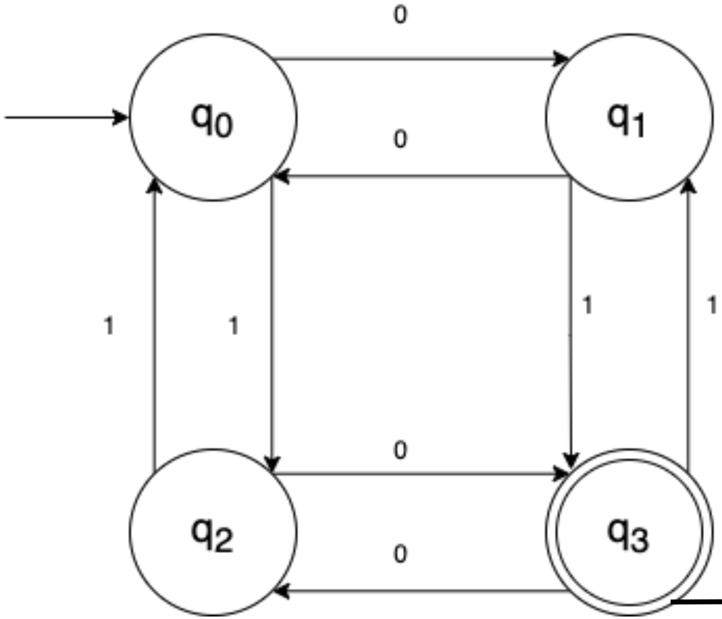
The double circle indicates that this
state is an *accepting state*, the automaton
outputs "yes."

Input: **0 1 0 1 1 0**

STATE DIAGRAM

$q_0 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_1 \xrightarrow{0} q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_0$

Always begin with start state which is "q_0" here.



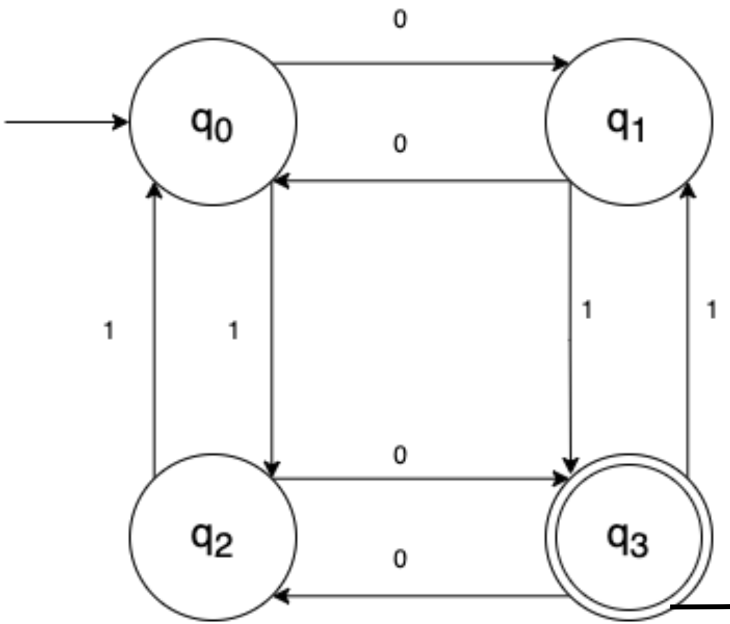
The double circle indicates that this state is an *accepting state*, the automaton outputs "yes."

Input: **1 0 1 0 0 0**

Rejected by machine

STATE DIAGRAM

Always begin with start state which is "q_0" here.



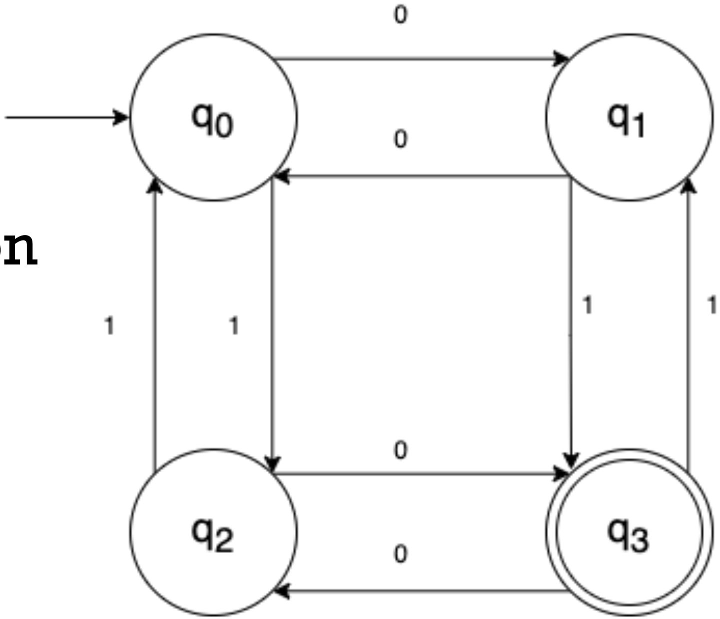
This state is not an accepting state, so the automaton says "*no*"

The double circle indicates that this state is an *accepting state*, the automaton outputs "yes."

Input: **1 0 1 0 0 0**

EXAMPLE: STATE DIAGRAM

$q_0 \xrightarrow{1} q_2 \xrightarrow{1} q_0 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{0} q_0 \xrightarrow{0} q_1$



Does the automaton *accept* or *reject*?

Rejected

Input: $\cancel{1}\ \cancel{1}\ \cancel{1}\ \cancel{0}\ \cancel{1}\ \cancel{1}\ \cancel{1}\ \cancel{0}\ \cancel{0}$

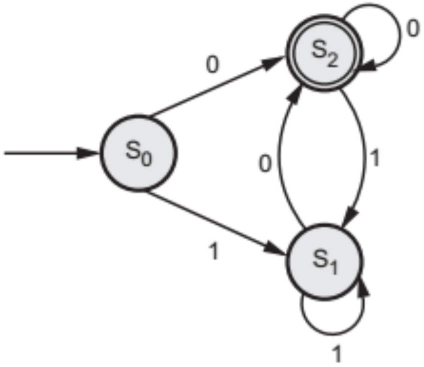
EXAMPLE

a. $s_0 \xrightarrow{0} s_2 \xrightarrow{1} s_1 \xrightarrow{0} s_2 \xrightarrow{0} s_2 \xrightarrow{0} s_2$

Accepted

b. $s_0 \xrightarrow{1} s_1 \xrightarrow{0} s_2 \xrightarrow{1} s_1 \xrightarrow{1} s_1$

Rejected



a. 01000
b. 1011

FINITE AUTOMATON

- Does the automaton accept input once it reaches the accepting/final state? after parsing through input & then decide
  - NO! , after parsing through input & then decide
- When do you consider if a finite automaton is accepted?
  - When it ends in final state. after reading all of input

SUMMARY

- A finite automaton is a collection of states joined by transitions.
- Some state is designated as the start state. O; only 1
- Some states are designated as accepting states. O; can be multiple
- The automaton processes a string by beginning in the start state and following the indicated transitions.
- If the automaton ends in an accepting state, it *accepts* the input. Otherwise, the automaton *rejects* the input.

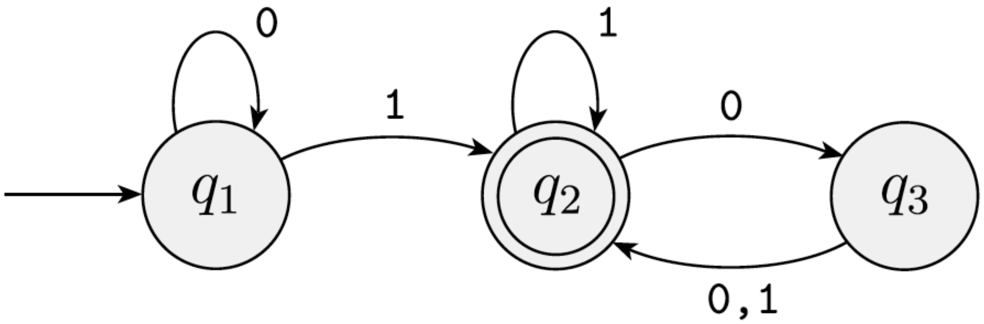
FORMAL DEFINITION OF FA

- A finite automaton is a 5-tuple (Q,Σ, δ, q0, F), where

1. Q → a finite set called the states,
2. Σ → a finite set called the alphabet,
3. δ → Q x Σ, transition function,
4. q0 → the start/initial state, q0 ∈ Q
5. F → the set of accept/final states, F ⊆ Q

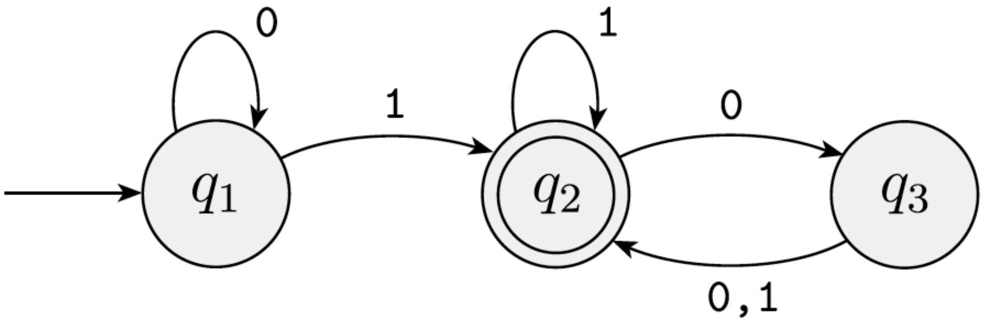
You can have multiple accept states

EXAMPLE



- Q ? $\{q_1, q_2, q_3\}$
- Σ ? $\{0, 1\}$
- δ ?
- q0 ? q_1
- F ? $\{q_2\}$

you can have multiple final states so need
{ }



Σ

δ , transition table is given below

Q	0	1
q1	q_1	q_2
q2	q_3	q_2
q3	q_2	q_2

- Q – $\{q_1, q_2, q_3\}$
- Σ - $\{0, 1\}$
- q0 - q_1
- F - $\{q_2\}$

The language of an automaton is the set of strings that it accepts.

If D is an automaton, we denote the language of D as $\mathscr{L}(D)$.

$$\mathscr{L}(D) = \{ w \in \Sigma^* \mid D \text{ accepts } w \}$$

If A is the set of all strings that machine M accepts, we say that A is the

language of machine M and write L(M) = A.

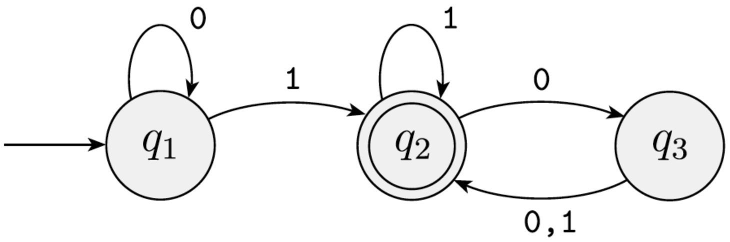
We say that M recognizes A or

That M accepts A.

A machine may accept many strings, but it always recognizes only one language.

if the machine accepts no strings, it still recognizes one language- ε or $\emptyset$

M accepts strings but recognizes a language.



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

Then L(M1) = A, or equivalently, M1 recognizes A.

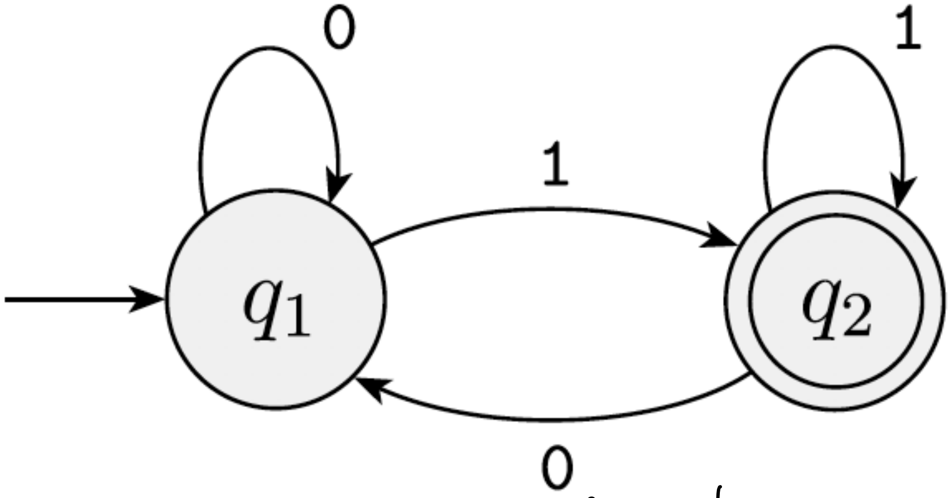
STATE DIAGRAM OF THE TWO-STATE FINITE AUTOMATON

$Q = \{q_1, q_2\}$

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

$q_0 = q_1$

$F = \{q_2\}$



δ :

	0	1
q_1	q_1	q_2
q_0	q_1	q_2

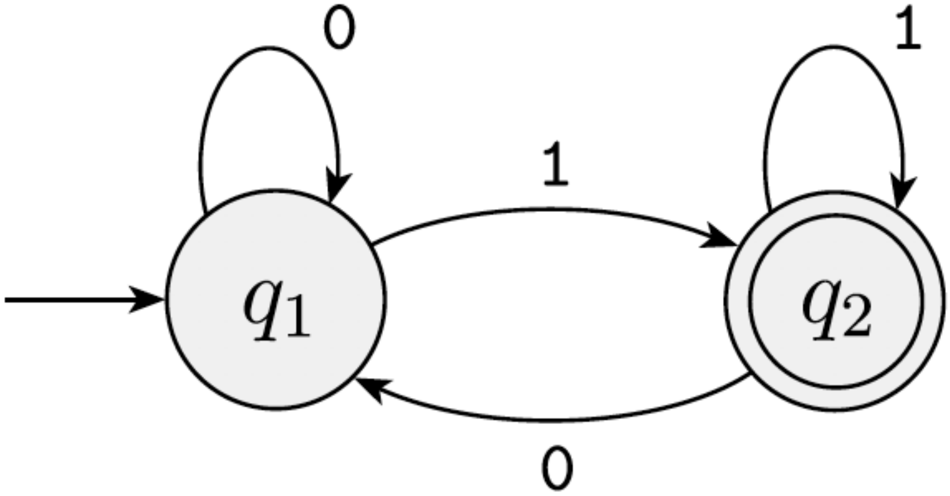
- What is the formal definition?
- What is the language it recognizes?

STATE DIAGRAM OF THE TWO-STATE FINITE AUTOMATON

$$110 \times \quad q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1$$

The string has to end in 1.

$$1101 \checkmark \quad q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_2$$



M2 =

	0	1
q1		
q2		

$$\checkmark 11 \quad q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2$$

L(M2) =

$Q = \{q_1, q_2\}$

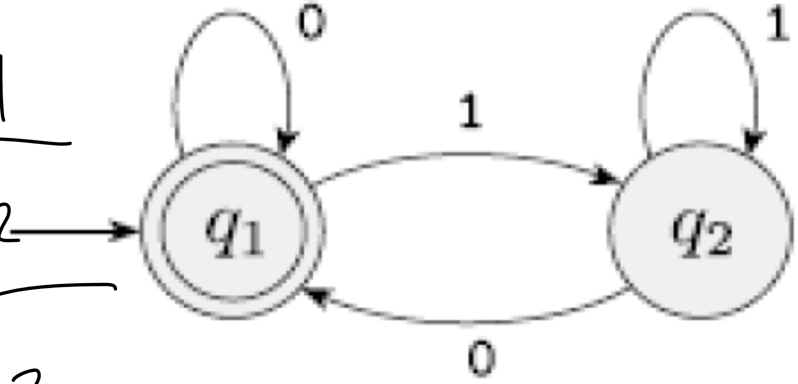
$\Sigma = \{0, 1\}$

$q_0 = q_1$

$F = \{q_1\}$

	0	1
q_1	q_1	q_2
q_2	q_1	q_2

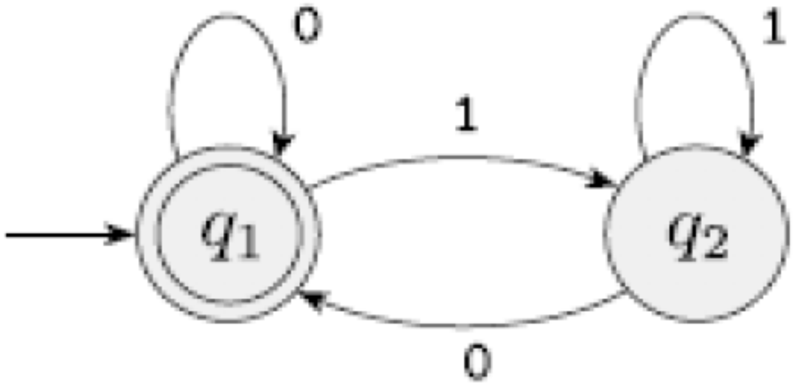
EXAMPLE TO TRY!



- What is the formal definition?
- What is the language it recognizes?

It should end in 0.

EXAMPLE TO TRY!



EXAMPLE

$\delta =$

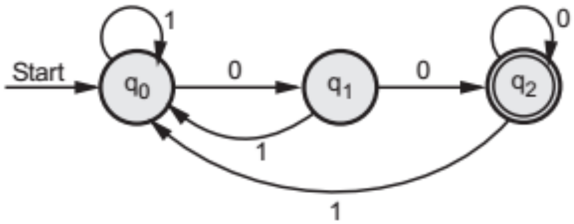
	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_0

✓ 100

$q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2$

✗ 1010

$q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_1$



- What is the formal definition?
- What is the language it recognizes?

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

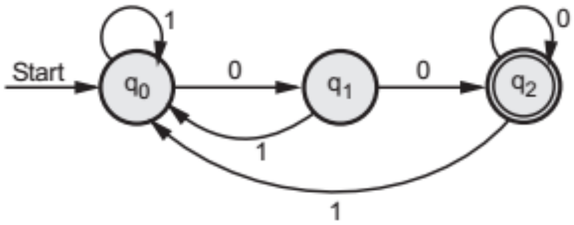
$\Sigma = \{0, 1\}$

$q_0 = q_0$

$F = \{q_2\}$

Your string should end in at least 2 zeros

EXAMPLE



EXAMPLE

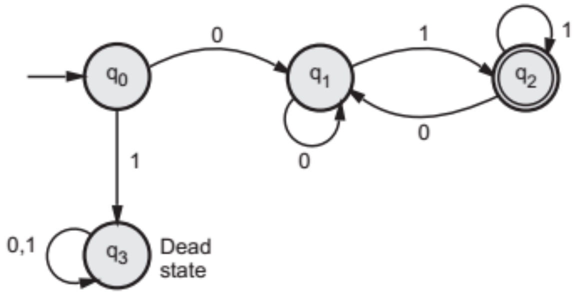
$Q = \{q_0, q_1, q_2, q_3\}$

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

$q_0 = q_0$

$F = \{q_2\}$

$\delta =$	0	1
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_2
q_3	q_3	q_3



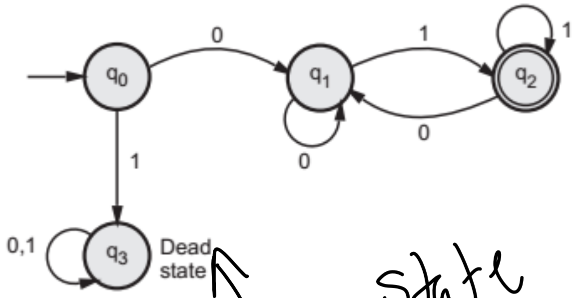
- What is the formal definition?
- What is the language it recognizes?

The string has to start with 0 and end with 1

0111

$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2$ ✓

EXAMPLE



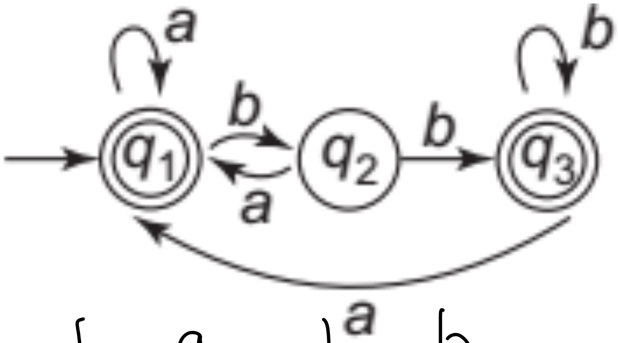
trap state

$$L = \{x \mid x \in \{0,1\}^* \text{ and }$$
$$x = 0yz1\}$$

$$0y^*1$$

CAN WE HAVE MORE THAN ONE FINAL STATE?

YES!!



What is the formal definition?

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

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

$\delta =$

	a	b
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_1	q_3

$q_0 = q_1$

$F = \{q_1, q_3\}$

FORMAL DEFINITION OF COMPUTATION

$M = (Q, \Sigma, \delta, q0, F)$ be a finite automaton and $w = w_1, w_2, w_3, \ldots . w_n$ be a string where each w_i is a member of the alphabet Σ.

Then M accepts w if a sequence of states $r_0, r_1, \ldots, r_n$ in Q exists with three conditions:

1. $r_0 = q_0$, Start with start state
2. $\delta(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, \ldots, n - 1$, and
3. $r_n \in F$. ends with accept state

M recognizes language A if $A = \{w \mid M \text{ accepts } w\}$.

FORMAL DEFINITION OF COMPUTATION

- String w is accepted if $\delta^*(q0, w) \in F$, that is, w leads from the start state to an accepting state.
- String w is rejected if it isn't accepted.
- A language is any set of strings over some alphabet.
- L(M), language recognized by finite automaton M = { w| w is accepted by M}.
- A language is *regular, or FA-recognizable*, if it is recognized by some finite automaton.

Can't develop languages like $0^n, 1^n$

ababa

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_1$

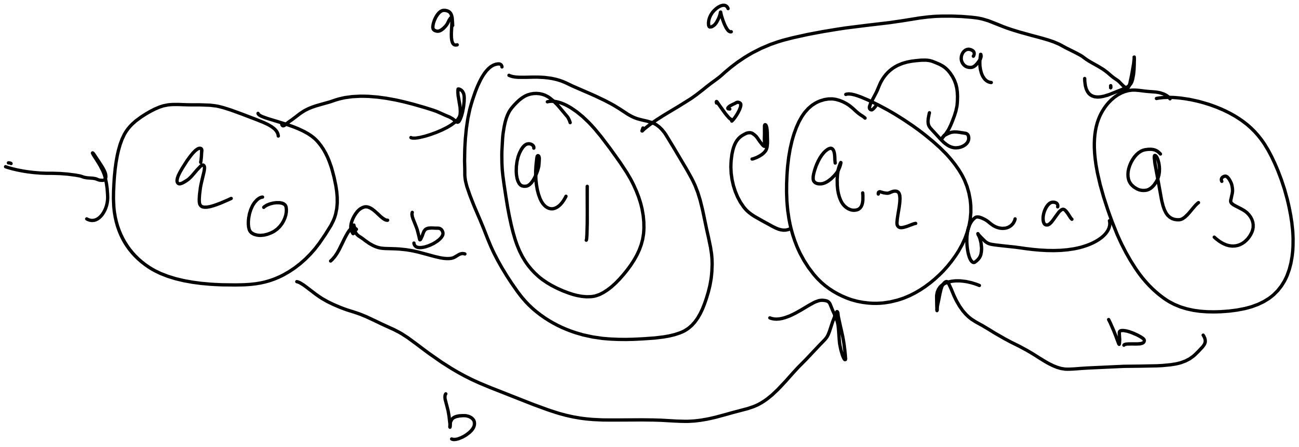
FA EXAMPLE

	a	b
→ q_0	q_1	q_2
☆ q_1	q_3	q_0
q_2	q_2	q_2
q_3	q_2	q_2

aaba

$q_0 \xrightarrow{a} q_1 \rightarrow$

- Let M: ({q0,q1,q2,q3},{a,b},q0,q1,δ) where transition is given by δ(q0,a)=q1, δ(q1,a)=q3, δ(q2,a)=q2, δ(q3,a)=q2; δ(q0,b)=q2, δ(q1,b)=q0, δ(q2,b)=q2, δ(q3,b)=q2.
  - Represent M by its state table
  - Represent M by its state diagram
  - Which of the following strings are accepted by M ababa, aabba.



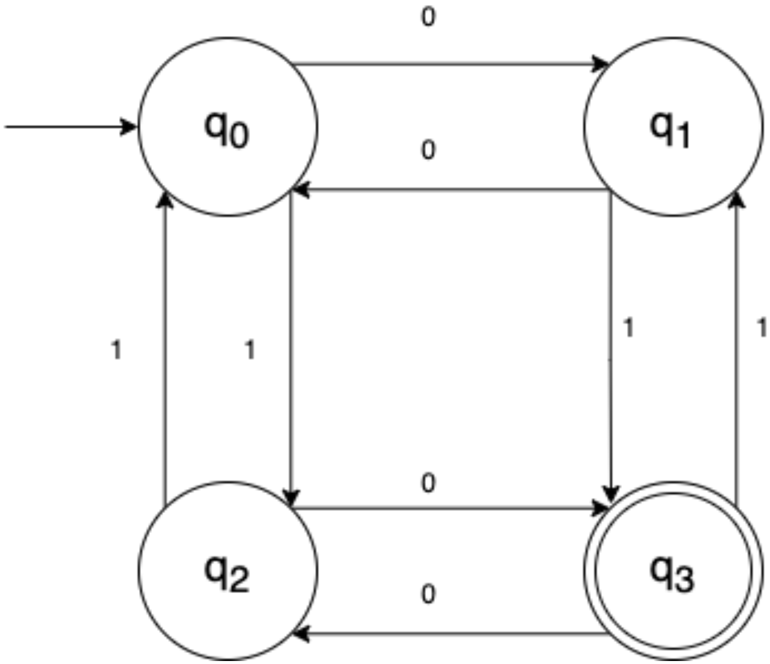
DETERMINISTIC FINITE AUTOMATON(DFA)

- A DFA is a
  - **D**eterministic
  - **F**inite
  - **A**utomaton
- DFAs are the simplest type of automaton.
- It has very limited memory

INFORMAL DEFINITION OF DFA

- A DFA is defined relative to some alphabet Σ.
- For each state in the DFA, there must be exactly one transition defined for each symbol in the alphabet.
  - This is the "deterministic" part of DFA.
- There is a unique start state.
- There are zero or more accepting states.

IS THIS A DFA?



Yes, it has
unique start state
1 accept state
0, 1 on each

IS THIS A DFA?

No

NFA