FINITE STATE AUTOMATA

- Finite State Automata (FSA) are the simplest automata.
- Only the finite memory in the control unit is available.
- The memory can be in one of finite states at a given time – hence the name.
  - One can remember only a (fixed) finite number of properties of the past input.
  - Since input strings can be of arbitrary length, it is not possible to remember unbounded portions of the input string.
- It comes in Deterministic and Nondeterministic flavors.

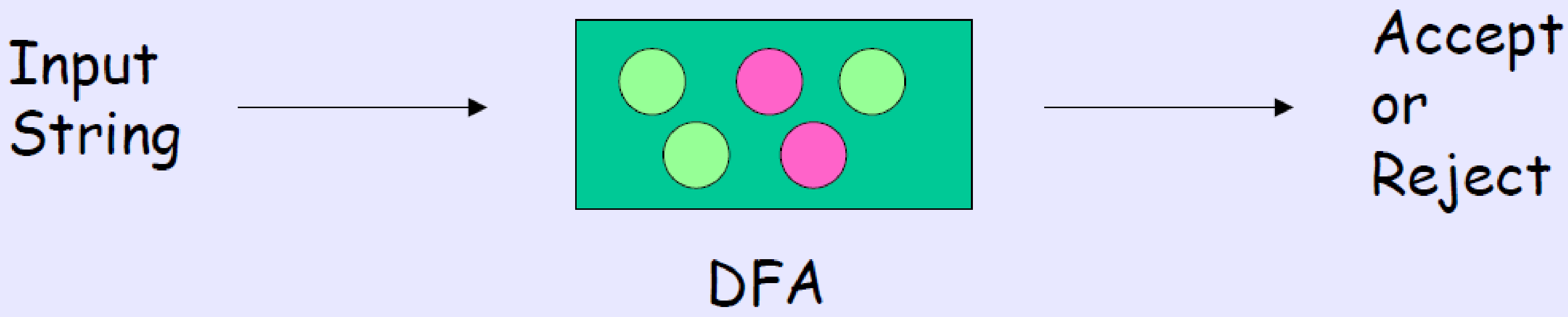
Example: Switch

Deterministic Finite State Automata (DFA)

- A DFA starts in a start state and is presented with an input string.
- It moves from state to state, reading the input string one symbol at a time.
- What state the DFA moves next depends on
  - the current state,
  - current input symbol
- When the last input symbol is read, the DFA decides whether it should accept the input string

Finite State Machines

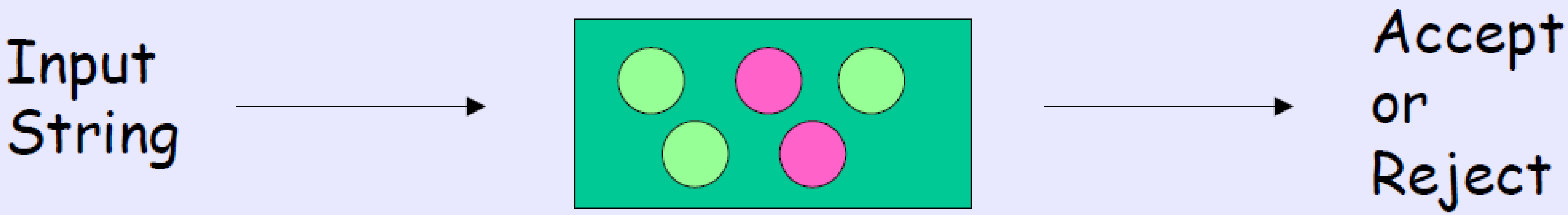
DFA



- A machine with finite number of states, some states are accepting states, others are rejecting states
- At any time, it is in one of the states
- It reads an input string, one character at a time

DFA

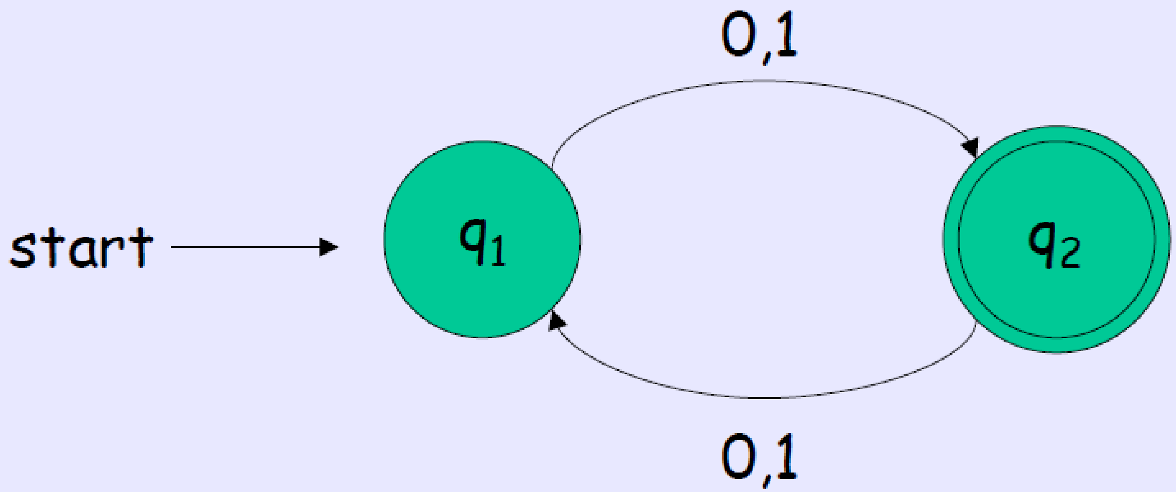
Example: Pattern recognition



DFA

- After reading each character, it moves to another state depending on what is read and what is the current state
- If reading all characters, the DFA is in an accepting state, the input string is accepted.
- Otherwise, the input string is rejected.

Example of DFA



- The circles indicates the states
- If accepting state is marked with double circle
- The arrows pointing from a state q indicates how to move on reading a character when current state is q

DFA – Formal Definition

- A Deterministic Finite State Acceptor (DFA) is defined as the 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where
  - Q is a finite set of states
  - Σ is a finite set of symbols – the alphabet
  - $\delta : Q \times \Sigma \to Q$ is the next-state function
  - $q_0 \in Q$ is the (label of the) start state
  - $F \subseteq Q$ is the set of final (accepting) states

DFA – Formal Definition

- A Deterministic Finite State Acceptor (DFA) is defined as the 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where
  - Q is a finite set of states
  - Σ is a finite set of symbols – the alphabet
  - $\delta : Q \times \Sigma \to Q$ is the next-state function
  - $q_0 \in Q$ is the (label of the) start state
  - $F \subseteq Q$ is the set of final (accepting) states

Note, there must be exactly one start state.
Final states can be many or even empty !

Some Terminology

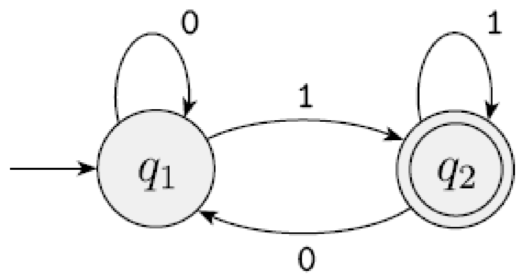
Let M be a DFA

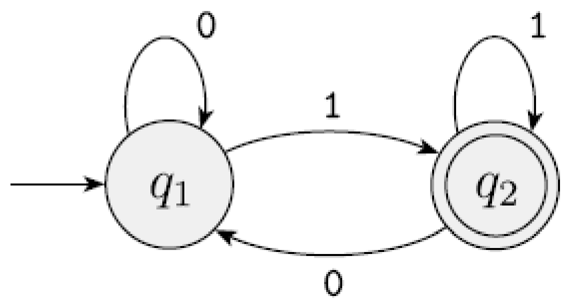
- Among all possible strings, M will accept some of them, and M will reject the remaining
- The set of strings which M accepts is called the language recognized by M
- That is, M recognizes A if

$$A = \{ w \mid M \text{ accepts } w \}$$

$L(M)$

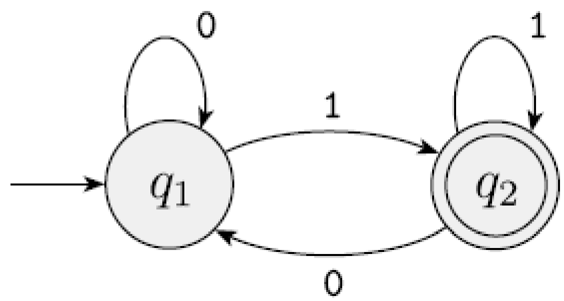
If A is the set of all strings that machine M accepts, we say that A is the ***language of machine M*** and write $L(M) = A$. We say that ***M recognizes A*** or that ***M accepts A***.


0
1
1
q_1
q_2
0



In the formal description, M_2 is $\big(\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_2\}\big)$. The transition function δ is

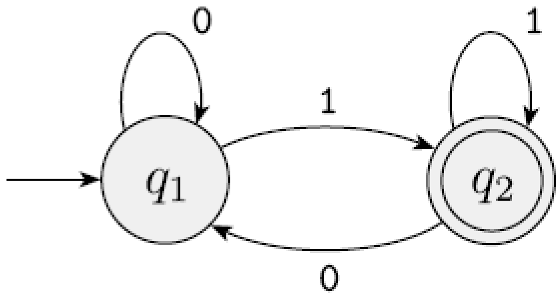
	0	1
q_1	q_1	q_2
q_2	q_1	q_2.



In the formal description, M_2 is $(\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_2\})$. The transition function δ is

	0	1
q_1	q_1	q_2
q_2	q_1	q_2.

Remember that the state diagram of M_2 and the formal description of M_2 contain the same information, only in different forms. You can always go from one to the other if necessary.



M_2

$$L(M_2) = \{w \mid w \text{ ends in a } 1\}.$$

Consider the finite automaton M_3.

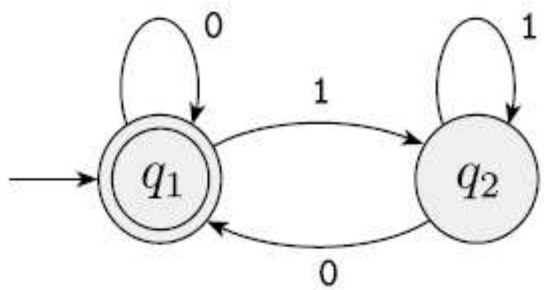


FIGURE 1.10
State diagram of the two-state finite automaton M_3

Can you describe this in the 5 tuple form?
In particular, can you write down the transition table?

Consider the finite automaton M_3.

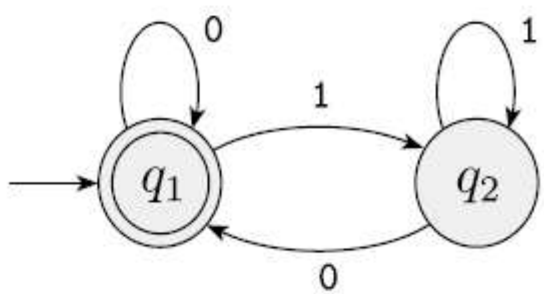


FIGURE 1.10
State diagram of the two-state finite automaton M_3

What language M_3 recognizes?

Consider the finite automaton M_3.

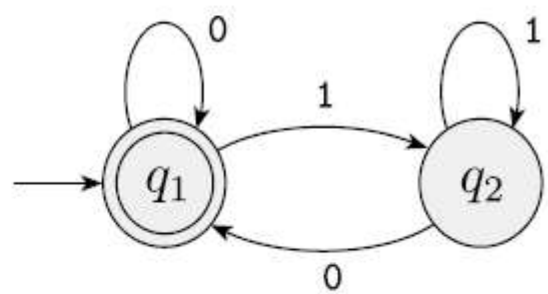


FIGURE 1.10
State diagram of the two-state finite automaton M_3

What language M_3 recognizes?

$L(M_3) = \{w|\ w \text{ is the empty string } \varepsilon \text{ or ends in a } 0\}$.

EXAMPLE 1.11

The following figure shows a five-state machine M_4.

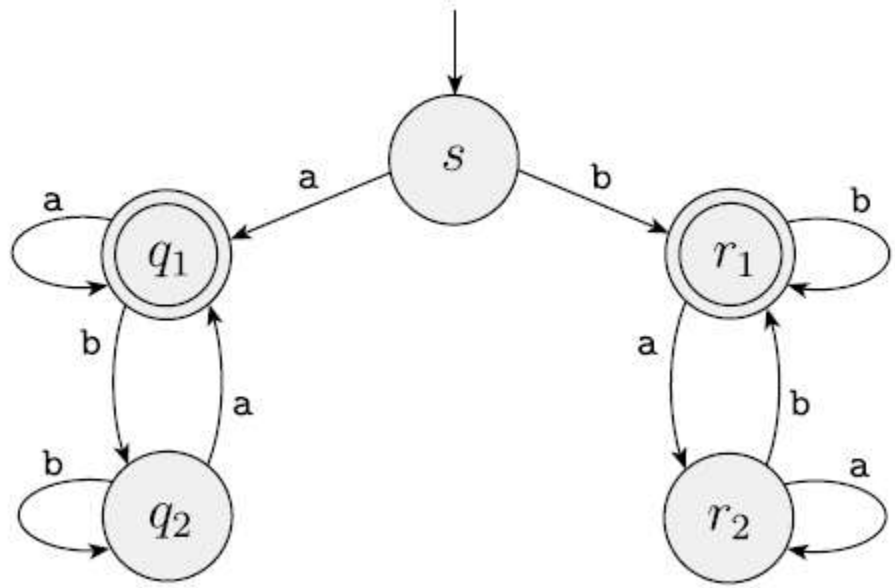


FIGURE 1.12
Finite automaton M_4

EXAMPLE 1.11

The following figure shows a five-state machine M_4.

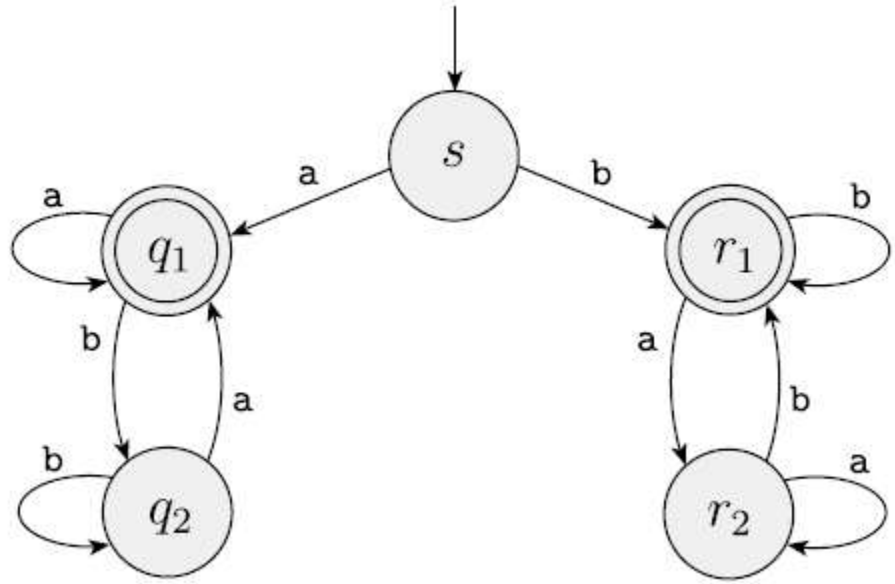


FIGURE 1.12
Finite automaton M_4

$L(M_4)$ = all strings that begin and end with the same character.