بسم الله الرحمن الرحیم

# Formal Languages & Automata

Ali Shakiba
Vali-e-Asr University of Rafsanjan
<ali.shakiba@vru.ac.ir>

# Chapter 2: Finite Automata

# Deterministic Acceptors

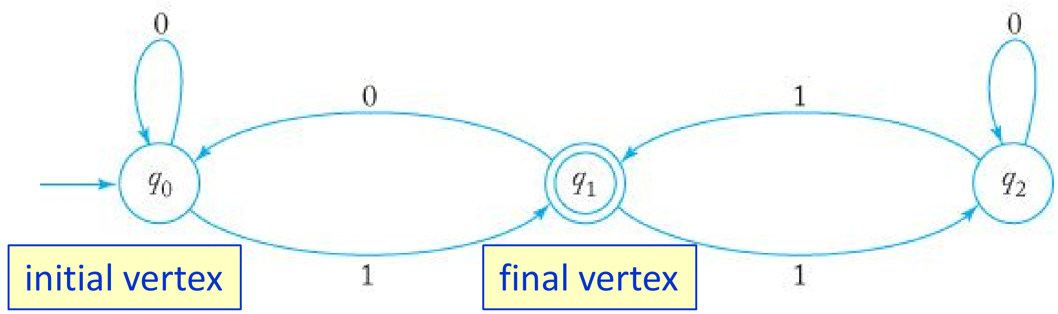
- A deterministic finite acceptor (DFA) is the quintuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

  where:
  - $Q$ is the finite set of internal states
  - $\Sigma$ is the input alphabet, a finite set of symbols
  - $\delta: Q \times \Sigma \rightarrow Q$ is a total transition function
  - $q_0 \in Q$ is the initial state
  - $F \subseteq Q$ is a set of final states

**Total function**: A function that is defined for all inputs of the right type (*i.e.*, for all inputs from a given domain).

# Transition Graph Example

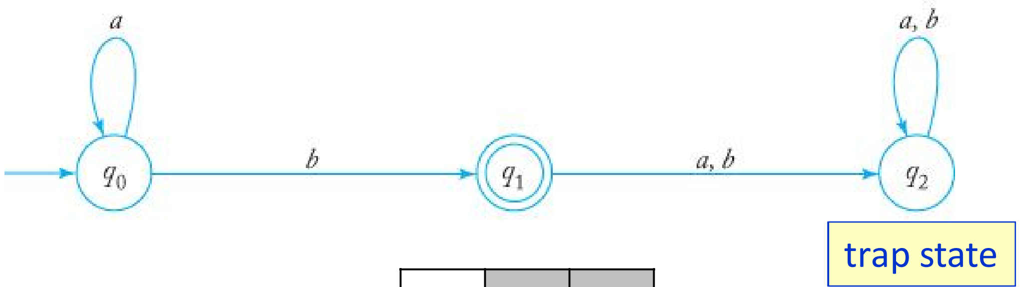



$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

where $\delta$ is given by

| | |
|---|---|
| $\delta(q_0, 0) = q_0$ | $\delta(q_0, 1) = q_1$ |
| $\delta(q_1, 0) = q_0$ | $\delta(q_1, 1) = q_2$ |
| $\delta(q_2, 0) = q_2$ | $\delta(q_2, 1) = q_1$ |

# State Transition Matrix




| | $a$ | $b$ |
|---|---|---|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_2$ | $q_2$ |
| $q_2$ | $q_2$ | $q_2$ |

# A Practical Application

- You are given the text of the novel *War and Peace* as a plain text file.
- Write a program to search the text for these names:
  - Boris Drubetskoy
  - Joseph Bazdeev
  - Makar Alexeevich
- For each name found, print the line number and the position within the line.
  - Line and position numbers start with 1

# Languages and DFAs

- Recall that an acceptor is an automaton that either accepts or rejects input strings.

- The set of all strings that the DFA
$$M = (Q, \Sigma, \delta, q_0, F)$$
accepts constitutes the language
$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$$

- The DFA represents the language's rules.

# DFA and Associated Transition Graph

- If we have a DFA

$$M = (Q, \Sigma, \delta, q_0, F)$$

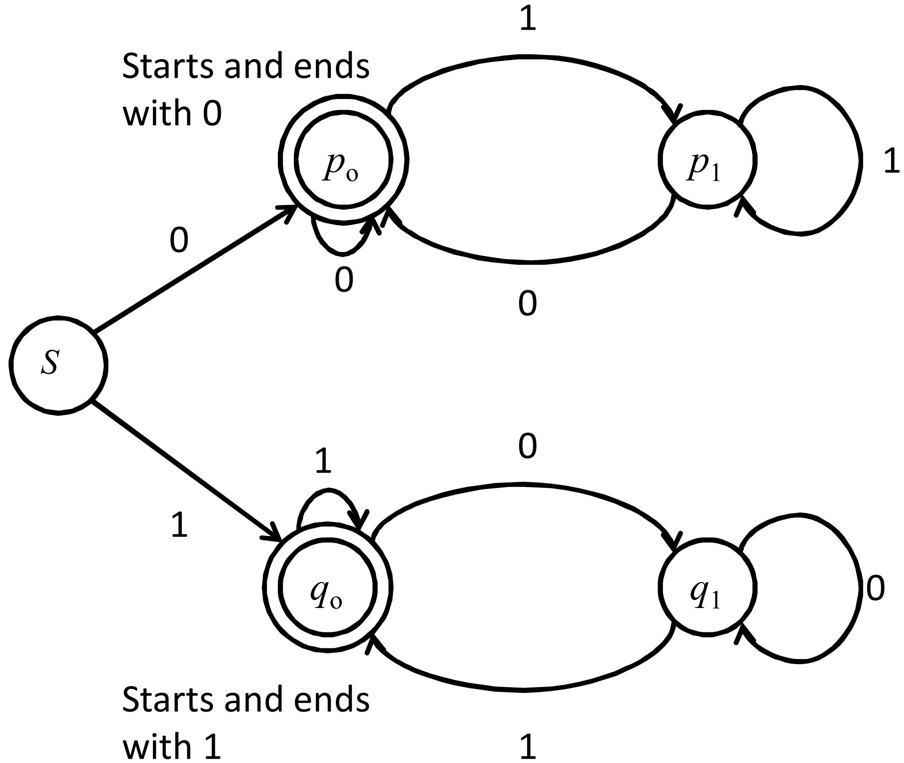
and its associated transition graph $G_M$, can we treat them both equally?

- Theorem 2.1 of the textbook basically says yes.
  - For every $q_i$, $q_j$ in $Q$, and $w$ in $\Sigma^+$:
  - $\delta^*(q_i, w) = q_j$ if and only if there is a path labeled $w$ in $G_M$ from $q_i$ to $q_j$.
  - Proof by induction on the length of $w$.

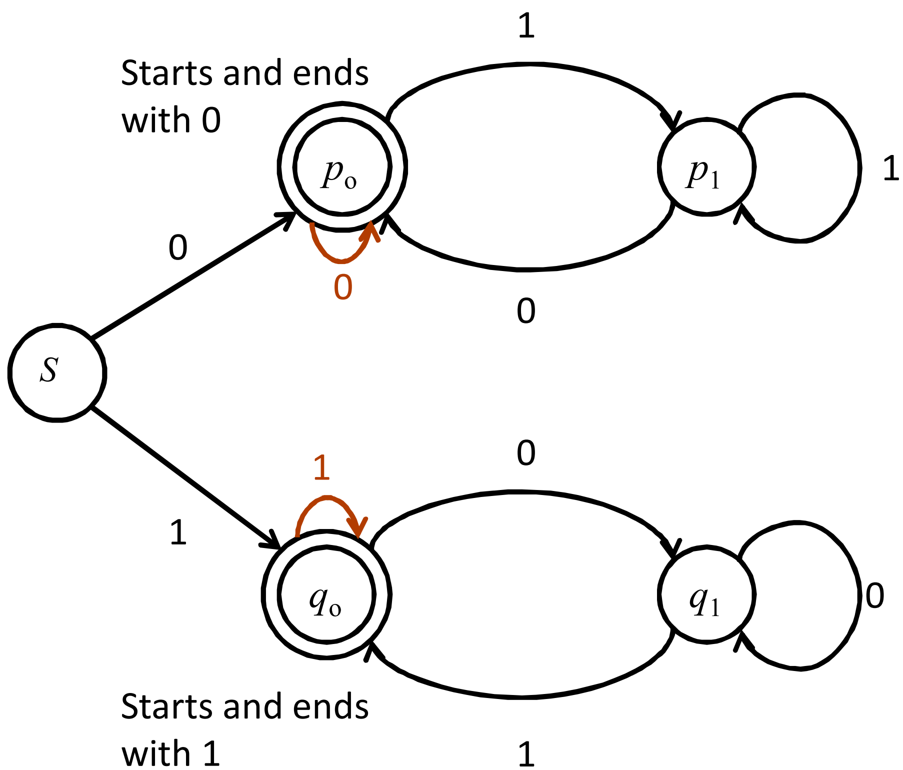
# DFA Example #1

- Create a DFA that accepts all strings on {0, 1} that begin and end with the same symbol.
- How can an automaton remember what was the beginning symbol of a string?
- Have a different set of states depending on the first symbol!

# DFA Example #1, *cont'd*



# DFA Example #1, *cont'd*



# DFA Example #2

- Create a DFA that accepts all strings on {0, 1} that do <u>not</u> contain the substring 001.

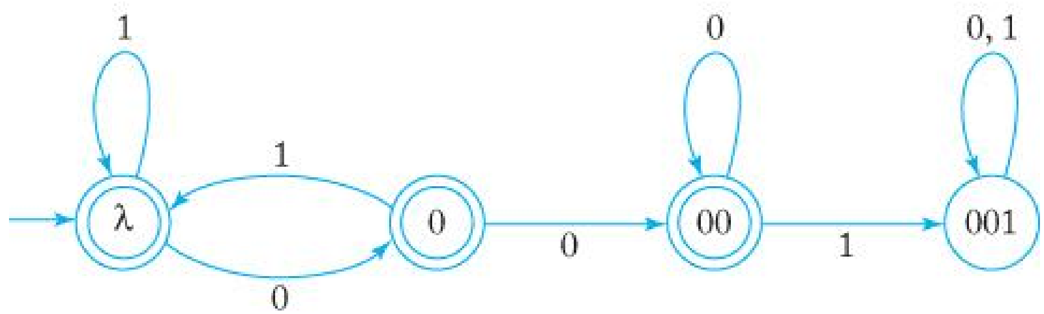
- The basic idea is that if the automaton ever reads 001, it should be in a non-final state.
  - Actually, that state should be a trap.

- How can the automaton remember the previous two symbols whenever it reads a 1?

# DFA Example #2, *cont'd*

- Again, we must accomplish this with states:
  - A state for having read a 0.
  - A state for having read 00.
  - A state for having read 001.

- We can label the states accordingly.



# Regular Languages

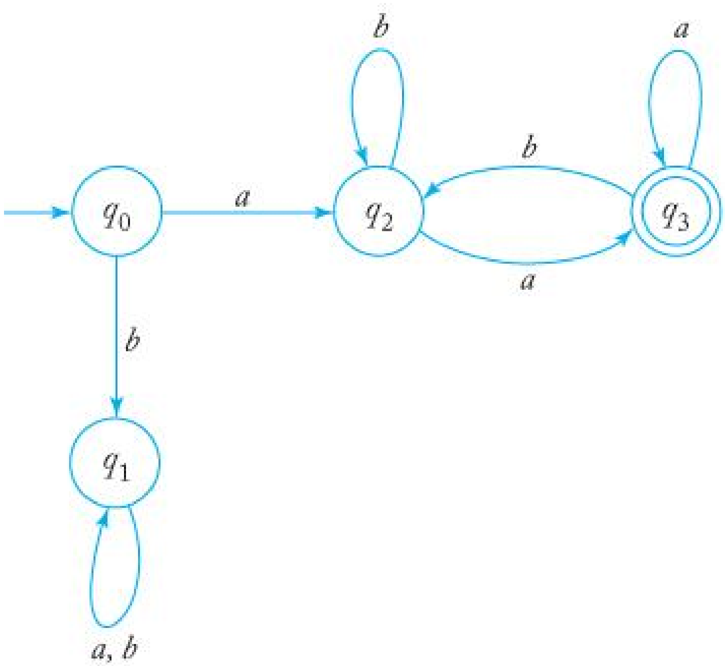
- A language $L$ is called regular if and only if there exists a DFA $M$ such that $L = L(M)$.

- Therefore, to show that a language is regular, all we have to do is find a DFA for it.

# Regular Language Example #1

$$L = \{awa : w \in \{a, b\}^*\}$$

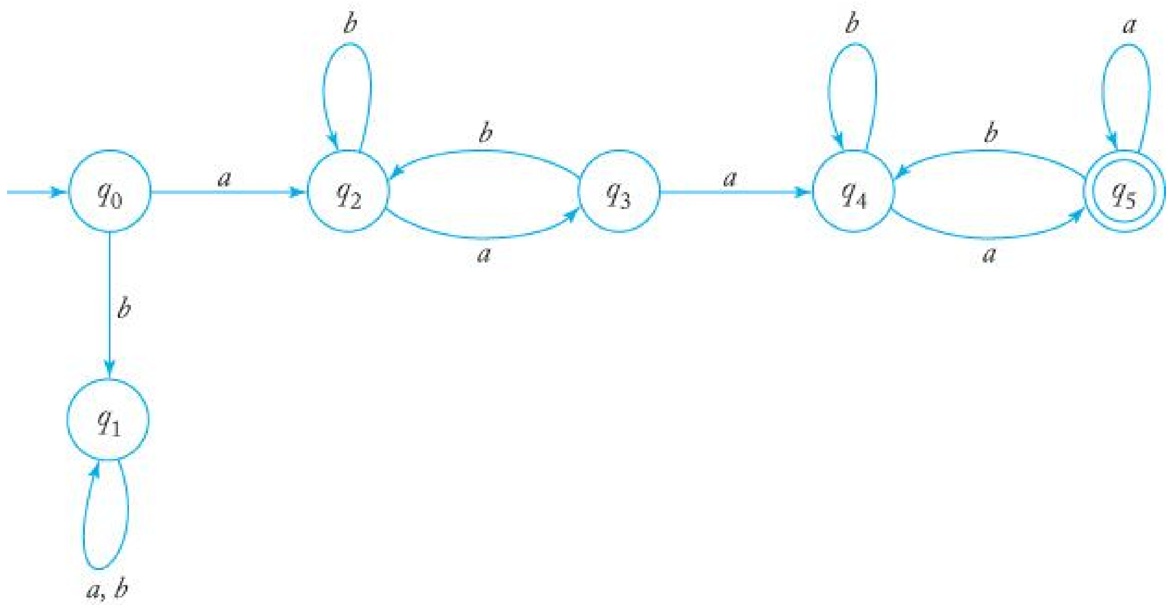
- Is the language $L = \{awa \mid w \in \{a, b\}^*\}$ regular?
  - It is if we can find a DFA for it.




After having read the leading $a$, we don't know if a subsequent $a$ is the last symbol of the string. So we can actually leave the final state and later return to it.

# Regular Language Example #2

- What about $L^2 = \{aw_1aaw_2a \mid w_1, w_2 \in \{a,b\}^*\}$?




and so $L^2$ is also regular.