

# Finite automata

---

# Context

- Đặt vấn đề
- Các khái niệm
- Deterministic Finite Automata
- Tóm tắt

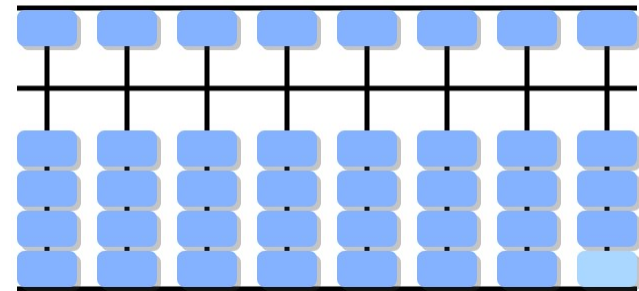
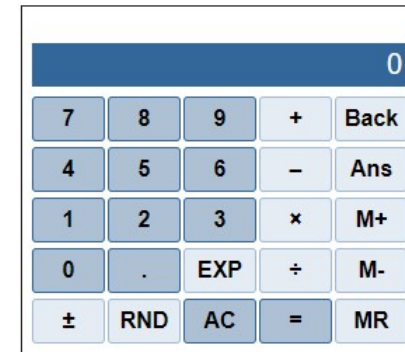
---

# Automata

- Automata (số nhiều) - automaton (số ít)
- Một automaton là một mô hình toán của một thiết bị
  - Trừu tượng hóa của máy tính thực, e.g: đồ thị ~ các mạng xã hội, đường giao thông
- Automata
  - Hiệu quả: mô hình hóa nhiều thiết bị tính toán
  - Đơn giản: suy luận kết quả với không gian nhớ nhỏ

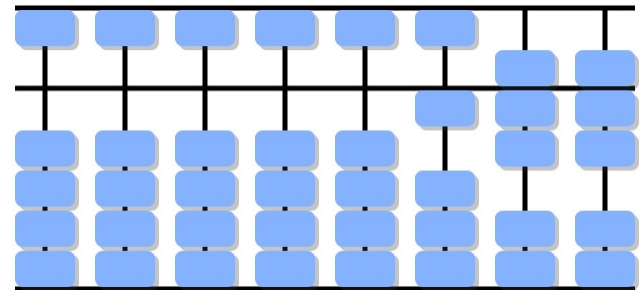
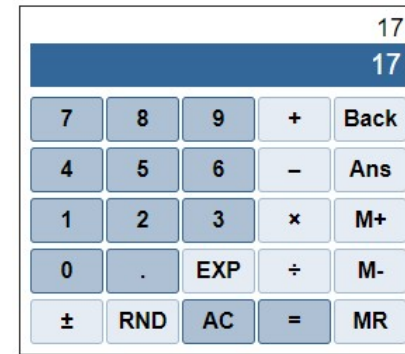
# Máy tính vs. Bàn tính

- Nhận đầu vào từ bên ngoài
- Đầu vào đưa vào tuần tự, một đơn vị ở một thời điểm
- Mỗi đầu vào làm hệ thống thay đổi cấu hình ~ tính toán



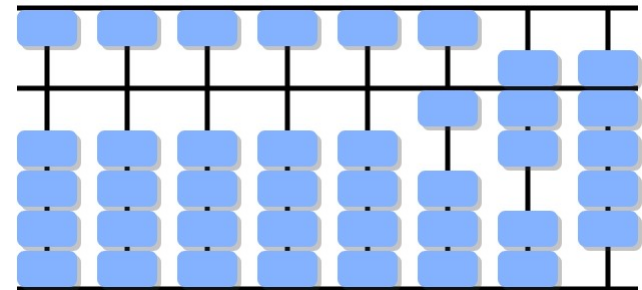
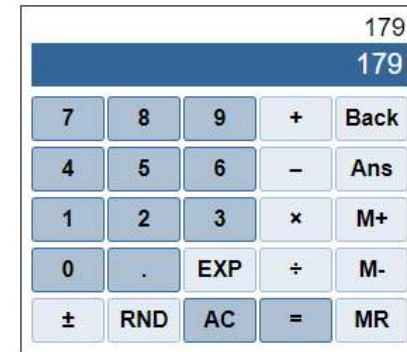
# Máy tính vs. Bàn tính

- Nhận đầu vào từ bên ngoài
- Đầu vào đưa vào tuần tự, một đơn vị ở một thời điểm
- Mỗi đầu vào làm hệ thống thay đổi cấu hình ~ tính toán



# Máy tính vs. Bàn tính

- Nhận đầu vào từ bên ngoài
- Đầu vào đưa vào tuần tự, **một đơn vị ở một thời điểm**
- Mỗi đầu vào làm hệ thống **thay đổi cấu hình** ~ tính toán
- Sau khi tất cả đầu vào được cung cấp, **một kết quả** được ghi nhận dựa trên cấu hình của thiết bị

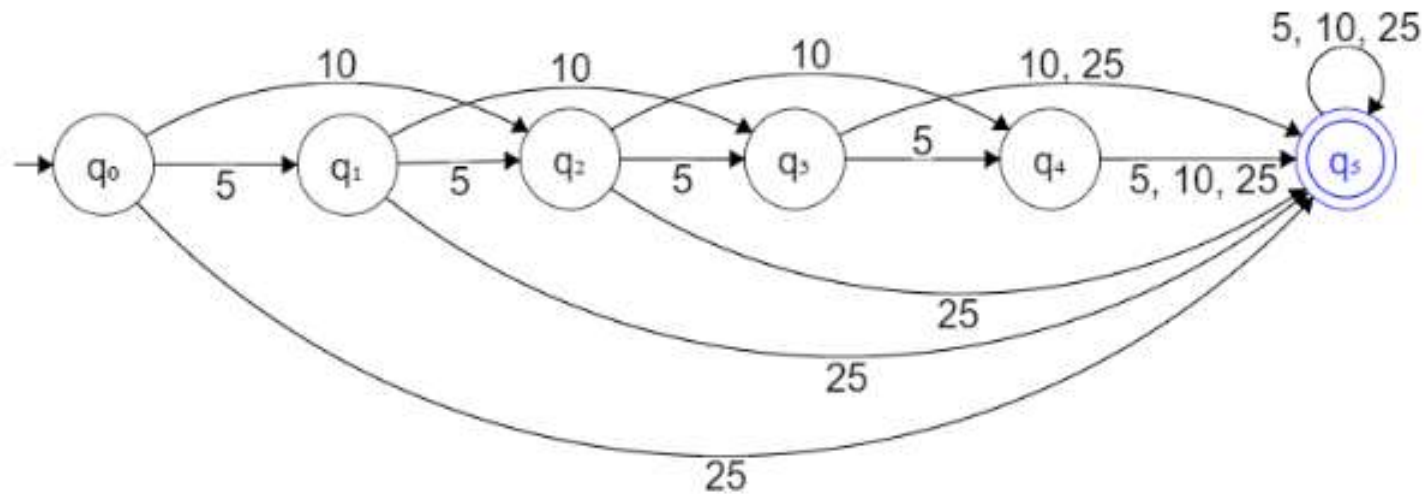


# Trạm thu phí

- Mỗi xe trả 25 cents để qua trạm
- Có các loại xu: 5, 10, 25 cents. Giả sử không trả lại tiền thừa
- Hoạt động: Chèn xu vào khe tiền đến khi số tiền đủ 25 cents. Hệ thống xác định tài xế trả đủ 25 cents hay chưa.
- 6 trạng thái
  - q0: chưa nhận xu nào
  - q1: nhận đúng 5 cents
  - q2: nhận đúng 10 cents
  - q3: nhận đúng 15 cents
  - q4: nhận đúng 20 cents
  - q5: nhận đúng 25 cents



# Trạm thu phí



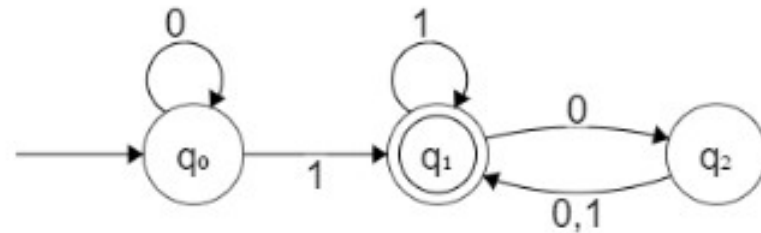
- Giả sử chuỗi (10, 5, 5, 10) được chèn  
 $q_0 \xrightarrow{10} q_2 \xrightarrow{5} q_3 \xrightarrow{5} q_4 \xrightarrow{10} q_5$

→ Gác chắn mở



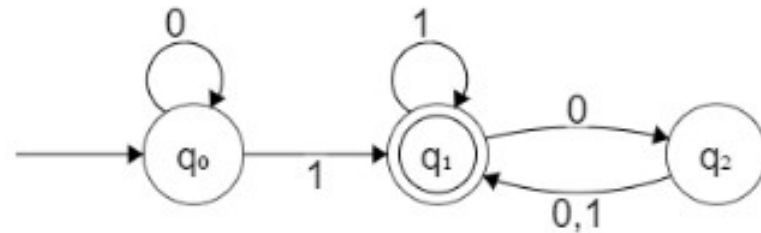
# Mô hình hóa

- Mô hình
  - Tập các trạng thái
  - Và các phép chuyển
- 1 trạng thái ~ 1 cấu hình có thể của hệ thống
- 1 phép chuyển ~ cách bộ nhớ thay đổi tương ứng với 1 đầu vào
- 1 trạng thái là trạng thái bắt đầu



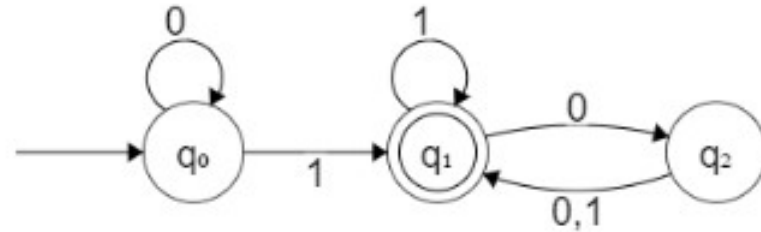
# Mô hình hóa

- Thiết bị xử lý các chuỗi ký tự
  - Mỗi ký tự ~ 1 ký tự đầu vào
  - Chuỗi ~ dãy ký tự vào
- Thực thi thiết bị
  - Bắt đầu từ trạng thái bắt đầu
  - Đọc chuỗi đầu vào từ trái sang phải
- 1 số trạng thái là **trạng thái chấp nhận** (double ring)
- Mỗi khi gặp 1 ký tự, thiết bị **đổi trạng thái** theo phép chuyển với nhãn là ký tự đó.
  - $w = 0101$ :  $q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_1 \rightarrow \text{YES}$
  - $w = 0101010$ :



# Mô hình hóa

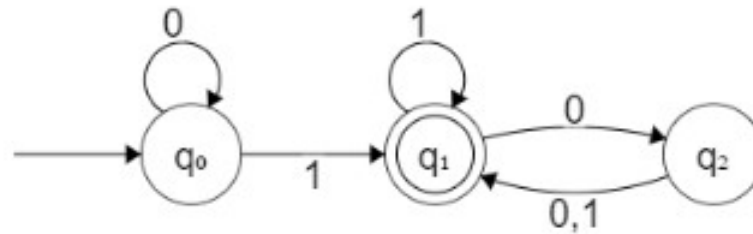
- Thiết bị xử lý các chuỗi ký tự
  - Mỗi ký tự ~ 1 ký tự đầu vào
  - Chuỗi ~ dãy ký tự vào
- Thực thi thiết bị
  - Bắt đầu từ trạng thái bắt đầu
  - Đọc chuỗi đầu vào từ trái sang phải
- 1 số trạng thái là **trạng thái chấp nhận** (double ring)
- Mỗi khi gặp 1 ký tự, thiết bị **đổi trạng thái** theo phép chuyển với nhãn là ký tự đó.
  - $w = 0101$ :  $q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_1 \rightarrow \text{YES}$
  - $w = 0101010$ :



**FINITE AUTOMATA**

# Finite automata

- Finite automata mô hình hóa các thiết bị
  - Có bộ nhớ giới hạn
  - Trả về YES/NO



---

# Context

- Đặt vấn đề
- Các khái niệm
- Deterministic Finite Automata
- Tóm tắt

# Strings

- Một bảng chữ cái là một tập hữu hạn không rỗng các ký tự.
  - Ký hiệu  $\Sigma$
- Một chuỗi trên một bảng chữ cái  $\Sigma$  là một dãy hữu hạn các ký tự từ  $\Sigma$ 
  - Một số chuỗi trên bảng chữ cái  $\Sigma = \{a, b\}$   
a   b   aabbbaaaaa   baabbbbaaaabaaaa
- Chuỗi rỗng là chuỗi không có ký tự, ký hiệu  $\varepsilon$

# Ngôn ngữ

- Một ngôn ngữ hình thức (formal language) là tập hợp các chuỗi.
- $L$  được gọi là một ngôn ngữ trên bảng chữ cái  $\Sigma$  nếu đó là một tập hợp các chuỗi trên  $\Sigma$ 
  - Ngôn ngữ chuỗi đối xứng trên  $\Sigma = \{a, b\}$  là tập hợp  $\{\epsilon, a, b, aa, bb, aaa, aba, bab, bbb, \dots\}$
- Tập hợp của tất cả các chuỗi được tạo từ các ký tự của  $\Sigma$  được ký hiệu là  $\Sigma^*$ 
  - $\Sigma^* = \{a, b\}^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, \dots\}$
- $L$  là một ngôn ngữ trên  $\Sigma$  nếu  $L \subseteq \Sigma^*$

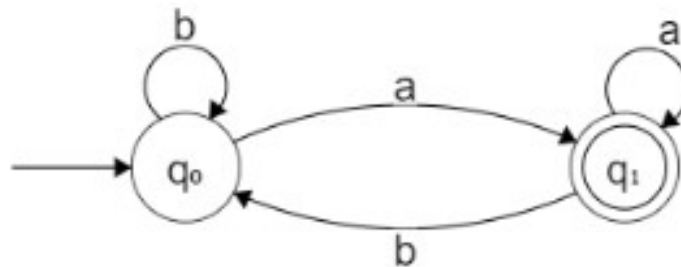
# Các khái niệm





# Automata hữu hạn và ngôn ngữ

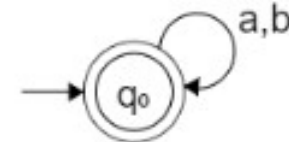
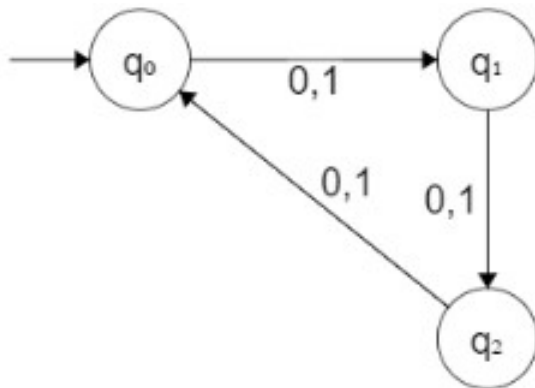
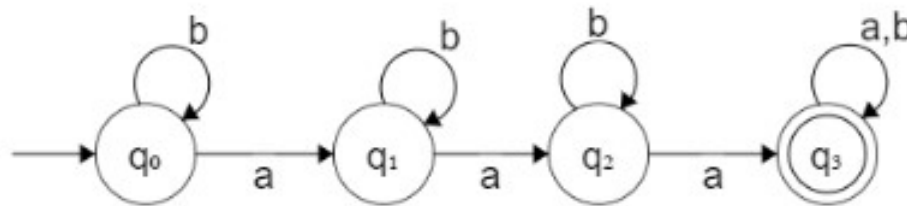
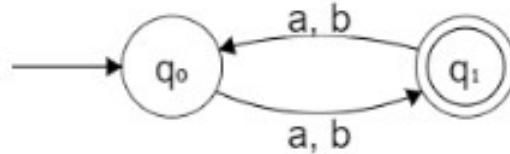
- Cho D là một automaton xử lý các chuỗi trên  $\{a, b\}$



- D chấp nhận các chuỗi tạo ra từ a, b có kết ký tự kết thúc là a và từ chối các chuỗi khác.
- Ngôn ngữ của D

$$\mathcal{L}(D) = \{w \in \{a, b\}^* \mid w \text{ kết thúc là } a\}$$

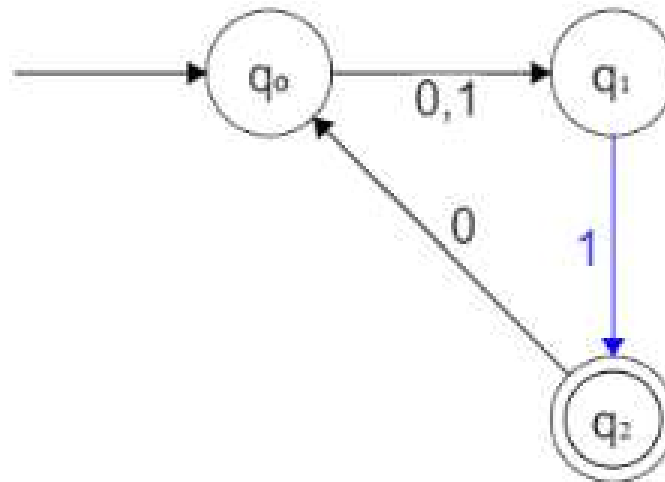
# Automata hữu hạn và ngôn ngữ



$\mathcal{L}(A) = \{w \in \{a, b\}^* \mid A \text{ chấp nhận } w\}$

# Một ví dụ automata hữu hạn

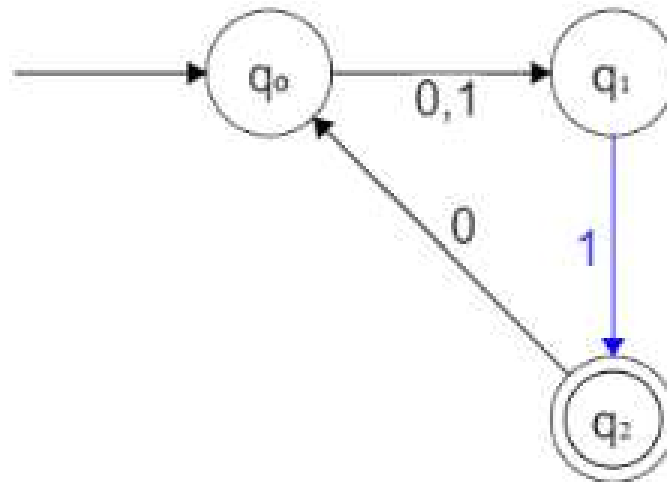
■  $w = 01101$



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

# Một ví dụ automata hữu hạn

■  $w = 01101$

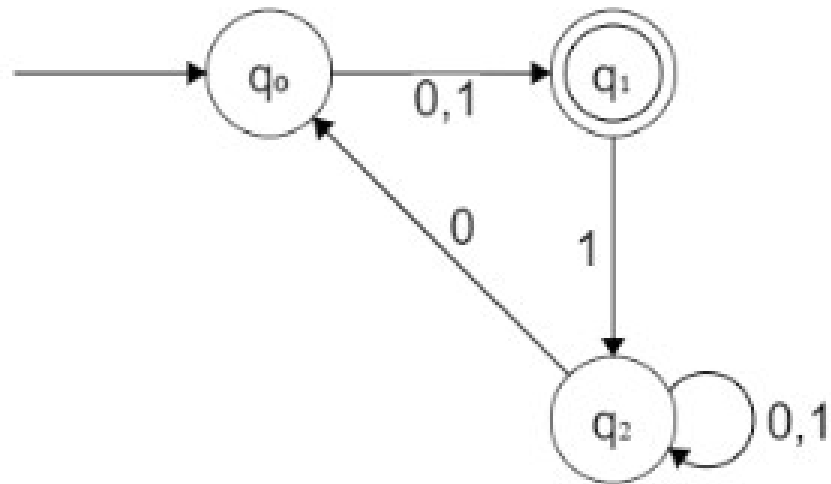


■  $q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1}$



# Một ví dụ automata hữu hạn

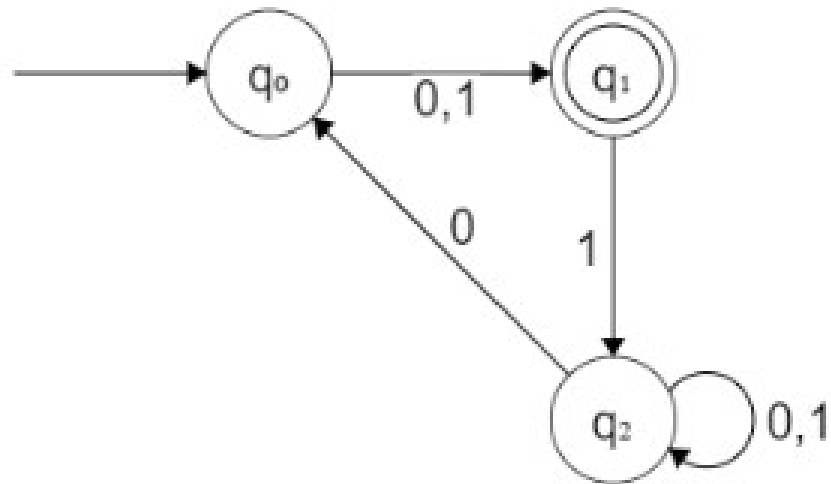
■  $w = 11010$



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

# Một ví dụ automata hữu hạn

■  $w = 11010$



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



---

# Context

- Đặt vấn đề
- Các khái niệm
- Deterministic Finite Automata
- Tóm tắt

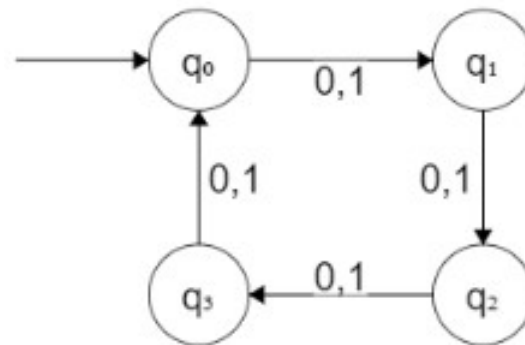
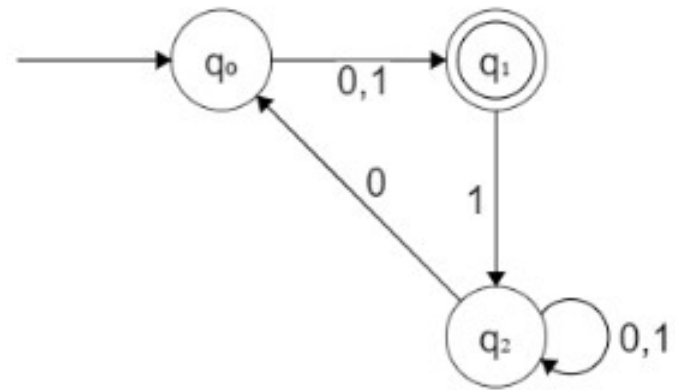
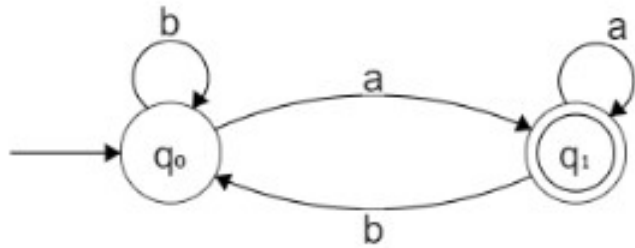
---

# DFA

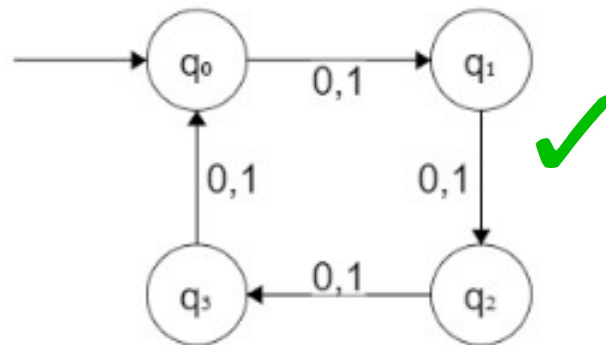
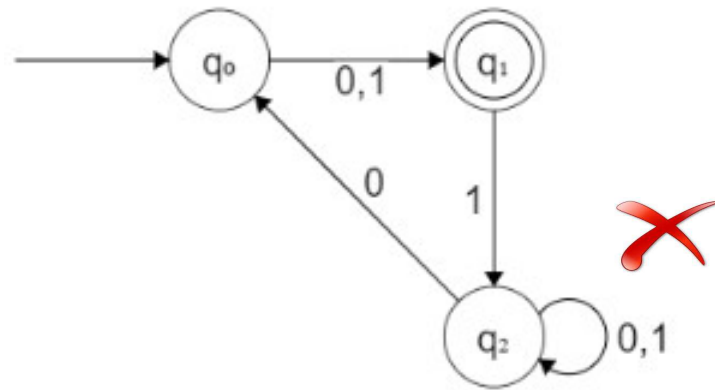
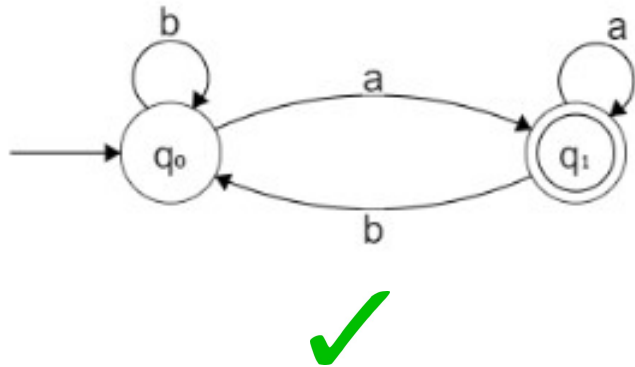
- Một Deterministic Finite Automaton là một automaton hữu hạn trên tập  $\Sigma$
- Mỗi trạng thái của DFA có đúng 1 phép chuyển ứng với mỗi ký tự của  $\Sigma$ 
  - Deterministic
- Có 1 trạng thái bắt đầu
- Có không hoặc nhiều trạng thái kết thúc



# DFA?



# DFA?



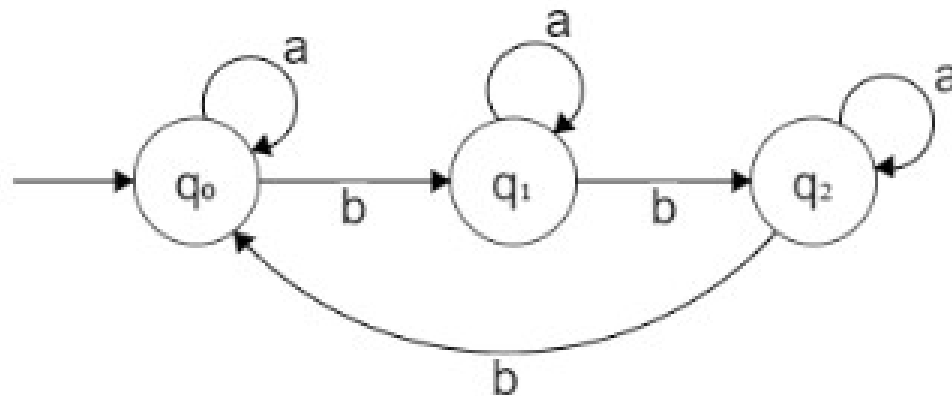
---

# Thiết kế DFA

- Mỗi thời điểm của quá trình tính toán, DFA chỉ nhớ trạng thái ở thời điểm đó
- Gợi ý: Mỗi trạng thái ứng ~ thông tin cần ghi nhớ
  - Mỗi trạng thái được xem như là 1 đối tượng ghi nhớ những gì đã đi qua của những gì sẽ xử lý tiếp theo.
  - Một số hữu hạn các trạng thái khác nhau ~ một số hữu hạn đối tượng khác nhau máy ghi nhớ

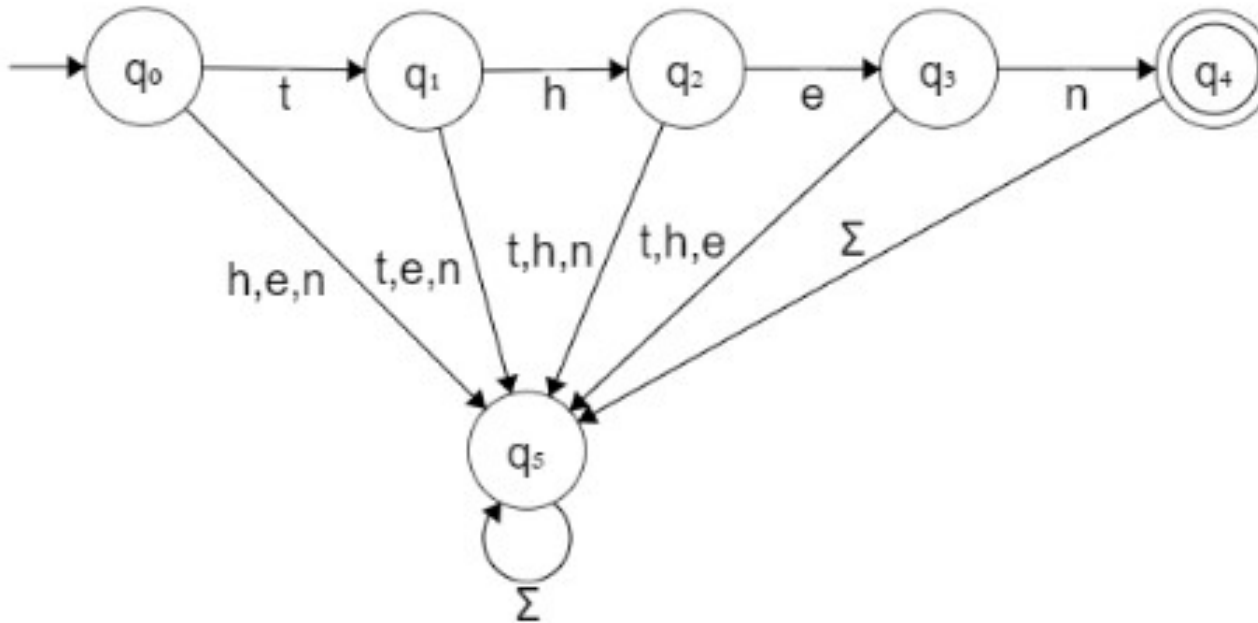
# Nhận biết ngôn ngữ với DFA

- $\mathcal{L}(A) = \{w \in \{a, b\}^* \mid w \text{ có số ký tự } b \text{ chia } 3 \text{ dư } 2\}$

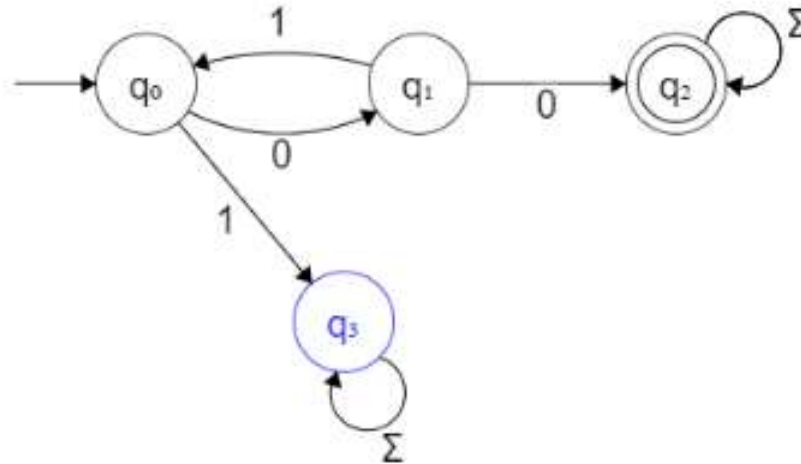


# Nhận biết ngôn ngữ với DFA

- $\mathcal{L}(A) = \{w \in \{t, h, e, n\} \mid w \text{ là từ khóa } \textit{then}\}$



# DFA dạng bảng



	0	1
->q <sub>0</sub>	q <sub>1</sub>	q <sub>3</sub>
q <sub>1</sub>	q <sub>2</sub>	q <sub>0</sub>
*q <sub>2</sub>	q <sub>2</sub>	q <sub>2</sub>
q <sub>3</sub>	q <sub>3</sub>	q <sub>3</sub>

---

# Context

- Đặt vấn đề
- Các khái niệm
- Deterministic Finite Automata
- Tóm tắt

---

# Tóm tắt

- Một automaton là mô hình của một thiết bị tính toán
- Một automaton hữu hạn
  - Mô hình của một thiết bị có bộ nhớ hữu hạn
  - Trả về YES/NO
- DFA ~ mỗi trạng thái có đúng 1 phép chuyển ứng với mỗi ký tự của bảng chữ cái  $\Sigma$
- Ngôn ngữ  $L$  là tập hợp các chuỗi trên tập  $\Sigma$
- DFA ~ ngôn ngữ  $L$