Formal Languages and Automata

Day 2: Finite Automata

Before indulging into DFAs

- Let's review some basic concepts:

- Languages
- Grammars
- Automata

Languages

- Complement $\bar{L} = \Sigma^* - L$
- Reverse $L^R = \{w^R : w \in L\}$
- Concatenation $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$
- L^n : easy $L^0 = \{\lambda\}$

- Star-closure $L^* = L^0 \cup L^1 \cup L^2 \cdots$ [Kleene closure]
- Positive closure $L^+ = L^1 \cup L^2 \cdots$ [Kleene plus]

Grammars

- A **grammar** G is defined as a quadruple

$$G = (V, T, S, P),$$

where V is a finite set of objects called **variables**,
T is a finite set of objects called **terminal symbols**,
$S \in V$ is a special symbol called the **start** variable,
P is a finite set of **productions**.

Sets V, T are nonempty and disjoint.

Grammars – cont'd

- $x \in (V \cup T)^+,\ y \in (V \cup T)^*$

$$w = uxv \xrightarrow{\text{Production rule } x \to y} z = uyv$$

- This is written as $w \Rightarrow z$: w derives z

Grammars – cont'd

- Let $G = (V, T, S, P)$ be a grammar. Then the set

$$L(G) = \left\{ w \in T^* : S \overset{*}{\Rightarrow} w \right\}$$

  is the language generated by G.

- If $w \in L(G)$, then the sequence

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \Rightarrow w$$

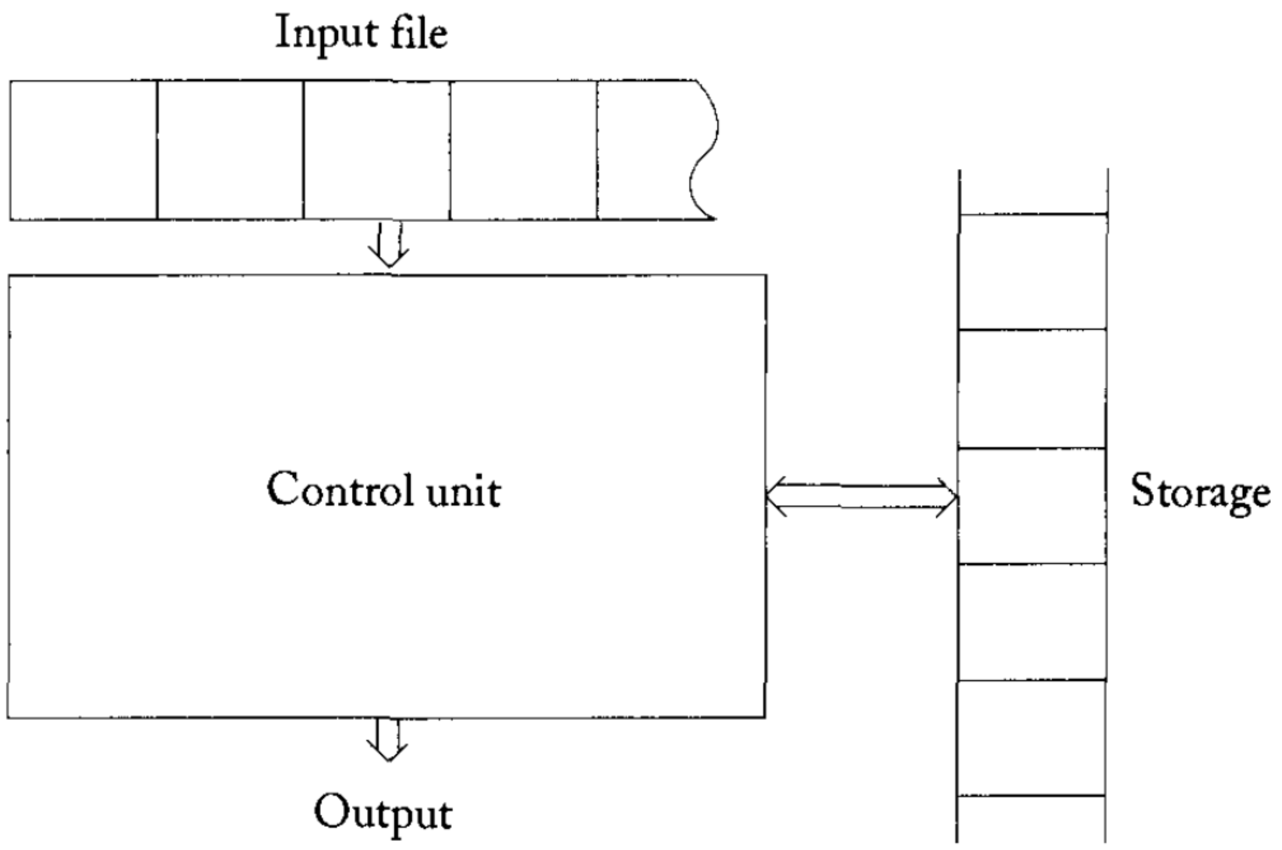
  is a **derivation** of the sentence w.

Automata

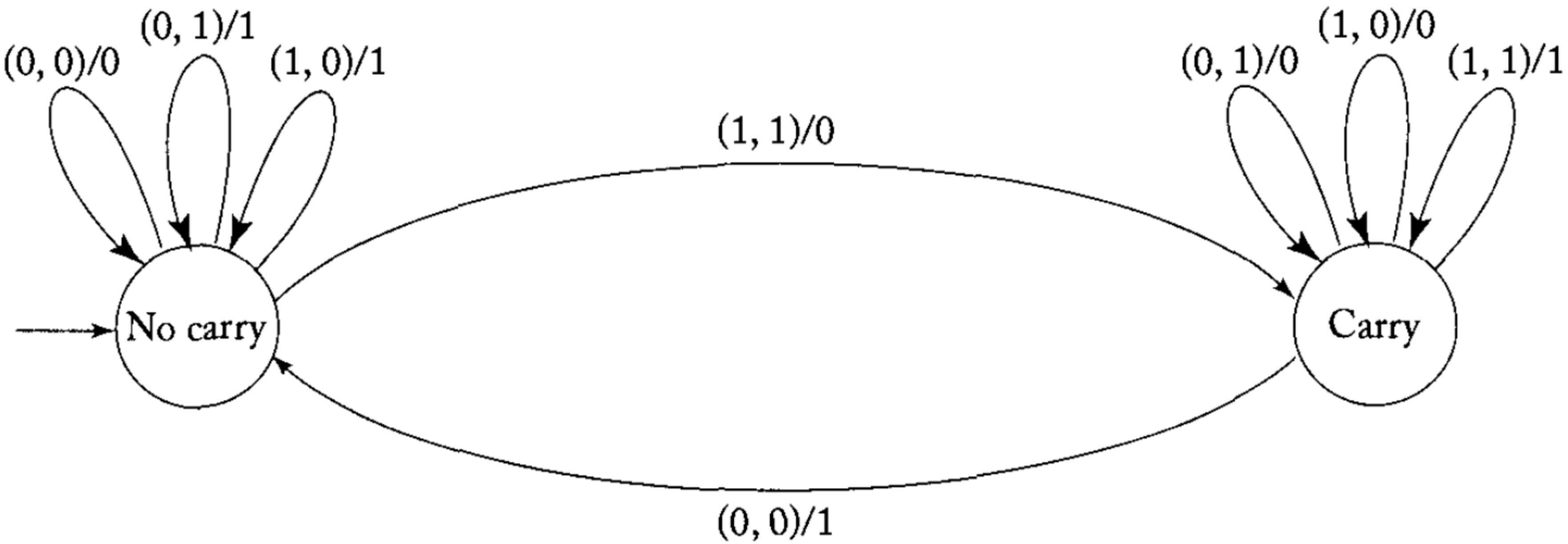
- Input file
- Storage
- Control unit

- **Internal states**
- **Transition function**

- Configuration
- Move



Automata – cont'd



Deterministic Finite Accepters

A **deterministic finite accepter** is defined by the quintuple

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

Q a finite set of **internal states**

Σ a finite set of symbols – **input alphabet**

$\delta\colon Q \times \Sigma \to Q$ a total function – **transition function**

$q_0 \in Q$ the **initial state**

$F \subseteq Q$ a set of **final states**

Internal states

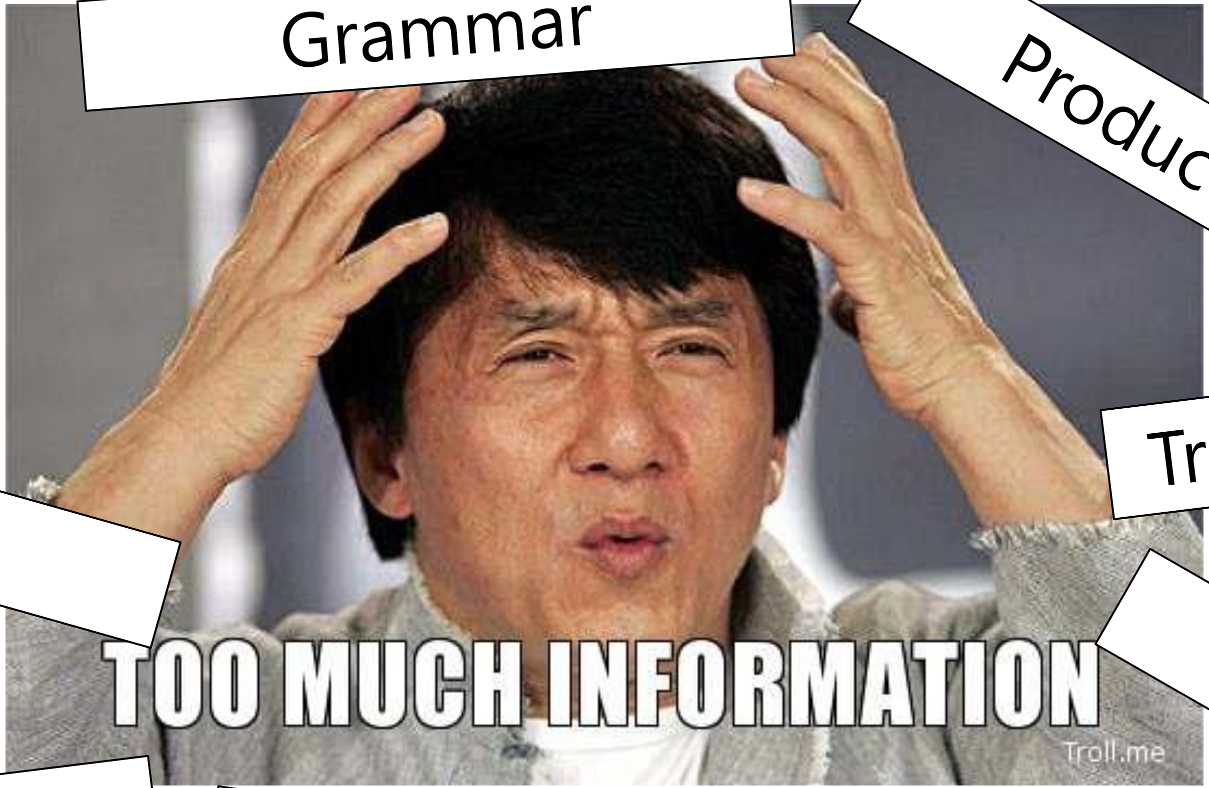
Grammar

Input alphabet

Productions

Initial state

Transition function



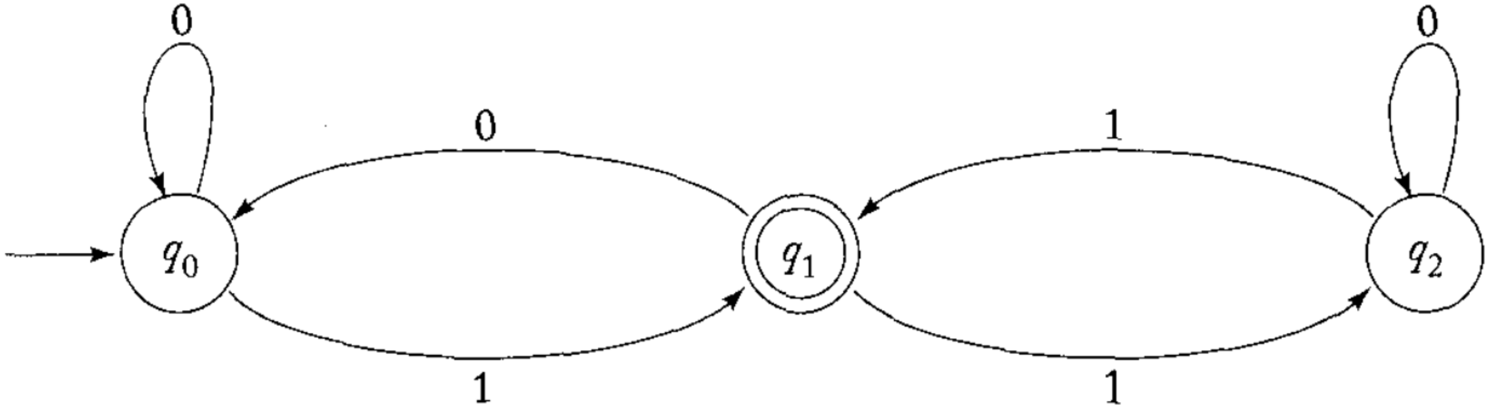
Variables

Terminal symbols

Final states

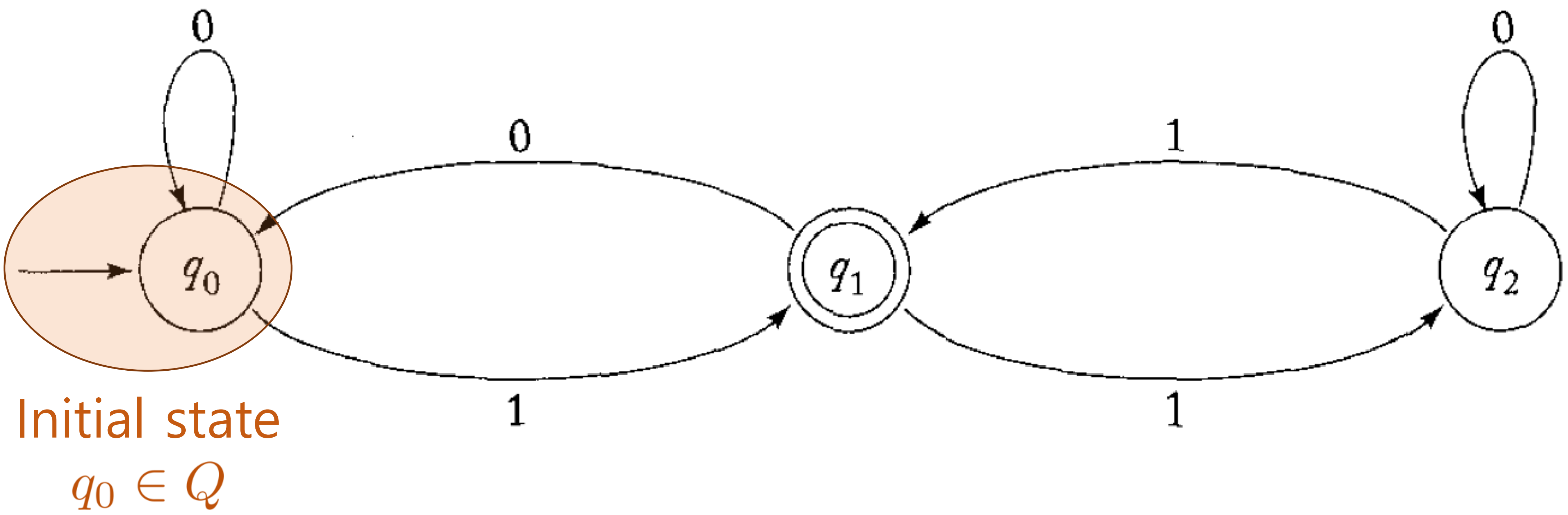
Deterministic Finite Accepters

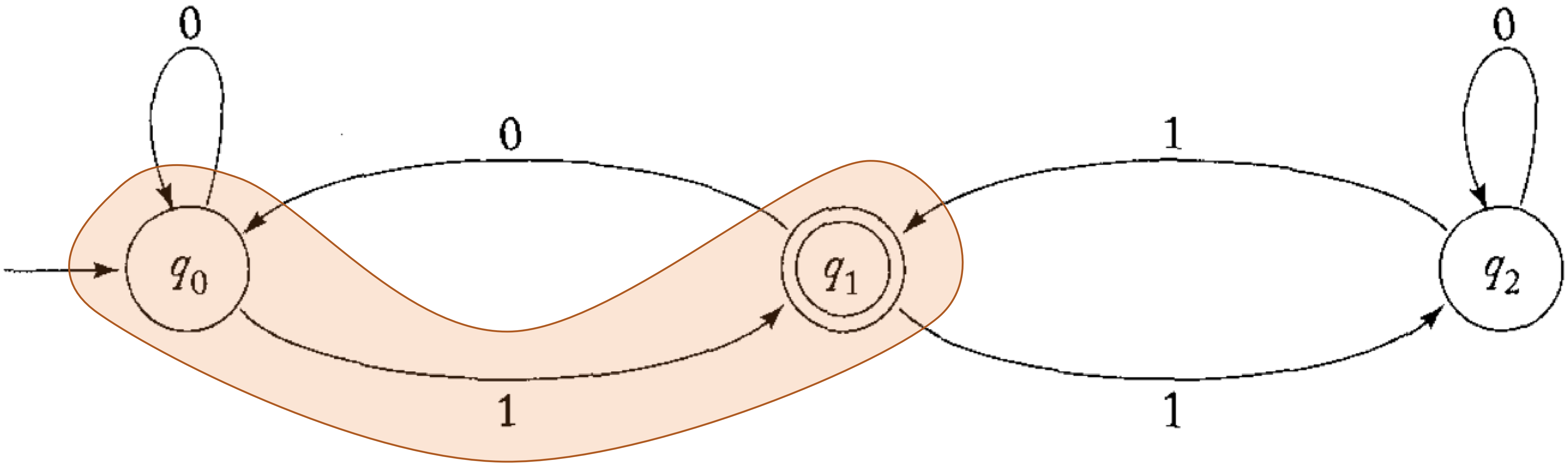
- **Transition graphs** – clear and intuitive



- Vertices = states

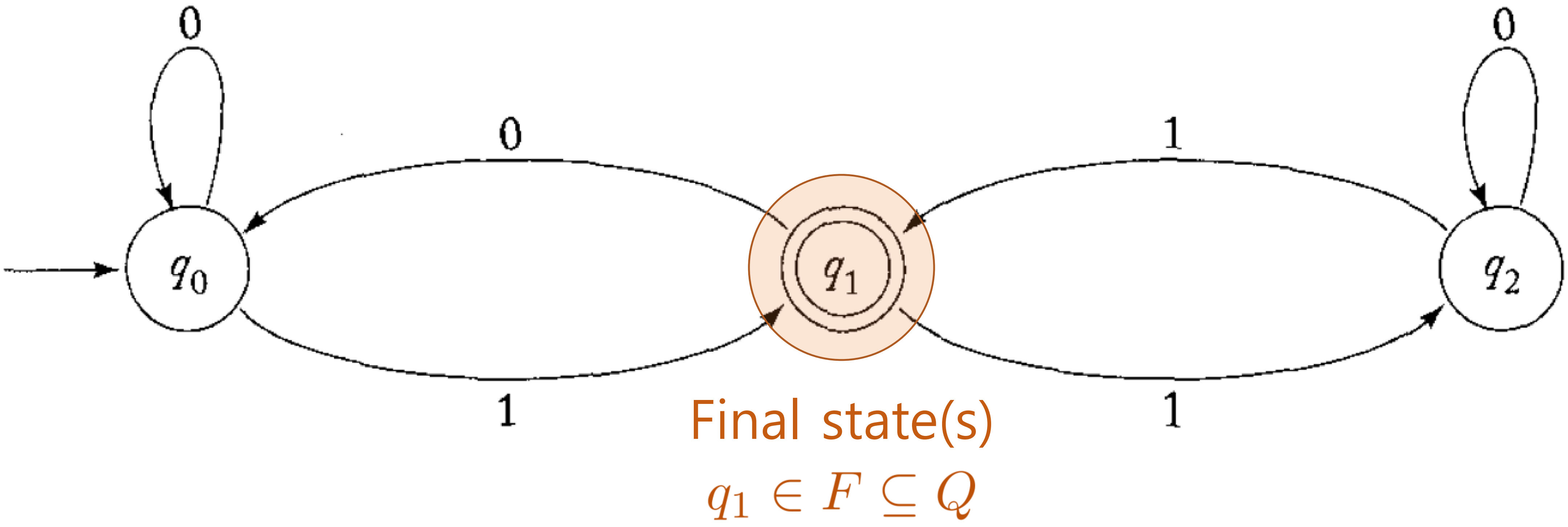
- Edges = transitions

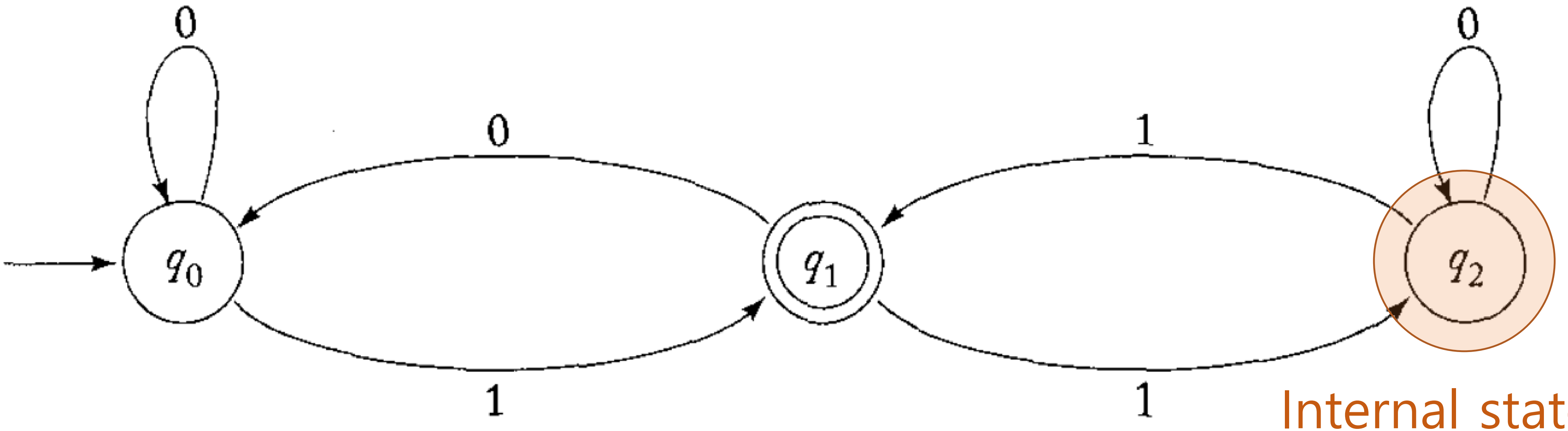

0
0
1
0
q0
q1
q2
1
1
Initial state
$q_0 \in Q$



Transition

$\delta(q_0, 1) = q_1$


0
0
1
0
1
1
Final state(s)
$q_1 \in F \subseteq Q$



Internal state(s)
$q_2 \in Q$

Deterministic Finite Accepters

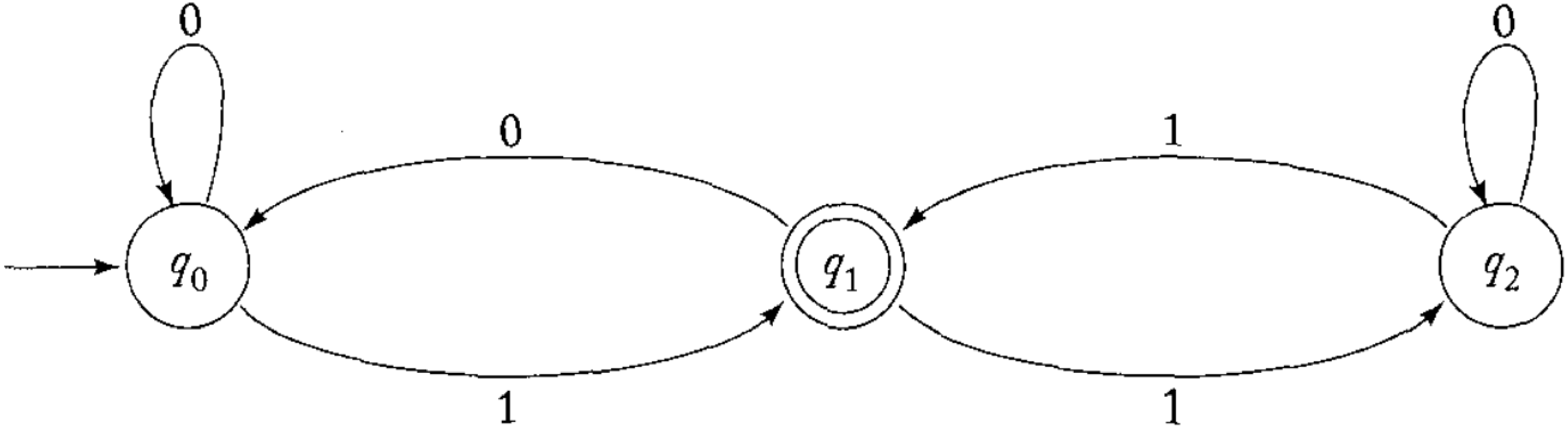
- Deterministic finite accepter $M = (Q, \Sigma, \delta, q_0, F)$

  $\Leftrightarrow$ Transition graph G_M

[Example 1] $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$
where δ is given by

$$\begin{aligned}
\delta(q_0, 0) &= q_0, & \delta(q_0, 1) &= q_1, \\
\delta(q_1, 0) &= q_0, & \delta(q_1, 1) &= q_2, \\
\delta(q_2, 0) &= q_2, & \delta(q_2, 1) &= q_1.
\end{aligned}$$

Deterministic Finite Accepters



Deterministic Finite Accepters

- Extended transition function $\delta^*\colon Q \times \Sigma^* \to Q$.

[Idea] $\delta(q_0, a) = q_1,\ \delta(q_1, b) = q_2 \Rightarrow \delta^*(q_0, ab) = q_2.$

Formally, we can define δ^* recursively by

$$\begin{cases} \delta^*(q, \lambda) = q \\ \delta^*(q, wa) = \delta(\delta(q, w), a) \end{cases} \qquad (q \in Q,\ w \in \Sigma^*,\ a \in \Sigma)$$

Languages and DFAs

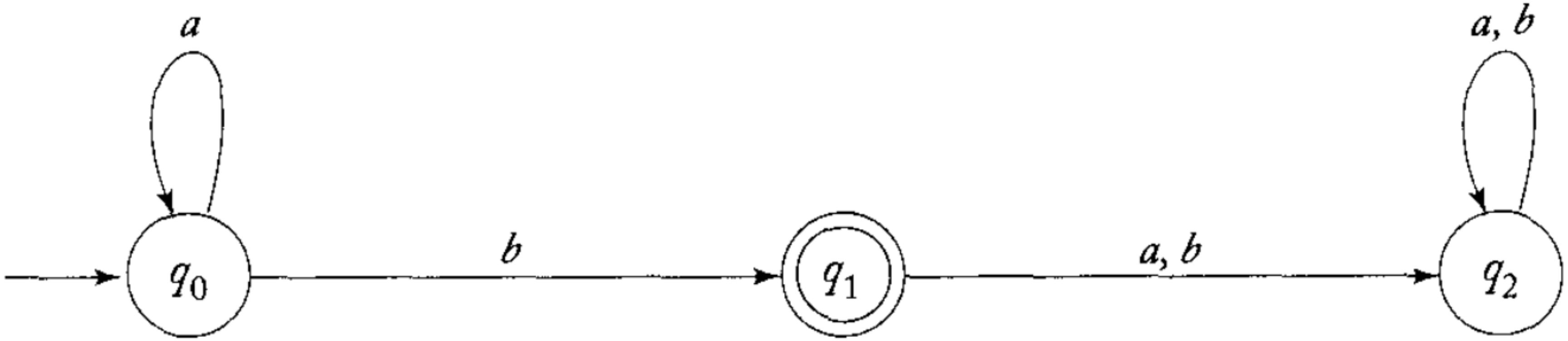
- The language accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is the set of all strings on Σ accepted by M.

- Formally,

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}.$$

Languages and DFAs

[Example 2] Find $L(M)$.

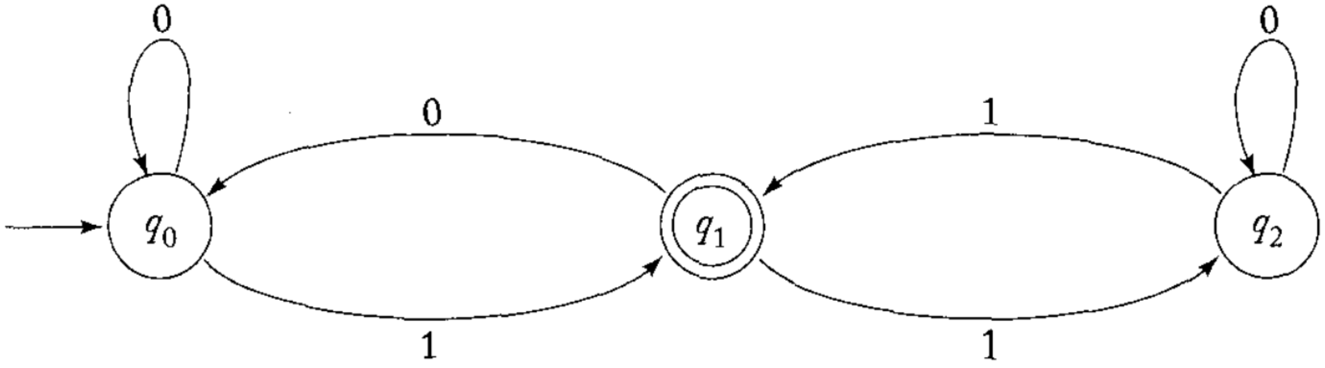


Languages and DFAs

Theorem

For every $q_i, q_j \in Q$, $w \in \Sigma^+$,
$\delta^*(q_i, w) = q_j$ if and only if there is in G_M [a walk with label w from q_i to q_j.]

<u>Proof</u> – Simple induction.



Languages and DFAs

[Example 3] Find a deterministic finite accepter that recognizes the set of all strings on $\Sigma = \{a, b\}$ with the prefix ab.

Languages and DFAs

[Example 4] Find a DFA that accepts all the strings on $\{0, 1\}$, except those containing the substring 001.

Regular Languages

- L is **regular** $\Leftrightarrow$ there exists some DFA M s.t. $L = L(M)$.

[Example 5] Show that the language

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

is regular.

– How about $L^2, L^3, \cdots$?

Nondeterministic Finite Accepters

A **deterministic finite accepter** is defined by the quintuple

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

- Q a finite set of **internal states**
- Σ a finite set of symbols – **input alphabet**
- δ: $Q \times \Sigma \rightarrow Q$ a total function – **transition function**
- $q_0 \in Q$ the **initial state**
- $F \subseteq Q$ a set of **final states**

Nondeterministic Finite Accepters

A **nondeterministic finite accepter** is defined by the quintuple

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

Q a finite set of **internal states**

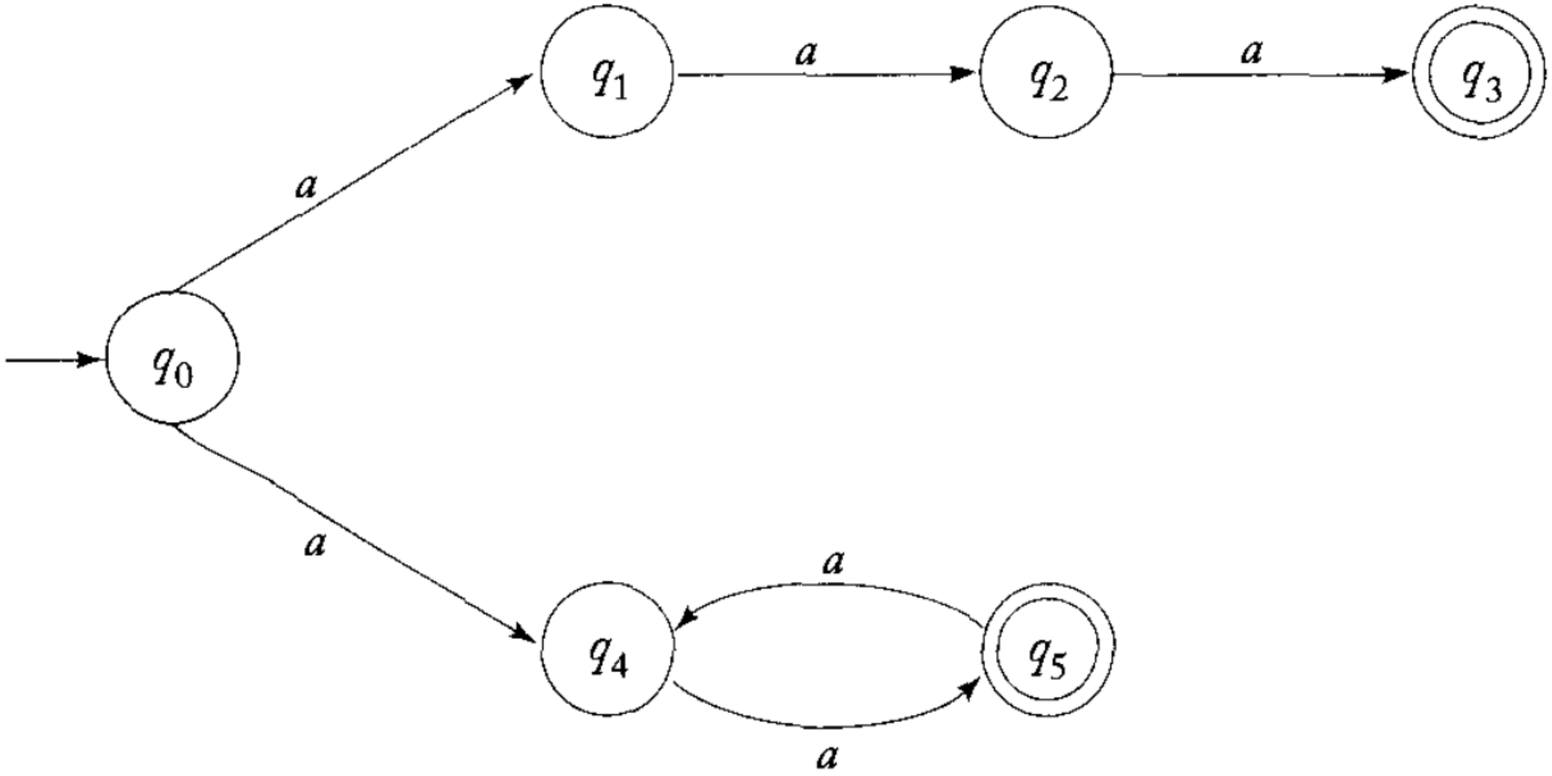
Σ a finite set of symbols – **input alphabet**

$\delta \colon Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$ – **transition function**

$q_0 \in Q$ the **initial state**

$F \subseteq Q$ a set of **final states**

Nondeterministic Finite Accepters



Nondeterministic Finite Accepters

- Extended transition function for NFA

$$\delta^*(q_i, w) = ?$$

$q_j \in \delta^*(q_i, w) \Leftrightarrow$ There is a walk in the transition graph from q_i to q_j labeled w.

Languages and NFAs

- The language accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is the set of all strings on Σ accepted by M.

- Formally,

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}.$$

Languages and NFAs

- The language accepted by a NFA $M = (Q, \Sigma, \delta, q_0, F)$ is the set of all strings on Σ accepted by M.

- Formally,

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F \neq \varnothing\}.$$

Equivalence of Finite Accepters

- M_1, M_2 are equivalent if $L(M_1) = L(M_2)$

- In what sense are DFAs and NFAs different?

procedure: nfa_to_dfa

1. Create a graph G_D with vertex $\{q_0\}$. Identify this vertex as the initial vertex.

2. Repeat the following steps until no more edges are missing.

Take any vertex $\{q_i, q_j, ..., q_k\}$ of G_D that has no outgoing edge for some $a \in \Sigma$.

Compute $\delta^*(q_i, a), \delta^*(q_j, a) ..., \delta^*(q_k, a)$.

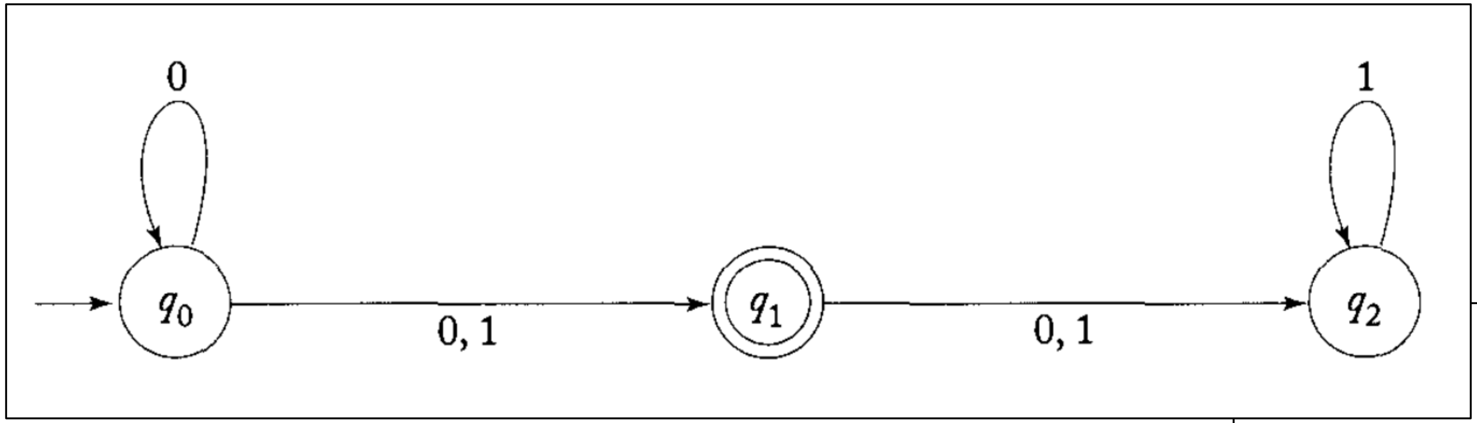
Then form the union of all these δ^*, yielding the set $\{q_l, q_m, ..., q_n\}$.

Create a vertex for G_D labeled $\{q_l, q_m, ..., q_n\}$ if it does not already exist.

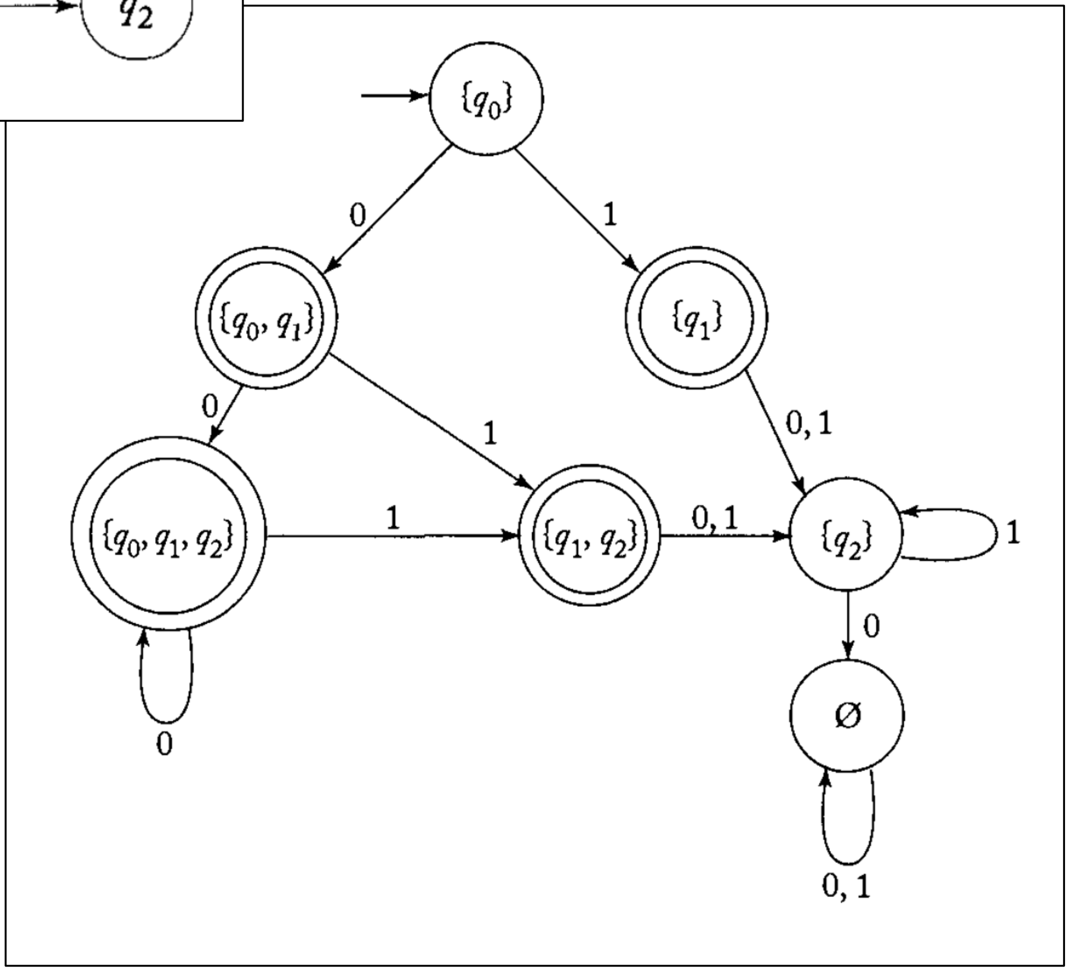
Add to G_D an edge from $\{q_i, q_j, ..., q_k\}$ to $\{q_l, q_m, ..., q_n\}$ and label it with a.

3. Every state of G_D whose label contains any $q_f \in F_N$ is identified as a final vertex.

4. If M_N accepts λ, the vertex $\{q_0\}$ in G_D is also made a final vertex.



"NFA-to-DFA"



Next seminar

- Regular Languages
- Regular Expressions
- Regular Grammars

- Some programming