Finite Automata

Alphabets and Languages

Recall

Alphabets and Languages

Recall

Σ - alphabet (finite set)

$\Sigma^n = \{a_1 a_2 .. a_n \mid a_i \in \Sigma\}$ is the set of words of length n

$\Sigma^* = \{w \mid \exists n \in \mathbb{N}. \exists\, a_1, a_2, .. , a_n \in \Sigma. w = a_1 a_2 .. a_n\}$ is the set of all words over Σ

Alphabets and Languages

Recall

$\Sigma^0 = \{\varepsilon\}$ contains only the empty word

Σ - alphabet (finite set)

$\Sigma^n = \{a_1a_2..a_n \mid a_i \in \Sigma\}$ is the set of words of length n

$\Sigma^* = \{w \mid \exists n \in \mathbb{N}.\ \exists\ a_1, a_2, .., a_n \in \Sigma.\ w = a_1a_2..a_n\}$ is the set of all words over Σ

Alphabets and Languages

Recall

$\Sigma^0 = \{\varepsilon\}$ contains only the empty word

Σ - alphabet (finite set)

$\Sigma^n = \{a_1 a_2 .. a_n \mid a_i \in \Sigma\}$ is the set of words of length n

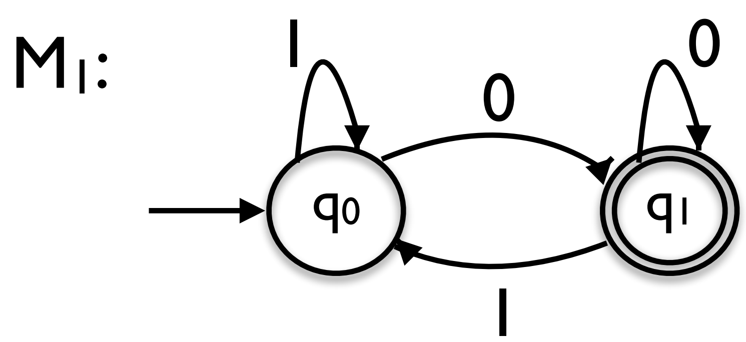
$\Sigma^* = \{w \mid \exists n \in \mathbb{N}.\ \exists\, a_1, a_2, .., a_n \in \Sigma.\ w = a_1 a_2 .. a_n\}$ is the set of all words over Σ

A language L over Σ is a subset $L \subseteq \Sigma^*$

Deterministic Automata (DFA)

Informal example

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



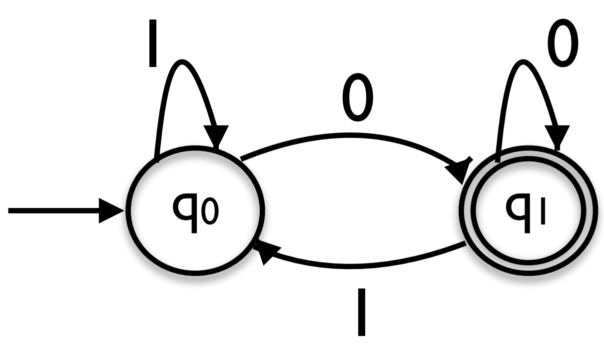
Deterministic Automata (DFA)

alphabet

Informal example

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

M_1:



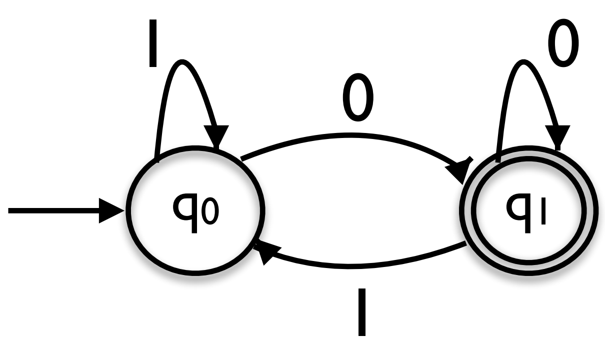
Deterministic Automata (DFA)

alphabet

Informal example

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

M_1:



q_0, q_1 are states

Deterministic Automata (DFA)

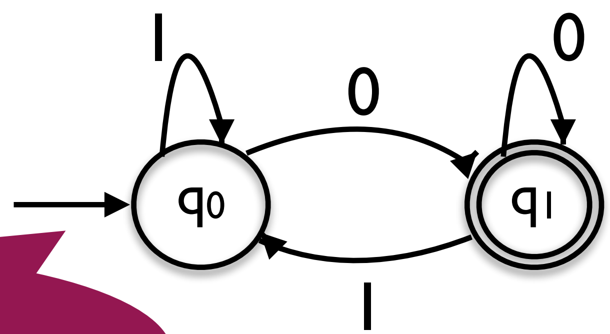
alphabet

Informal example

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

q_0, q_1 are states

M_1:



q_0 is initial

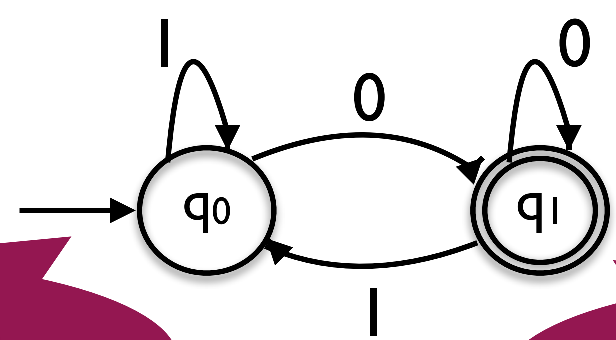
Deterministic Automata (DFA)

alphabet

Informal example

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

M_1:



q_0, q_1 are states

q_0 is initial

q_1 is final

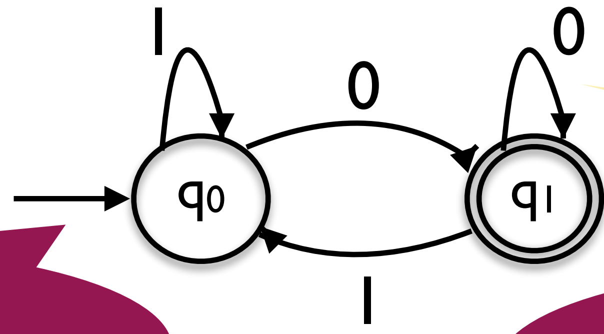
Deterministic Automata (DFA)

alphabet

Informal example

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

M_1:



q_0, q_1 are states

transitions, labelled by alphabet symbols

q_0 is initial

q_1 is final

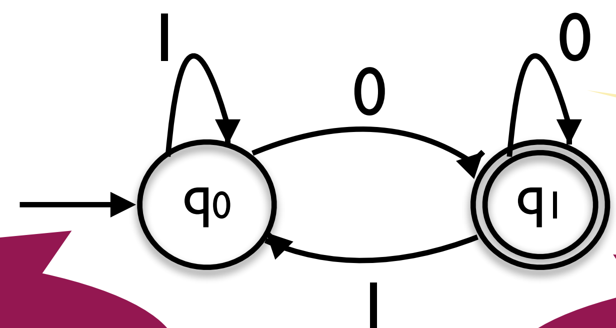
Deterministic Automata (DFA)

alphabet

Informal example

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

M_1:



q_0, q_1 are states

transitions, labelled by alphabet symbols

q_0 is initial

q_1 is final

Accepts the language $L(M_1) = \{w \in \Sigma^* \mid w \text{ ends with a } 0\} = \Sigma^* 0$

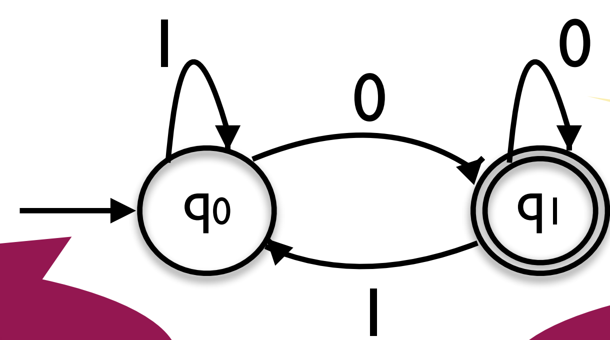
Deterministic Automata (DFA)

Informal example

alphabet

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

M_1:



q_0, q_1 are states

transitions, labelled by alphabet symbols

q_0 is initial

q_1 is final

Accepts the language $L(M_1) = \{w \in \Sigma^* \mid w \text{ ends with a } 0\} = \Sigma^*0$

regular language

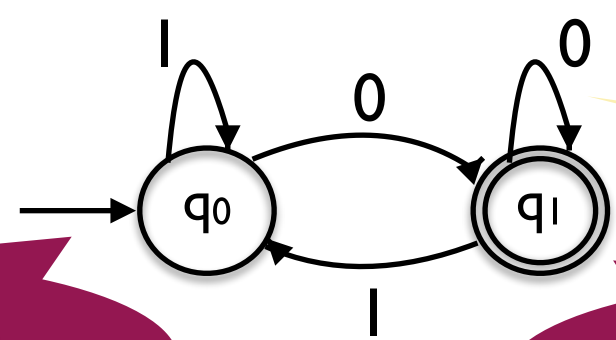
Deterministic Automata (DFA)

Informal example

alphabet

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

M_1:



q_0, q_1 are states

transitions, labelled by alphabet symbols

q_0 is initial

q_1 is final

Accepts the language $L(M_1) = \{w \in \Sigma^* \mid w \text{ ends with a } 0\} = \Sigma^* 0$

regular language

regular expression

DFA

Definition

A deterministic automaton M is a tuple M = $(Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

DFA

Definition

A deterministic automaton M is a tuple M = $(Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

$Q = \{q_0, q_1\}$

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

$Q = \{q_0, q_1\}$

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

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

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

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

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

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

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

$\delta(q_0, 0) = q_1$, $\delta(q_0, 1) = q_0$
$\delta(q_1, 0) = q_1$, $\delta(q_1, 1) = q_0$

DFA

The extended transition function

DFA

The extended transition function

Given $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma \longrightarrow Q$ to

$\delta^*: Q \times \Sigma^* \longrightarrow Q$

inductively, by:

$\delta^*(q, \varepsilon) = q$ and $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

DFA

The extended transition function

Given $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma \longrightarrow Q$ to

$\delta^*: Q \times \Sigma^* \longrightarrow Q$

In M_1, $\delta^*(q_0, 110010) = q_1$

inductively, by:

$\delta^*(q, \varepsilon) = q$ and $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

DFA

The extended transition function

Given $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma \longrightarrow Q$ to

$$\delta^*: Q \times \Sigma^* \longrightarrow Q$$

In M_1, $\delta^*(q_0, 110010) = q_1$

inductively, by:

$$\delta^*(q, \varepsilon) = q \quad \text{and} \quad \delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

Definition

The language recognised / accepted by a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ is

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

DFA

The extended transition function

Given $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma \longrightarrow Q$ to

$\delta^*: Q \times \Sigma^* \longrightarrow Q$

In M_1, $\delta^*(q_0, 110010) = q_1$

inductively, by:

$\delta^*(q, \varepsilon) = q$ and $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

Definition

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$

The language recognised / accepted by a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ is

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

Regular languages and operations

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular languages and operations

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$
is regular

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular languages and operations

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$
is regular

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular operations

Regular languages and operations

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$
is regular

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular operations

Let L, L_1, L_2 be languages over Σ. Then
$L_1 \cup L_2$, $L_1 \cdot L_2$, and L^* are languages, where

$$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

$$L^* = \{w \mid \exists n \in \mathbb{N}. \exists\, w_1, w_2, .., w_n \in L.\ w = w_1 w_2 .. w_n\}$$

Regular languages and operations

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$
is regular

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular operations

Let L, L_1, L_2 be languages over Σ. Then $L_1 \cup L_2$, $L_1 \cdot L_2$, and L^* are languages, where

$$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

$$L^* = \{w \mid \exists n \in \mathbb{N}.\ \exists\, w_1, w_2, .., w_n \in L.\ w = w_1 w_2 .. w_n\}$$

$\varepsilon \in L^*$ always

Closure under regular operations

Closure under regular operations

Theorem C1

The class of regular languages is closed under union

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

Theorem C2

The class of regular languages is closed under complement

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

Theorem C4

The class of regular languages is closed under Kleene star

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

We can already prove these!

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

Theorem C4

The class of regular languages is closed under Kleene star

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

We can already prove these!

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

But not yet these two…

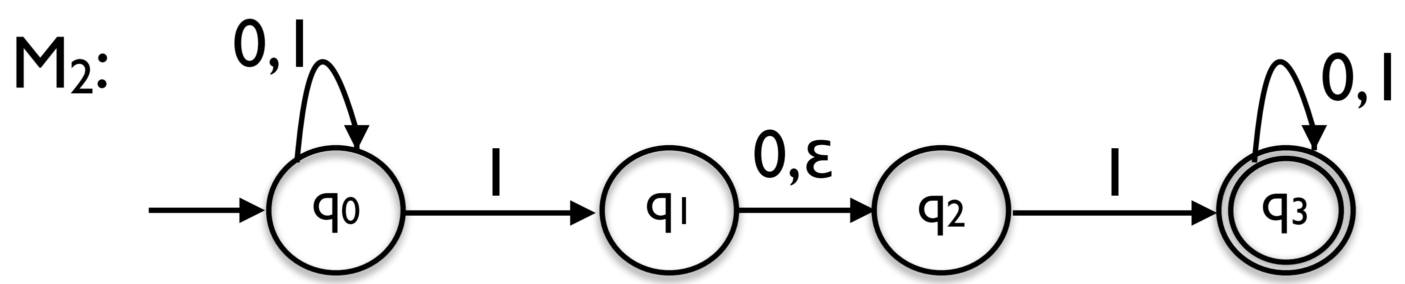
Theorem C4

The class of regular languages is closed under Kleene star

Nondeterministic Automata (NFA)

Informal example

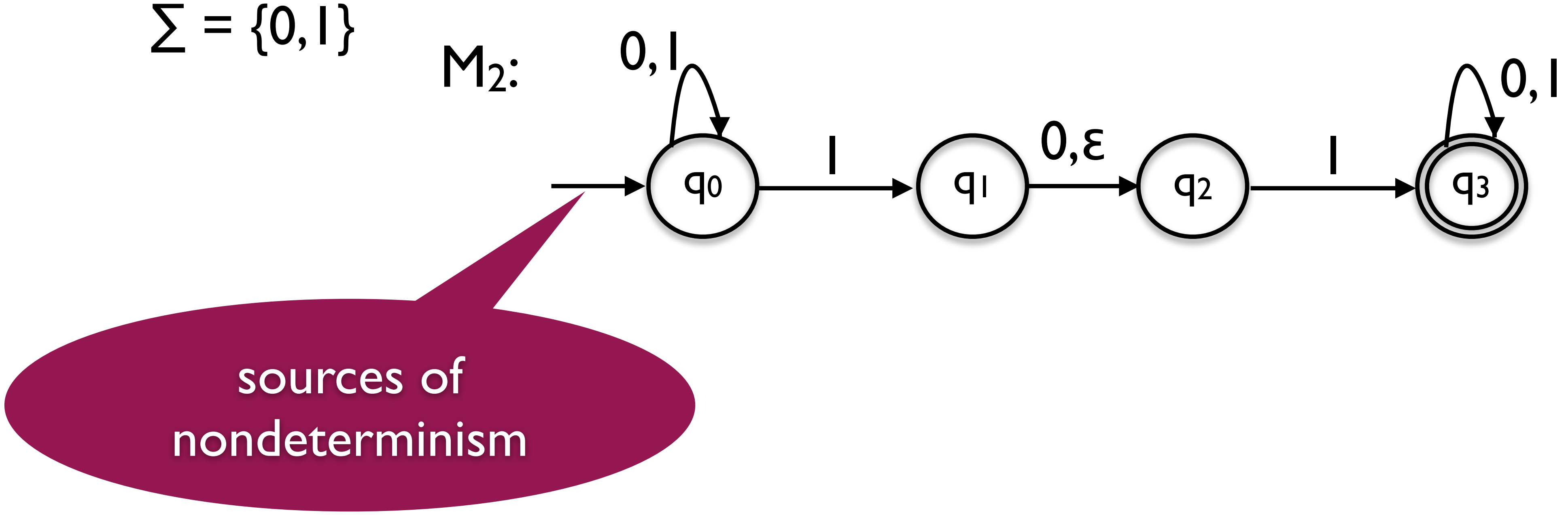
$\Sigma = \{0,1\}$



Nondeterministic Automata (NFA)

Informal example

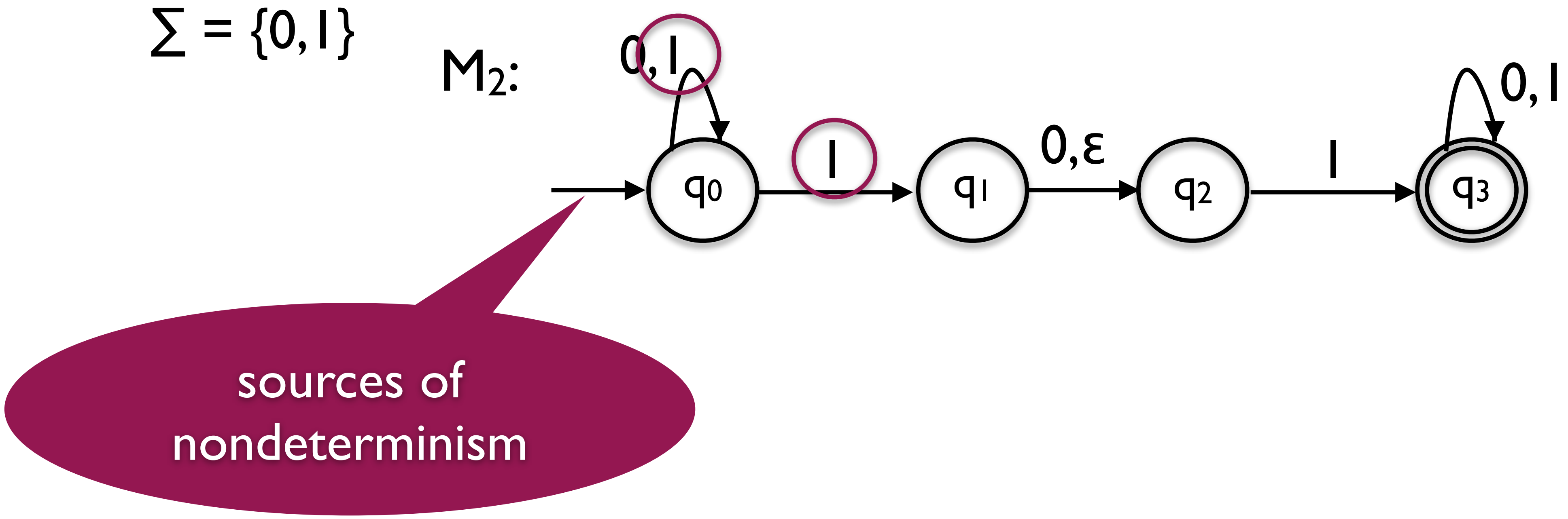
$\Sigma = \{0,1\}$



Nondeterministic Automata (NFA)

Informal example

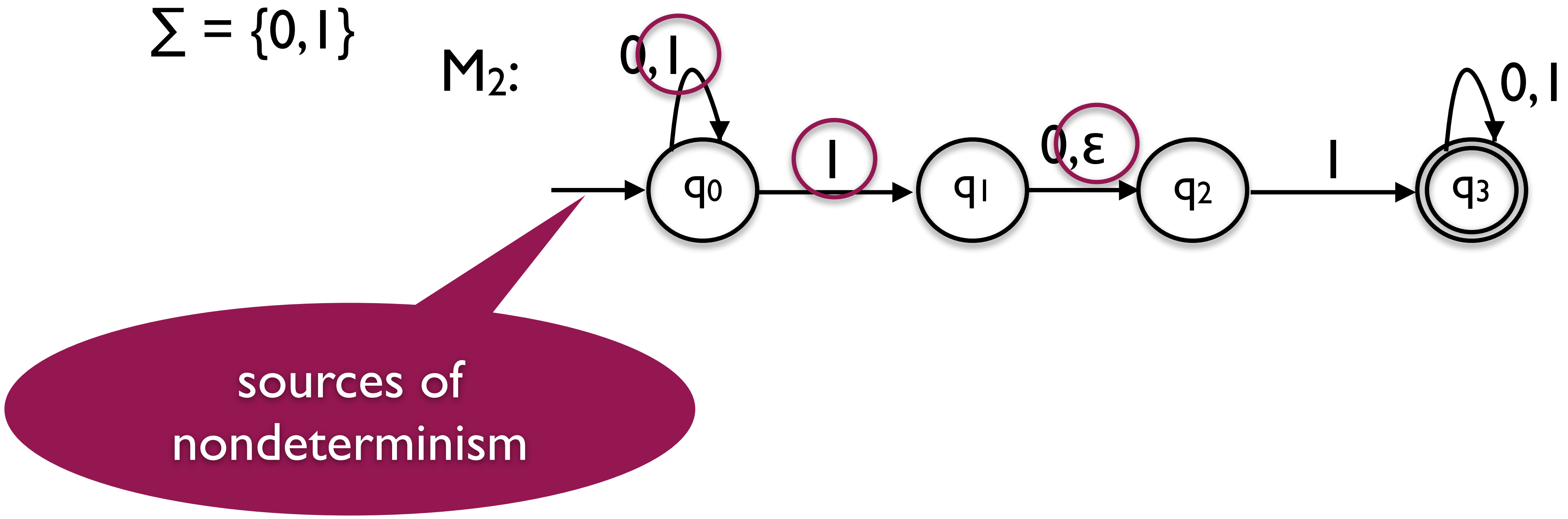
$\Sigma = \{0,1\}$



Nondeterministic Automata (NFA)

Informal example

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

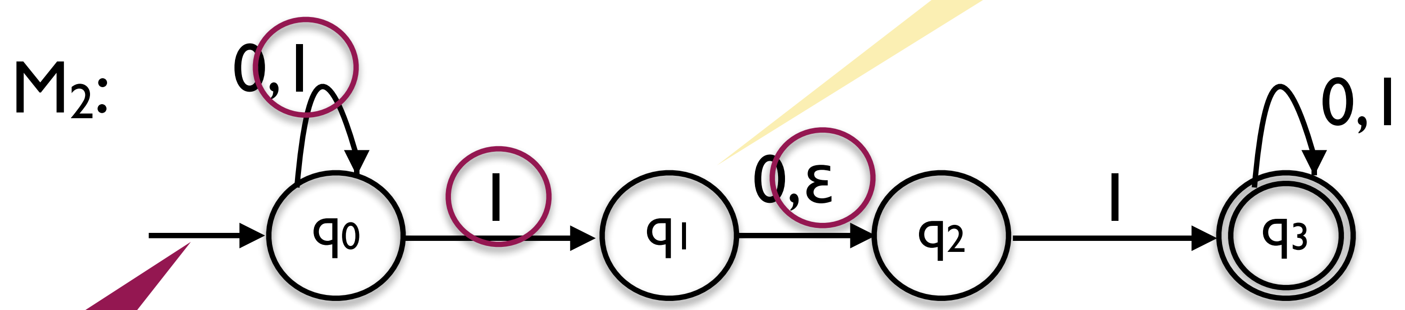


Nondeterministic Automata (NFA)

no 1 transition

Informal example

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

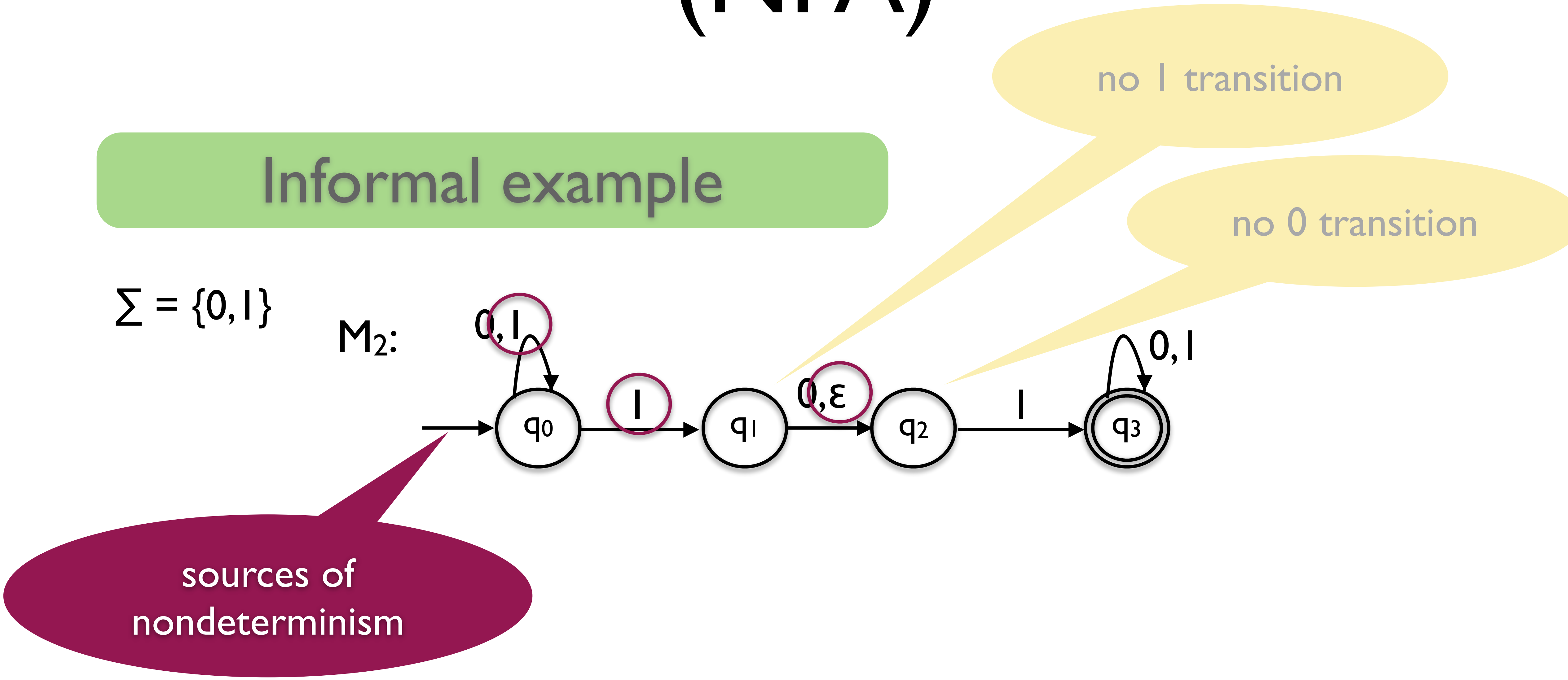


sources of nondeterminism

Nondeterministic Automata (NFA)

Informal example

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



Nondeterministic Automata (NFA)

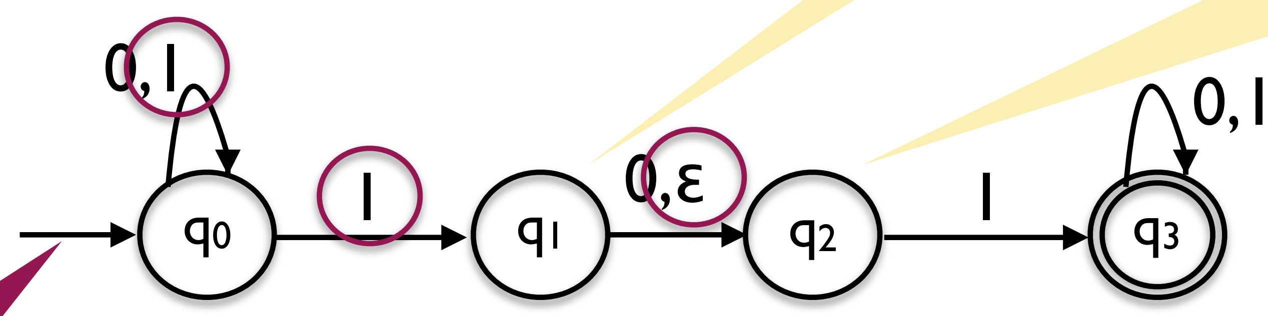
no 1 transition

Informal example

no 0 transition

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

M_2:



sources of nondeterminism

Accepts a word iff there exists an accepting run

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

NFA

Definition

A nondeterministic automaton M is a tuple M = $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
- q_0 is the initial state, $q_0 \in Q$
- F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

NFA

Definition

A nondeterministic automaton M is a tuple M = $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
- q_0 is the initial state, $q_0 \in Q$
- F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

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

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

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

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

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

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

$\Sigma = \{0, 1\}$ $F = \{q_3\}$

NFA

Definition

A nondeterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

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

$\Sigma = \{0, 1\}$ $F = \{q_3\}$

$\delta(q_0, 0) = \{q_0\}$
$\delta(q_0, 1) = \{q_0, q_1\}$
$\delta(q_0, \varepsilon) = \varnothing$
…..

NFA

The extended transition function

NFA

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$

inductively, by:

$\delta^*(q, \varepsilon) = E(q)$ and $\delta^*(q,wa) = E(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a))$

NFA

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n\text{-}1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

inductively, by:

$$\delta^*(q, \varepsilon) = E(q) \quad \text{and} \quad \delta^*(q, wa) = E\Big(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a)\Big)$$

ε-closure of q, all states reachable by ε-transitions from q

NFA

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon),\ \text{for } i = 0, .., n-1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

inductively, by:

$$\delta^*(q, \varepsilon) = E(q) \quad \text{and} \quad \delta^*(q, wa) = E\left(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a)\right)$$

NFA

ε-closure of q, all states reachable by ε-transitions from q

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n-1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

$E(X) = \bigcup_{x \in X} E(x)$

inductively, by:

$$\delta^*(q, \varepsilon) = E(q) \quad \text{and} \quad \delta^*(q, wa) = E\left(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a)\right)$$

NFA

ε-closure of q, all states reachable by ε-transitions from q

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n-1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$

inductively, by:

$\delta^*(q, \varepsilon) = E(q)$ and $\delta^*(q, wa) = E(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a))$

$E(X) = \bigcup_{x \in X} E(x)$

In M_2, $\delta^*(q_0, 0110) = \{q_0, q_2, q_3\}$

ε-closure of q, all states reachable by ε-transitions from q

NFA

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n-1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$\delta^* : Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$

$E(X) = \bigcup_{x \in X} E(x)$

In M_2, $\delta^*(q_0, 0110) = \{q_0, q_2, q_3\}$

inductively, by:

$\delta^*(q, \varepsilon) = E(q)$ and $\delta^*(q, wa) = E(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a))$

Definition

The language recognised / accepted by a nondeterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ is

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$$

ε-closure of q, all states reachable by ε-transitions from q

NFA

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n-1\}$

The extended transition function

Given an NFA M = $(Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$

$E(X) = \bigcup_{x \in X} E(x)$

In M_2, $\delta^*(q_0, 0110) = \{q_0, q_2, q_3\}$

inductively, by:

$\delta^*(q, \varepsilon) = E(q)$ and $\delta^*(q, wa) = E(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a))$

Definition

The language recognised / accepted by a ... automaton M = $(Q, \Sigma, \delta, q_0, F)$ is

$L(M_2) = \{u101w \mid u,w \in \{0,1\}^*\} \cup \{u11w \mid u,w \in \{0,1\}^*\}$

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