Introduction to Finite Automata

Dr. Chuck Rocca
roccac@wcsu.edu

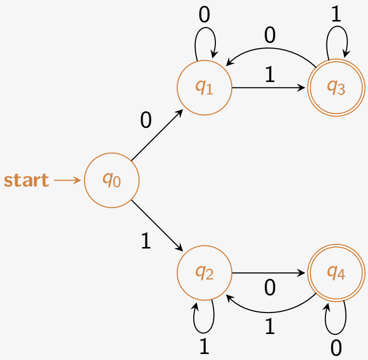
http://sites.wcsu.edu/roccac

WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES

Table of Contents

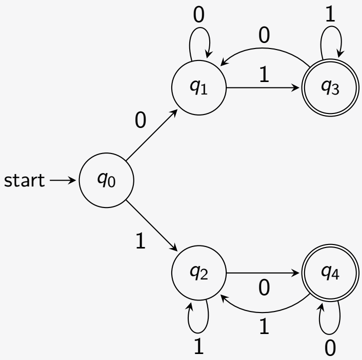
Table of Contents

State Diagram and Finite Automaton



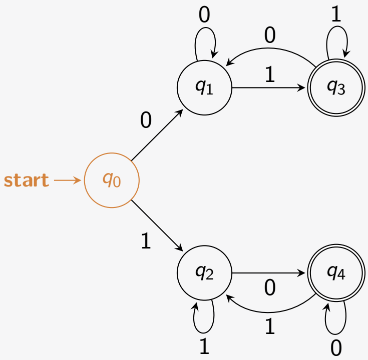
- **States:** $Q = \{q_0, q_1, q_2, q_3, q_4\}$

State Diagram and Finite Automaton



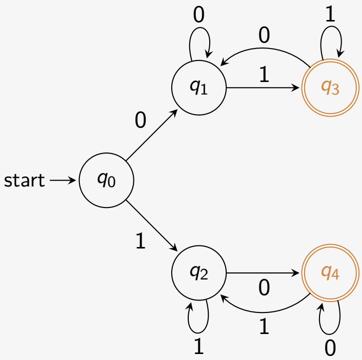
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- **Alphabet:** $\Sigma = \{0, 1\}$

State Diagram and Finite Automaton



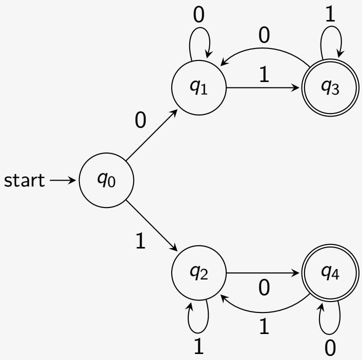
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- **Start State:** q_0

State Diagram and Finite Automaton



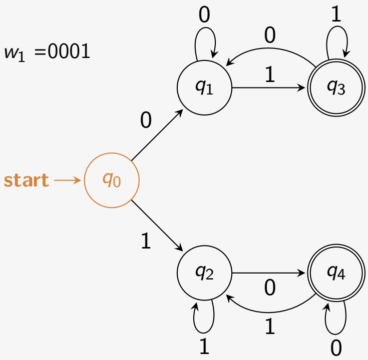
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- **Accept States:** $F = \{q_3, q_4\}$

State Diagram and Finite Automaton



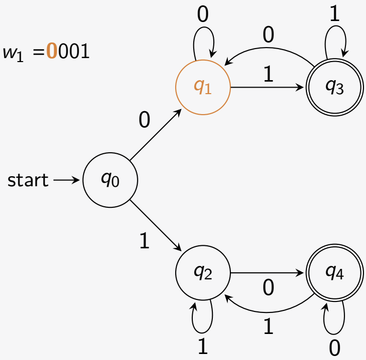
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- **Language:** $L = \{1 * 0, 0 * 1\}$

State Diagram and Finite Automaton



- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

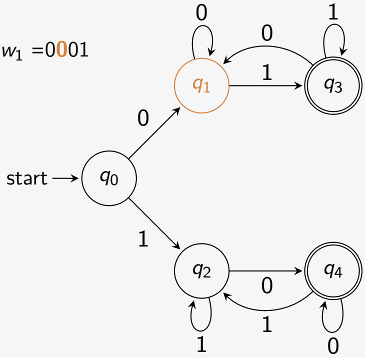
State Diagram and Finite Automaton



- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

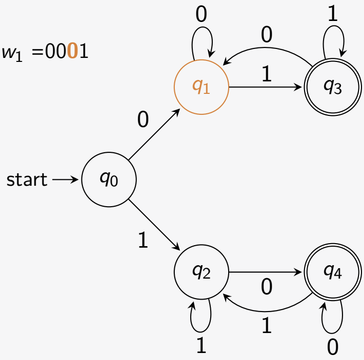
State Diagram and Finite Automaton

w_1 =0001



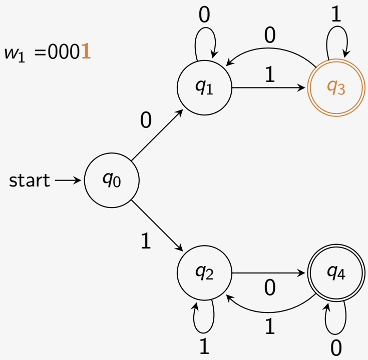
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

State Diagram and Finite Automaton



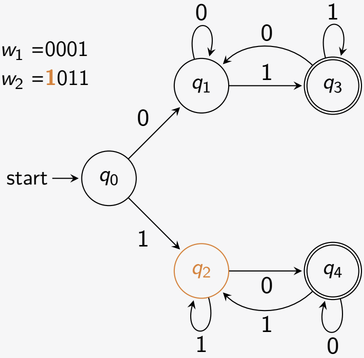
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

State Diagram and Finite Automaton



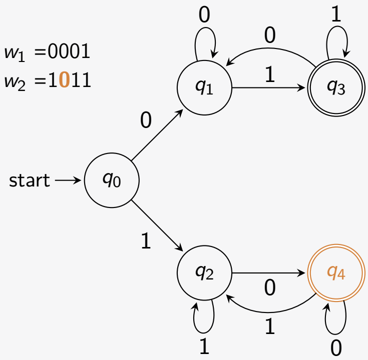
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0, 0 * 1\}$

State Diagram and Finite Automaton



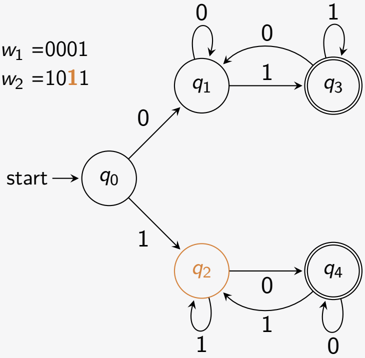
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

State Diagram and Finite Automaton



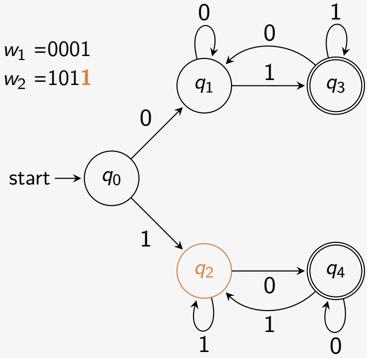
- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

State Diagram and Finite Automaton



- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

State Diagram and Finite Automaton



- States: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Alphabet: $\Sigma = \{0, 1\}$
- Start State: q_0
- Accept States: $F = \{q_3, q_4\}$
- Language: $L = \{1 * 0,\ 0 * 1\}$

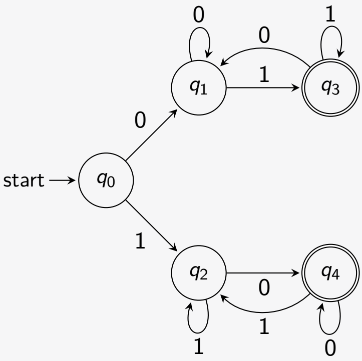
Formal Definition

Definition (Finite Automaton)

A ***finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set of ***states***,
2. Σ is a finite ***alphabet***,
3. $\delta : Q \times \Sigma \to Q$ is the ***transition function***,
4. $q_o \in Q$ is the ***start or initial state***, and
5. $F \subseteq Q$ is the set of ***accept or final states***.

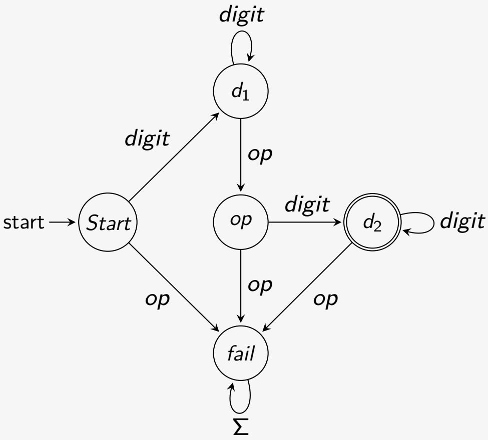
Transition Function



$$\delta : Q \times \Sigma \to Q$$

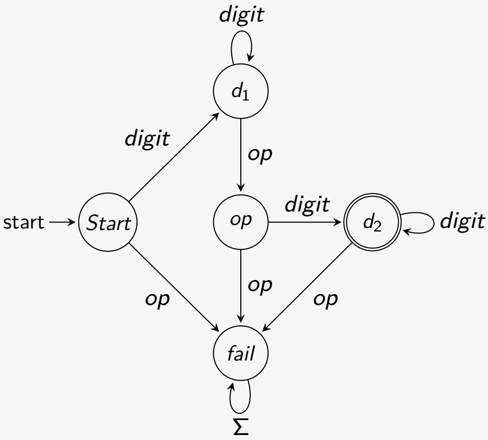
δ	0	1
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
q_3	q_1	q_3
q_4	q_4	q_2

Another Example



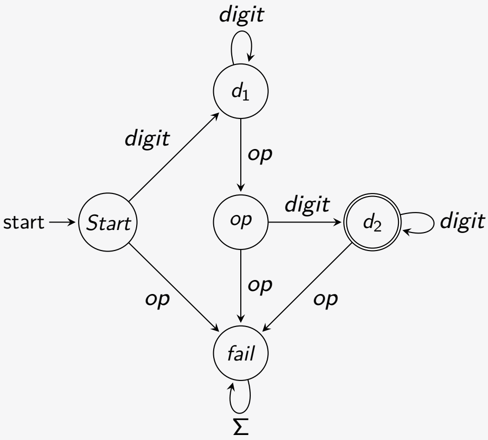
- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- L=?

Another Example



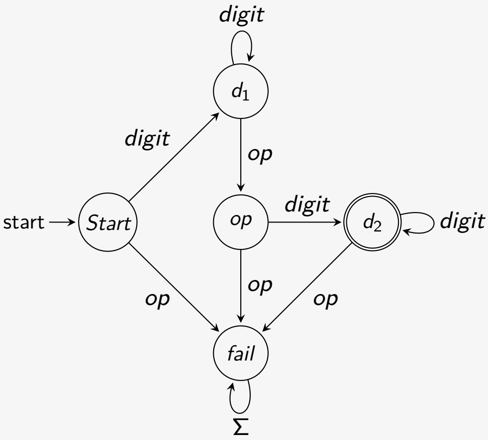
- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- $L = ?$
- $w = 25 + 362$

Another Example



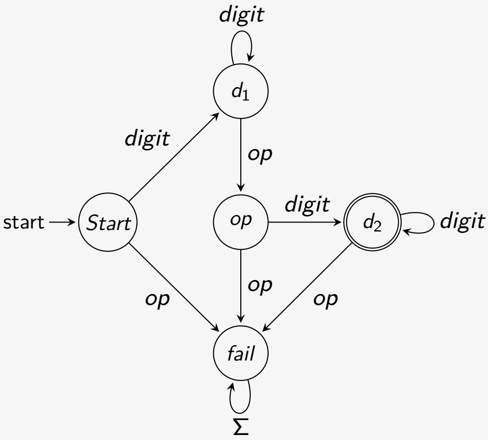
- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- $L = ?$
- $w = 25 + 362$
- $w = \times 35 - 67$

Another Example



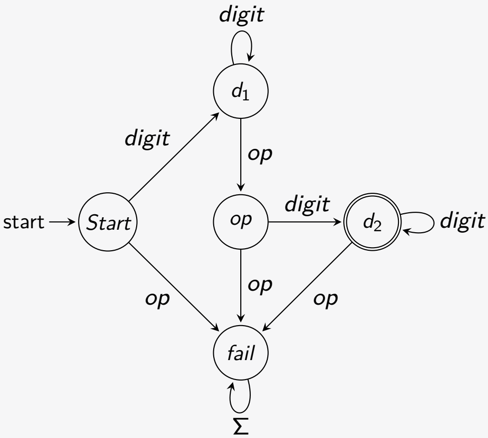
- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- $L = ?$
- $w = 25 + 362$
- $w = \times 35 - 67$
- $w = 265 - 456 \times 18$

Another Example



- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- L=*Combine two integers*
- $w = 25 + 362$
- $w = \times 35 - 67$
- $w = 265 - 456 \times 18$

Another Example



- $Q = \{Start, d_1, d_2, op, fail\}$
- $\Sigma = op \cup digit$
  - $op = \{+, -, \times\}$
  - $digit = \{0, 1, 2, \ldots, 9\}$
- $F = \{d_2\}$
- L=*Combine two integers*
- $w = 25 + 362$
- $w = \times 35 - 67$
- $w = 265 - 456 \times 18$

The language L is called a ***regular language*** because is is recognized by a finite automaton.

Satisfying a Description

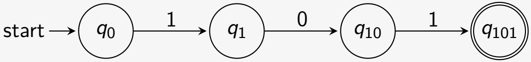
Problem

Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.

Satisfying a Description

Problem

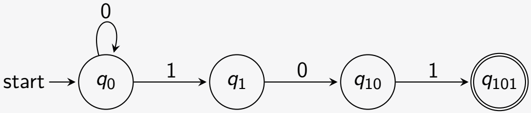
Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Satisfying a Description

Problem

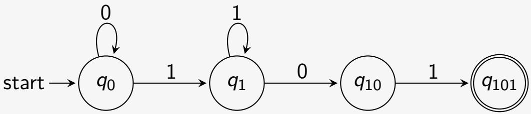
Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Satisfying a Description

Problem

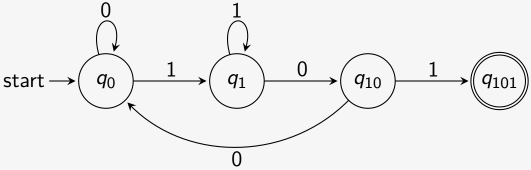
Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Satisfying a Description

Problem

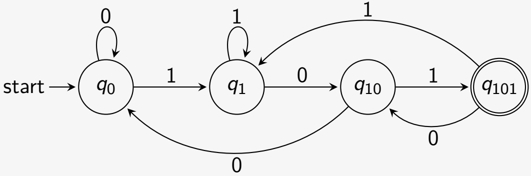
Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Satisfying a Description

Problem

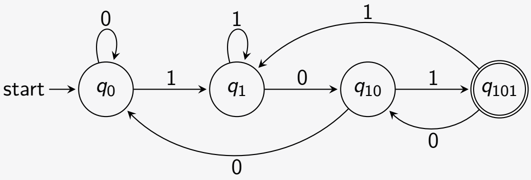
Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Satisfying a Description

Problem

Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.

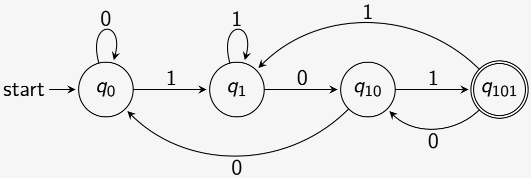


Check what happens to 101 independent of where we start.

Satisfying a Description

Problem

Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.

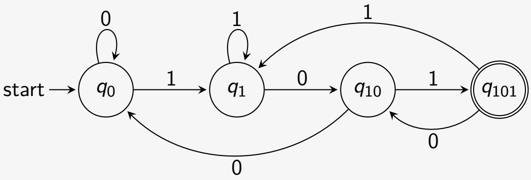


Check what happens to 010 independent of where we start.

Satisfying a Description

Problem

Construct a finite automaton that accepts only words in $\Sigma = \{0, 1\}$ ending in 101.



Note that this means we don't need to "remember" the whole string to check its ending.

Table of Contents

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \mathit{\ or\ } x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \mathit{\ and\ } y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \mathit{\ and\ } \forall i\ x_i \in A\}$

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \, or \, x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \, and \, y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \, and \, \forall i \, x_i \in A\}$

- $A = \{0, 1\}$

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \, or \, x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \, and \, y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \, and \, \forall i \, x_i \in A\}$

- $A = \{0, 1\}$
- $B = \{a, b\}$

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \, or \, x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \, and \, y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \, and \, \forall i \, x_i \in A\}$

- $A = \{0, 1\}$
- $B = \{a, b\}$
- $A \cup B = \{0, 1, a, b\}$

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \textit{ or } x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \textit{ and } y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \textit{ and } \forall i \; x_i \in A\}$

- $A = \{0, 1\}$
- $B = \{a, b\}$
- $A \cup B = \{0, 1, a, b\}$
- $A \circ B = \{0a, 0b, 1a, 1b\}$

Creating New Automata

Definition

Let A and B be regular languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- ***Union***: $A \cup B = \{x | x \in A \, or \, x \in B\}$
- ***Concatenation***: $A \circ B = \{xy | x \in A \, and \, y \in B\}$
- ***Star***: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \, and \, \forall i \, x_i \in A\}$

- $A = \{0, 1\}$
- $B = \{a, b\}$
- $A \cup B = \{0, 1, a, b\}$
- $A \circ B = \{0a, 0b, 1a, 1b\}$
- $A^* = \{\varepsilon, 0, 1, 00, 01, , 10, 11, 000, \ldots\}$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \textit{ and } M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

their union is constructed as follows:

- $Q = Q_1 \times Q_2$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \; and \; M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

their union is constructed as follows:

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \text{ and } M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

their union is constructed as follows:

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $\forall (r_1, r_2) \in Q \; \forall a \in \Sigma : \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \textit{ and } M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

their union is constructed as follows:

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $\forall (r_1, r_2) \in Q \, \forall a \in \Sigma : \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
- $q_0 = (q_1, q_2)$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \text{ and } M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

their union is constructed as follows:

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $\forall (r_1, r_2) \in Q \, \forall a \in \Sigma : \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) | r_1 \in F_1 \text{ or } r_2 \in F_2\}$

Details on Union

Given to finite automata

$$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1) \textit{ and } M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$$

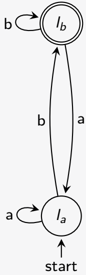
their union is constructed as follows:

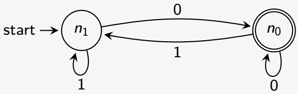
- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $\forall (r_1, r_2) \in Q \; \forall a \in \Sigma : \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) | r_1 \in F_1 \textit{ or } r_2 \in F_2\}$

(Note $r_1 \in F_1 \textit{ and } r_2 \in F_2$ would be intersection)

Union Example

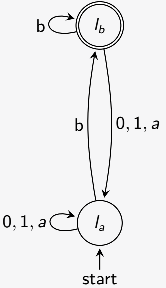
What words do each of these DFAs recognize? Before we begin we will need to adjust them to use the same alphabet.

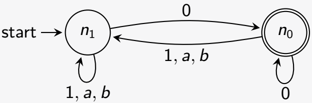




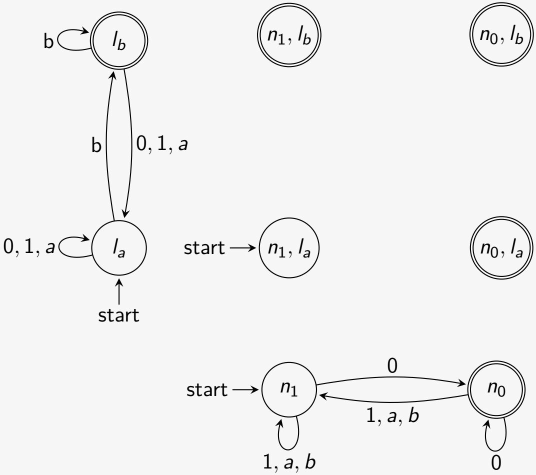
Union Example

To simplify the problem we change one to accept words ending in 0 and the other to be words ending in b. This changes the problem slightly but makes a cleaner example.

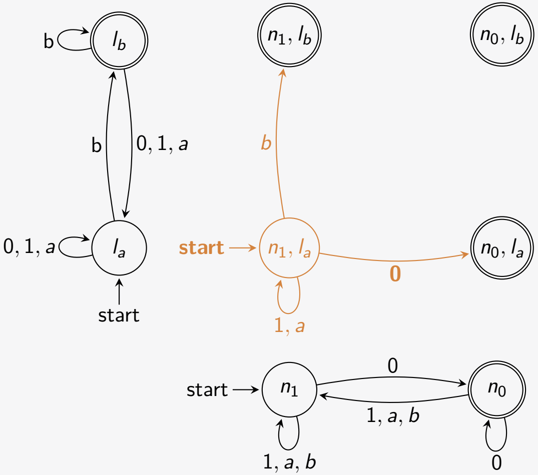




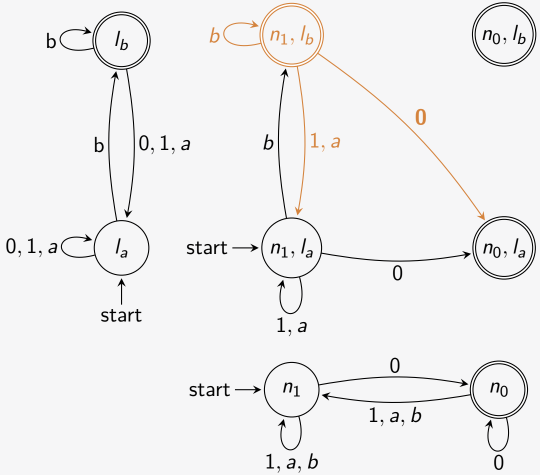
Union Example



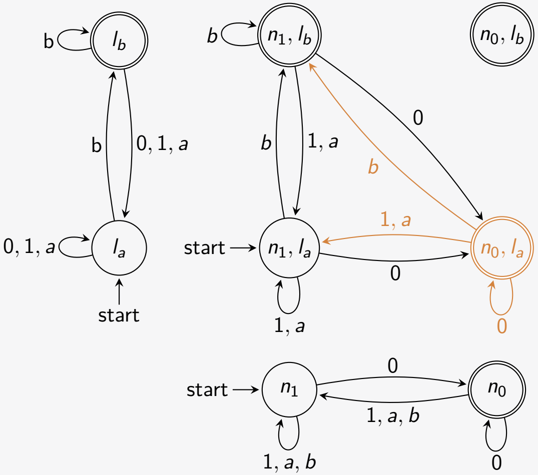
Union Example



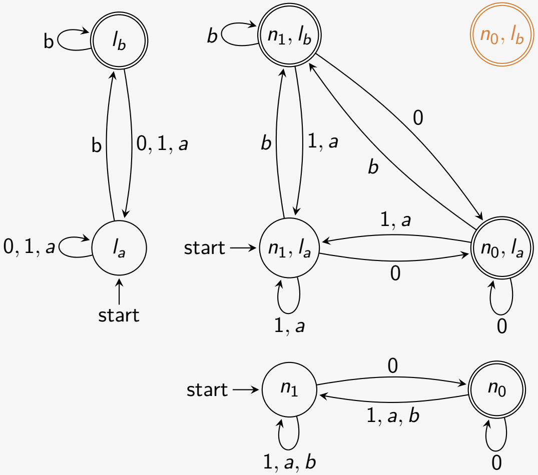
Union Example



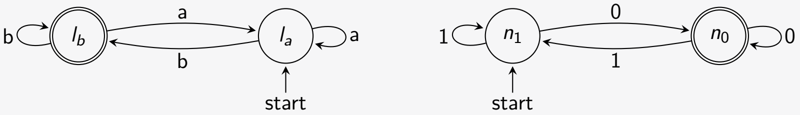
Union Example



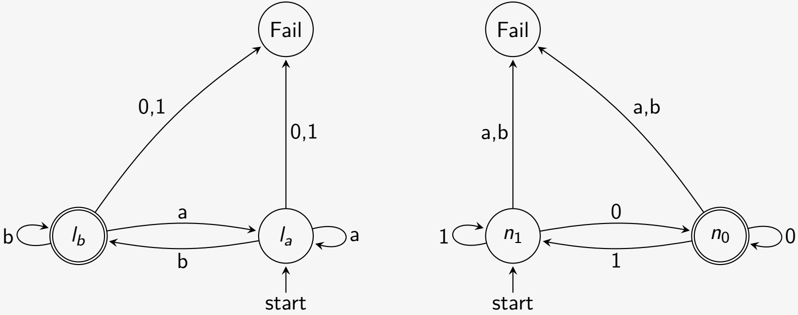
Union Example



Union Example Take Two



Union Example Take Two



Union Example Take Two

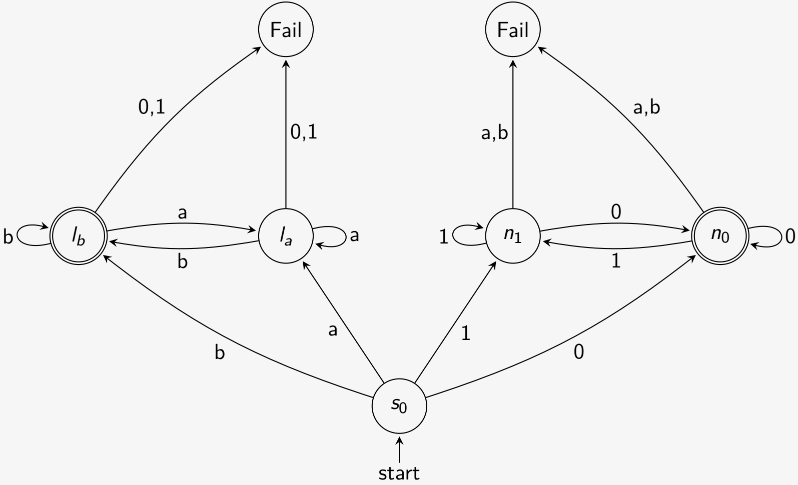
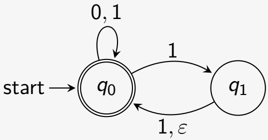


Table of Contents

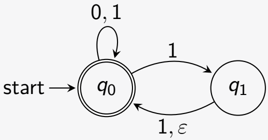
Non-Deterministic Automata



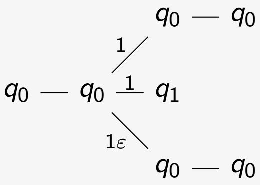
δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$

$$w = 010$$

Non-Deterministic Automata

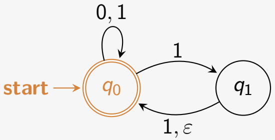


δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$

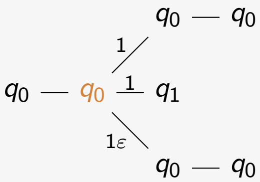


$w = 010$

Non-Deterministic Automata

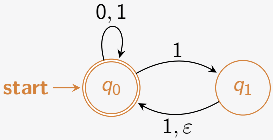


δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$

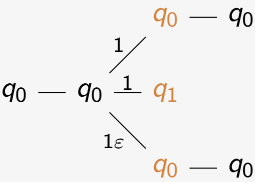


$w = \mathbf{0}10$

Non-Deterministic Automata

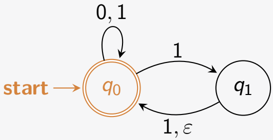


δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$

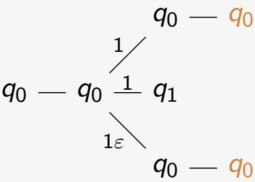


$w = 010$

Non-Deterministic Automata

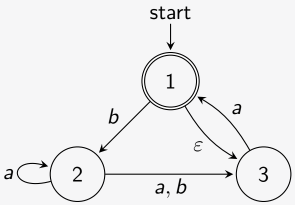


δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$



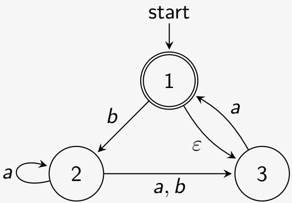
$w = 01\mathbf{0}$

Equivalence to Deterministic (Slide 1)



δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

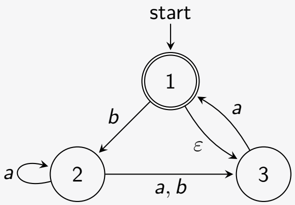
Equivalence to Deterministic (Slide 1)



δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

Equivalence to Deterministic (Slide 1)

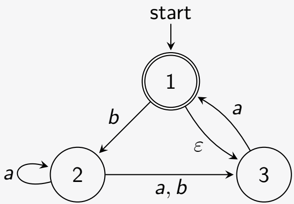


δ	a	b	ε
1	$\varnothing$	{2}	{3}
2	{2,3}	{3}	$\varnothing$
3	{1}	$\varnothing$	$\varnothing$

- $Q' = \mathscr{P}(Q)$ **(Power set of Q)**

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

Equivalence to Deterministic (Slide 1)

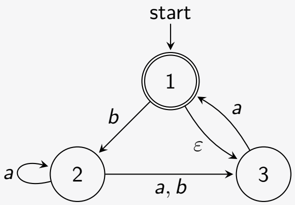


δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

- $Q' = \mathscr{P}(Q)$ (Power set of Q)
- $\Sigma' = \{a, b\}$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

Equivalence to Deterministic (Slide 1)

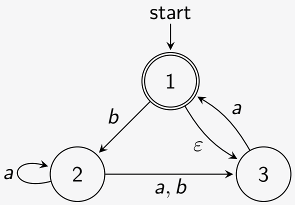


δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

- $Q' = \mathscr{P}(Q)$ (Power set of Q)
- $\Sigma' = \{a, b\}$
- $E(\{1\}) = \{1, 3\}$ (**states reached from 1 using ε**)

Equivalence to Deterministic (Slide 1)

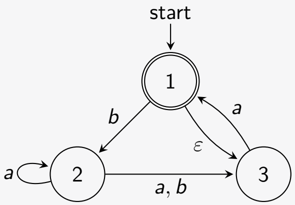


δ	a	b	ε
1	$\varnothing$	{2}	{3}
2	{2,3}	{3}	$\varnothing$
3	{1}	$\varnothing$	$\varnothing$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

- $Q' = \mathscr{P}(Q)$ (Power set of Q)
- $\Sigma' = \{a, b\}$
- $E(\{1\}) = \{1, 3\}$ (states reached from 1 using ε)
- $Start = \{1, 3\}$

Equivalence to Deterministic (Slide 1)

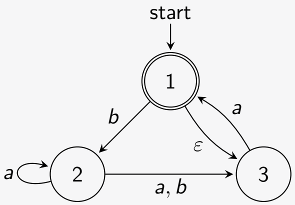


δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

- $Q' = \mathscr{P}(Q)$ (Power set of Q)
- $\Sigma' = \{a, b\}$
- $E(\{1\}) = \{1, 3\}$ (states reached from 1 using ε)
- $Start = \{1, 3\}$
- $F' = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}$

Equivalence to Deterministic (Slide 1)



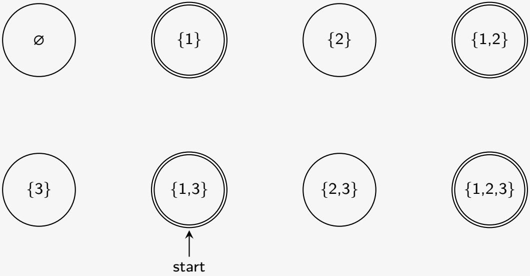
δ	a	b	ε
1	$\varnothing$	{2}	{3}
2	{2,3}	{3}	$\varnothing$
3	{1}	$\varnothing$	$\varnothing$

- $Q = \{1, 2, 3\}$
- $\Sigma_\varepsilon = \{a, b, \varepsilon\}$
- $Start = \{1\}$
- $F = \{1\}$

- $Q' = \mathscr{P}(Q)$ (Power set of Q)
- $\Sigma' = \{a, b\}$
- $E(\{1\}) = \{1, 3\}$ (states reached from 1 using ε)
- $Start = \{1, 3\}$
- $F' = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}$
- $\delta' =?$

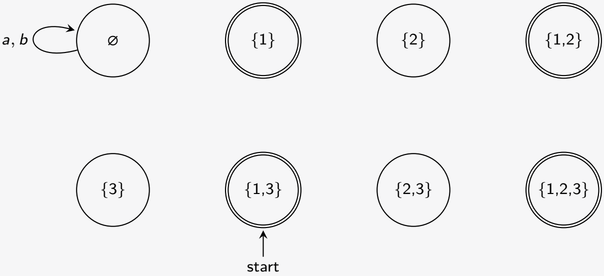
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\emptyset$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\emptyset$
3	$\{1\}$	$\emptyset$	$\emptyset$



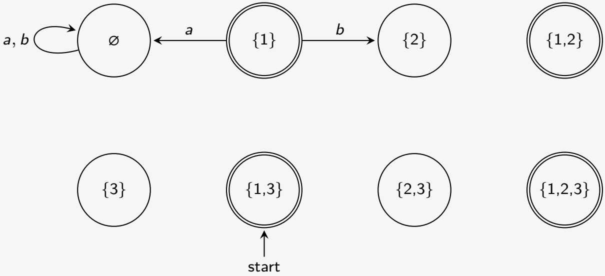
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



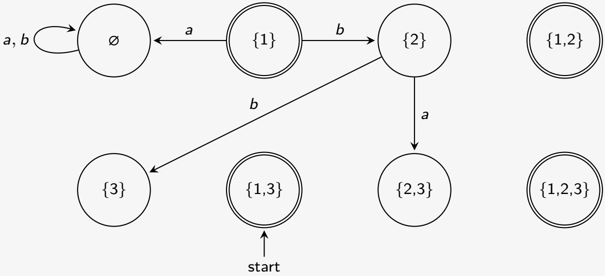
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



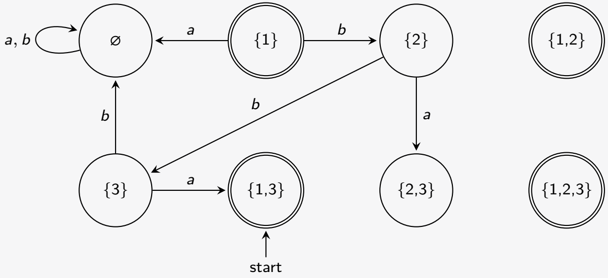
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



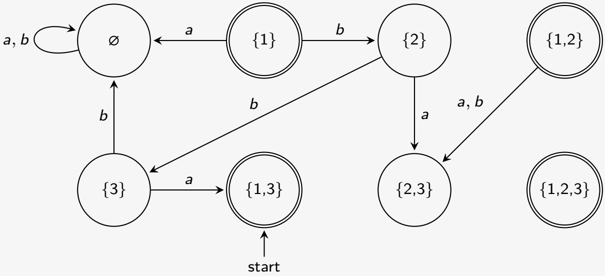
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



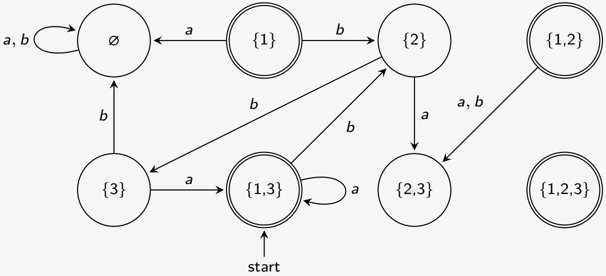
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



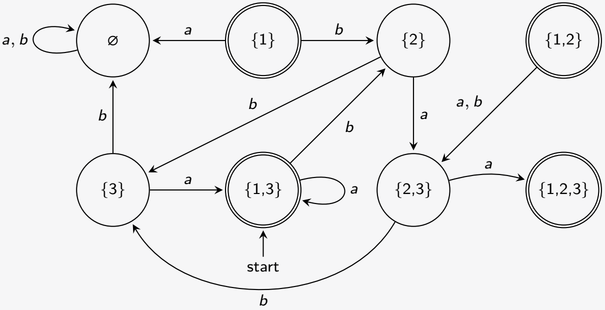
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



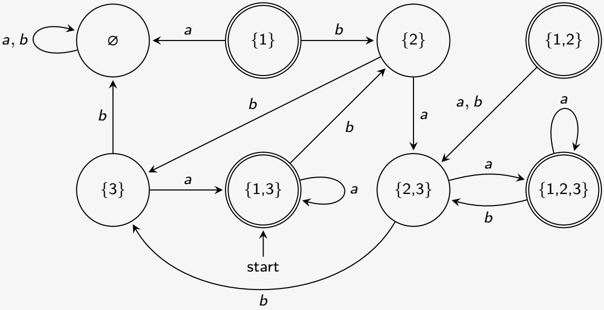
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



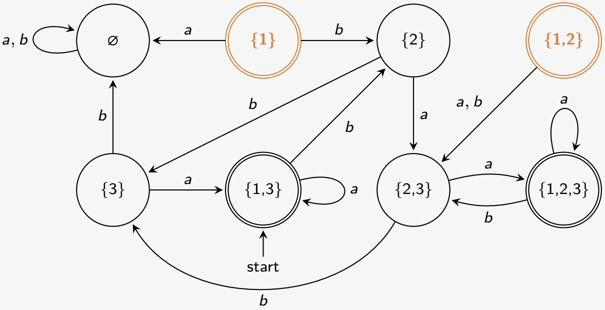
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



Equivalence to Deterministic (Slide 2)

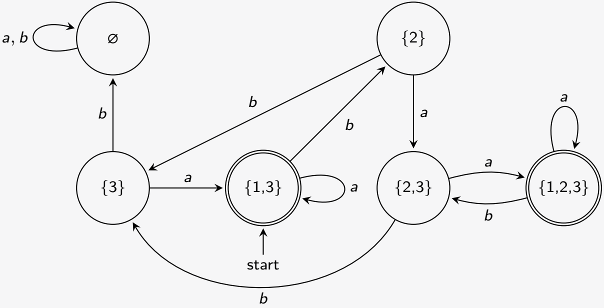
δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$



We can eliminate states that are only “sources.”

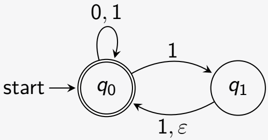
Equivalence to Deterministic (Slide 2)

δ	a	b	ε
1	$\varnothing$	$\{2\}$	$\{3\}$
2	$\{2,3\}$	$\{3\}$	$\varnothing$
3	$\{1\}$	$\varnothing$	$\varnothing$

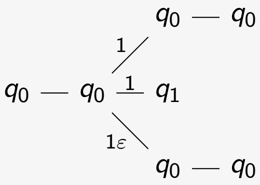


We can eliminate states that are only “sources.”

Non-Deterministic Automata



δ	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0\}$	$\{q_0\}$

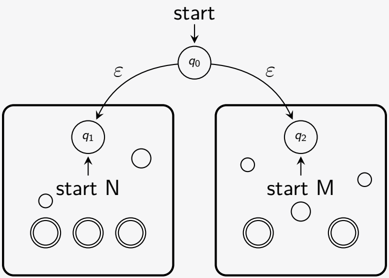


$w = 010$

Table of Contents

Union Revisited: $A \cup B$

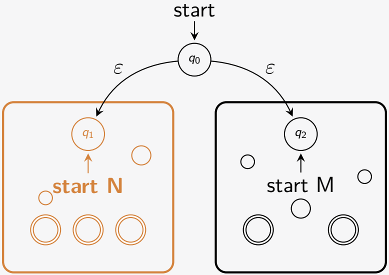
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **union** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \wedge a = \varepsilon \\ \varnothing & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Union Revisited: $A \cup B$

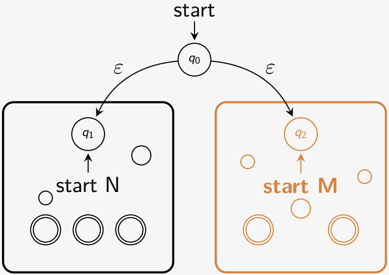
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **union** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \wedge a = \varepsilon \\ \varnothing & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Union Revisited: $A \cup B$

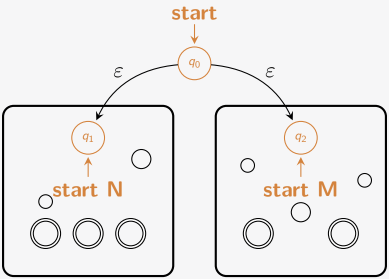
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **union** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \land a = \varepsilon \\ \varnothing & q = q_0 \land a \neq \varepsilon \end{cases}$$

Union Revisited: $A \cup B$

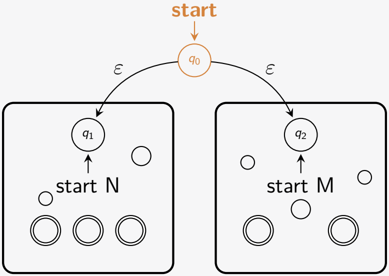
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **union** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \wedge a = \varepsilon \\ \emptyset & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Union Revisited: $A \cup B$

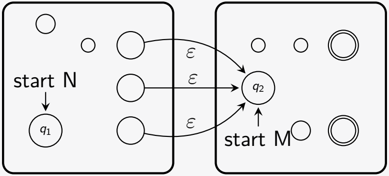
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **union** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \wedge a = \varepsilon \\ \varnothing & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Concatenation: $A \circ B$

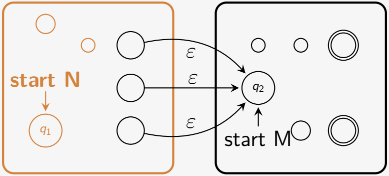
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **concatenation** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \wedge q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \wedge a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \wedge a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

Concatenation: $A \circ B$

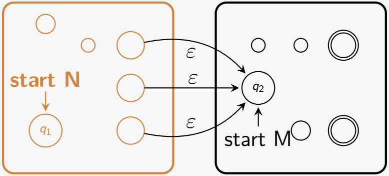
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **concatenation** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \land a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

Concatenation: $A \circ B$

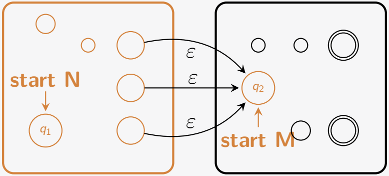
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **concatenation** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \wedge q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \wedge a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \wedge a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

Concatenation: $A \circ B$

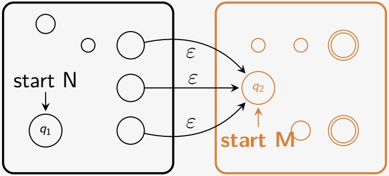
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **concatenation** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \land a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

Concatenation: $A \circ B$

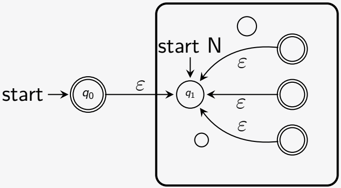
Assume that the regular language A is represented by the finite automaton N and the regular language B is represented by the finite automaton M, then their **concatenation** can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \wedge q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \wedge a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \wedge a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

Star: A^*

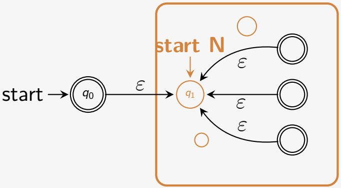
Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \land a = \varepsilon \\ \{q_1\} & q = q_0 \land a = \varepsilon \\ \varnothing & q = q_0 \land a \neq \varepsilon \end{cases}$$

Star: A^*

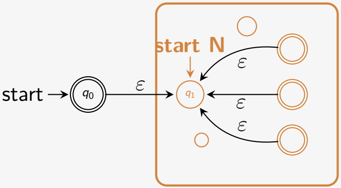
Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \land a = \varepsilon \\ \{q_1\} & q = q_0 \land a = \varepsilon \\ \emptyset & q = q_0 \land a \neq \varepsilon \end{cases}$$

Star: A^*

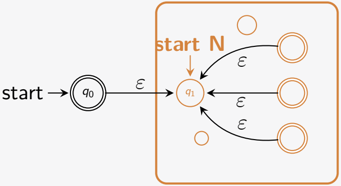
Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \wedge a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \wedge a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \wedge a = \varepsilon \\ \{q_1\} & q = q_0 \wedge a = \varepsilon \\ \emptyset & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Star: A^*

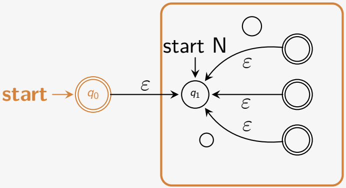
Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \land a = \varepsilon \\ \{q_1\} & q = q_0 \land a = \varepsilon \\ \varnothing & q = q_0 \land a \neq \varepsilon \end{cases}$$

Star: A^*

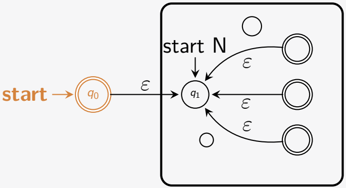
Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \land a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \land a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \land a = \varepsilon \\ \{q_1\} & q = q_0 \land a = \varepsilon \\ \varnothing & q = q_0 \land a \neq \varepsilon \end{cases}$$

Star: A^*

Assume that the regular language A is represented by the finite automaton N then the unary **star, $*$,** operator applied to A can be represented as below.



$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \wedge a \notin F_1 \\ \delta_1(q,a) & q \in F_1 \wedge a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \wedge a = \varepsilon \\ \{q_1\} & q = q_0 \wedge a = \varepsilon \\ \emptyset & q = q_0 \wedge a \neq \varepsilon \end{cases}$$

Table of Contents

Next Class

- **Regular Expressions**

Next Class

- Regular Expressions
- **Generalized Nondeterministic Finite Automata**

Next Class

- Regular Expressions
- Generalized Nondeterministic Finite Automata
- **Pumping Lemma**

Next Class

- Regular Expressions
- Generalized Nondeterministic Finite Automata
- Pumping Lemma
- **Nonregular Languages**

Introduction to Finite Automata

Dr. Chuck Rocca

roccac@wcsu.edu

http://sites.wcsu.edu/roccac

WESTERN CONNECTICUT STATE UNIVERSITY
MACRICOSTAS SCHOOL OF ARTS & SCIENCES