Lecture 7: DFA Minimization and Non-regular Languages

CSC 320: Foundations of Computer Science

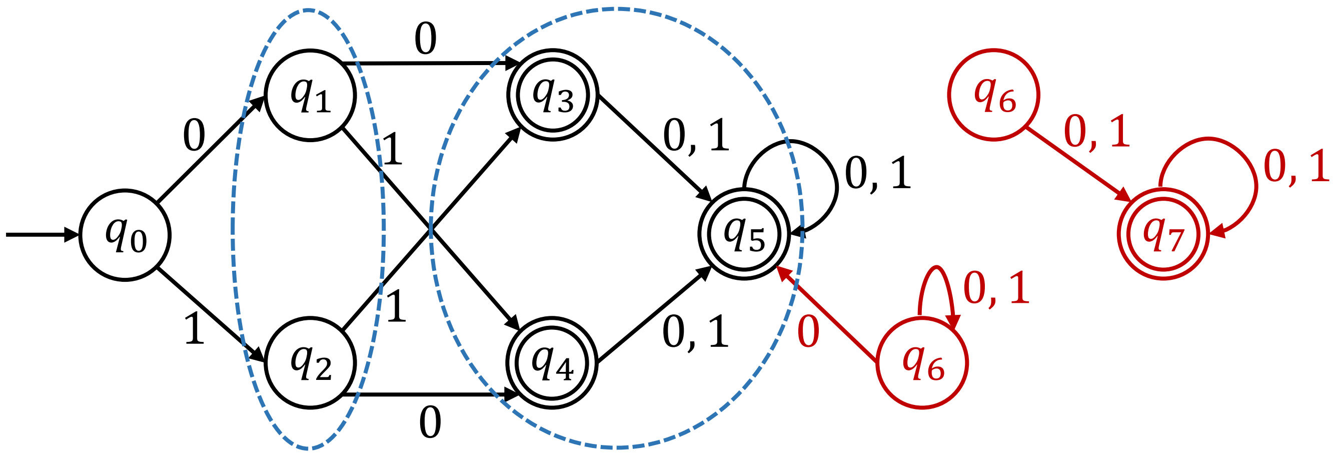
Quinton Yong

quintonyong@uvic.ca

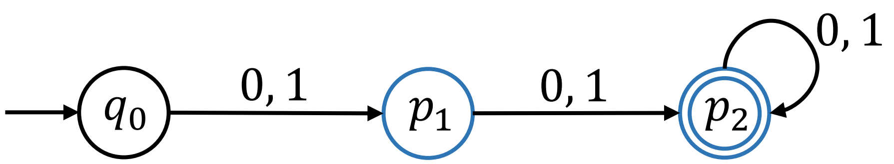
University of Victoria

DFA State Minimization

- Given a DFA D, **reduce the number of states** without changing the language recognized by D



1. **Remove unreachable states**
2. Identify and **collapse equivalent states** (while still maintaining DFA rules)



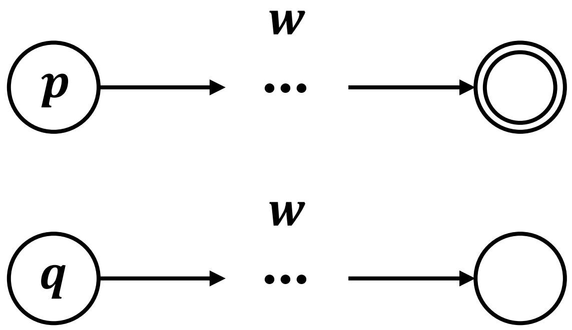
DFA State Minimization

Which states **cannot** be collapsed?

- Clearly, an **accept** and a **non-accept** state are not equivalent / collapsible

In general, we **cannot collapse** states p and q if there is any string w such that:

- when w is processed starting at p, we yield **acceptance**
- when w is processed starting at q, we yield **non-acceptance**

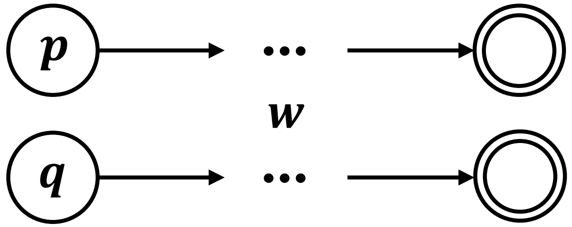


DFA State Equivalence

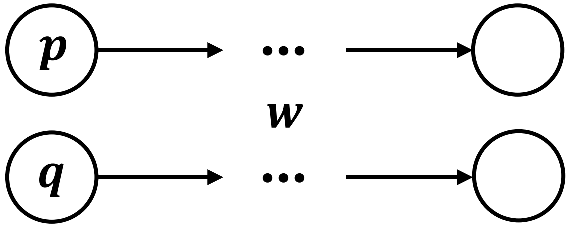
Two states $\boldsymbol{p}$ and $\boldsymbol{q}$ of a **DFA** are **equivalent** (denoted $\boldsymbol{p} \sim \boldsymbol{q}$) **if and only if for all** $\boldsymbol{w} \in \boldsymbol{\Sigma}^*$:

- computation of $\boldsymbol{w}$ starting at state $\boldsymbol{p}$ yields **acceptance** **if and only if**
- computation of $\boldsymbol{w}$ starting at $\boldsymbol{q}$ yields **acceptance**

For all $\boldsymbol{w} \in \boldsymbol{\Sigma}^*$



or



In other words, $\boldsymbol{p}$ and $\boldsymbol{q}$ have **exactly the same role** in the DFA in terms of accepting and not accepting strings

DFA State Minimization Algorithm

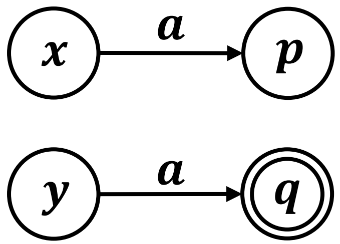
We work backwards, and **mark** which pairs of states **cannot be equivalent**

- **Accept** and **non-accept states** are not equivalent

$\cancel{\{p, q\}}$

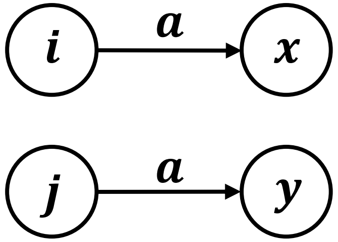
- If a pair of states $\boldsymbol{x}$ and $\boldsymbol{y}$ have a transition for the same symbol $\boldsymbol{a}$ **to non-equivalent states**, they also cannot be equivalent

$\cancel{\{x, y\}}$



- Continue **until no more changes occur**

$\cancel{\{i, j\}}$



DFA State Minimization Algorithm

Let $\boldsymbol{M}$ be a DFA with **no inaccessible states** (first remove unreachable states)

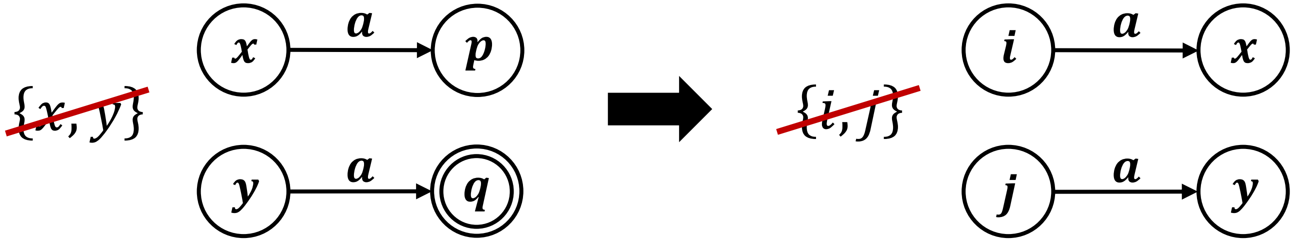
1. Write down all unordered pairs $\{\boldsymbol{p}, \boldsymbol{q}\}$ of states in $\boldsymbol{M}$
2. Mark each pair $\{\boldsymbol{p}, \boldsymbol{q}\}$ where $\boldsymbol{p} \in \boldsymbol{F}$ and $\boldsymbol{q} \notin \boldsymbol{F}$ (or vice versa)

$\cancel{\{p, q\}}$

3. Repeat the following **until no changes occur**:

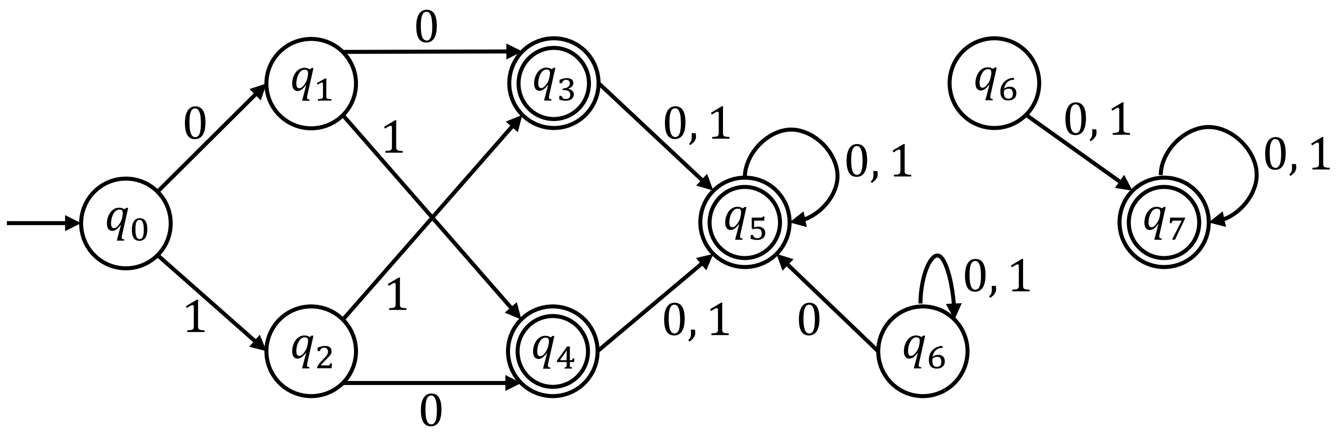
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$



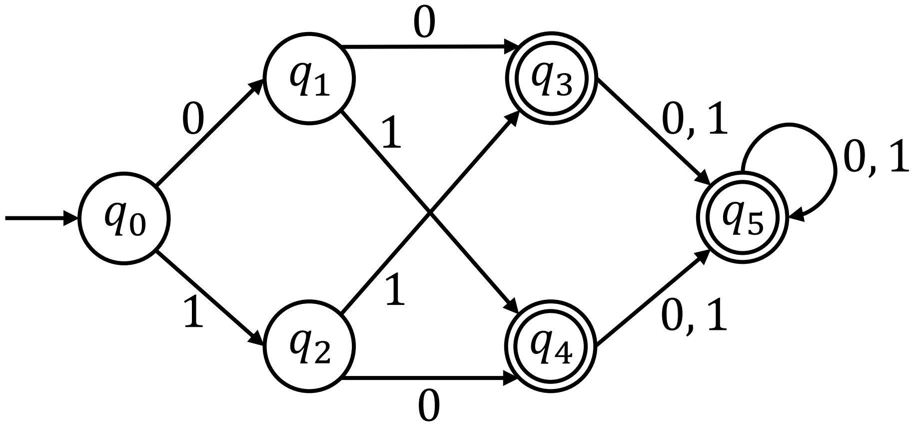
This algorithm produces state equivalence: $\{\boldsymbol{p}, \boldsymbol{q}\}$ unmarked if and only if $\boldsymbol{p} \sim \boldsymbol{q}$

DFA State Minimization Algorithm Example



Remove **unreachable** states

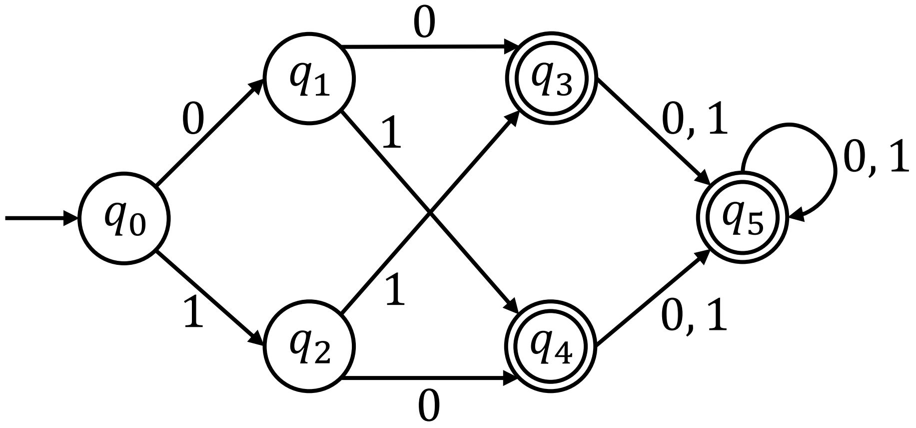
DFA State Minimization Algorithm Example



Write down all unordered pairs $\{\boldsymbol{p}, \boldsymbol{q}\}$ of states in $\boldsymbol{M}$

$\{q_0, q_1\}$	$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\{q_0, q_2\}$	$\{q_1, q_3\}$	$\{q_2, q_4\}$	$\{q_3, q_5\}$	
$\{q_0, q_3\}$	$\{q_1, q_4\}$	$\{q_2, q_5\}$		
$\{q_0, q_4\}$	$\{q_1, q_5\}$			
$\{q_0, q_5\}$				

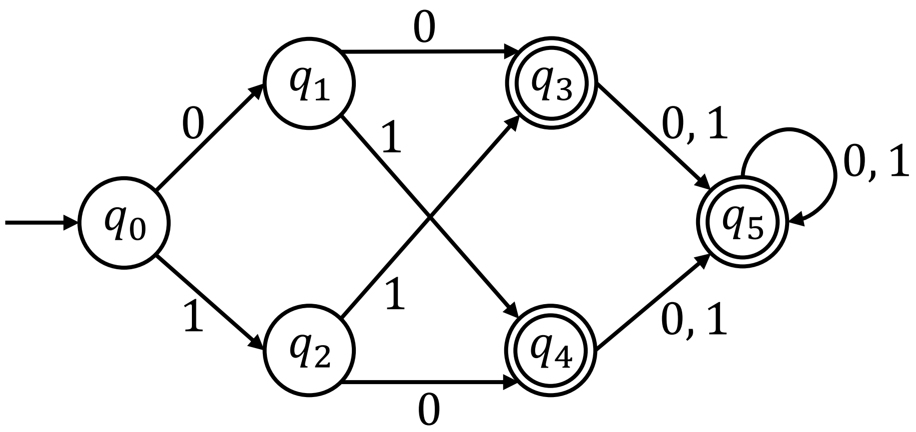
DFA State Minimization Algorithm Example



Mark each pair $\{\boldsymbol{p}, \boldsymbol{q}\}$ where $\boldsymbol{p} \in \boldsymbol{F}$ and $\boldsymbol{q} \notin \boldsymbol{F}$ (or vice versa)

$\{q_0, q_1\}$	$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\{q_0, q_2\}$	$\{q_1, q_3\}$	$\{q_2, q_4\}$	$\{q_3, q_5\}$	
$\{q_0, q_3\}$	$\{q_1, q_4\}$	$\{q_2, q_5\}$		
$\{q_0, q_4\}$	$\{q_1, q_5\}$			
$\{q_0, q_5\}$				

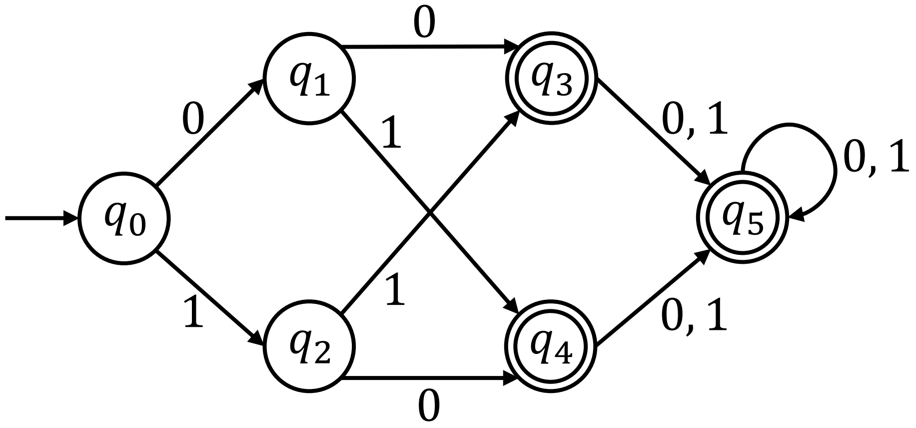
DFA State Minimization Algorithm Example



Mark each pair $\{\boldsymbol{p}, \boldsymbol{q}\}$ where $\boldsymbol{p} \in \boldsymbol{F}$ and $\boldsymbol{q} \notin \boldsymbol{F}$ (or vice versa)

$\{q_0, q_1\}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\{q_0, q_2\}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



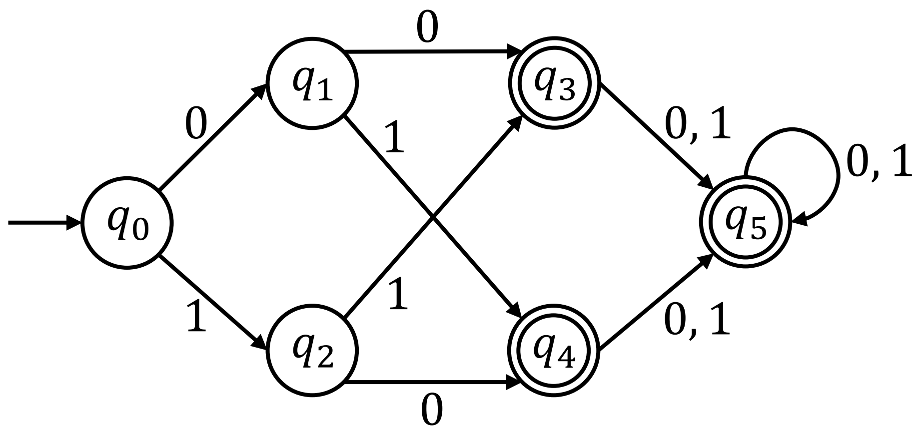
Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\{q_0, q_1\}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\{q_0, q_2\}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



$\{\delta(q_0, 0), \delta(q_1, 0)\} = \{q_1, q_3\}$

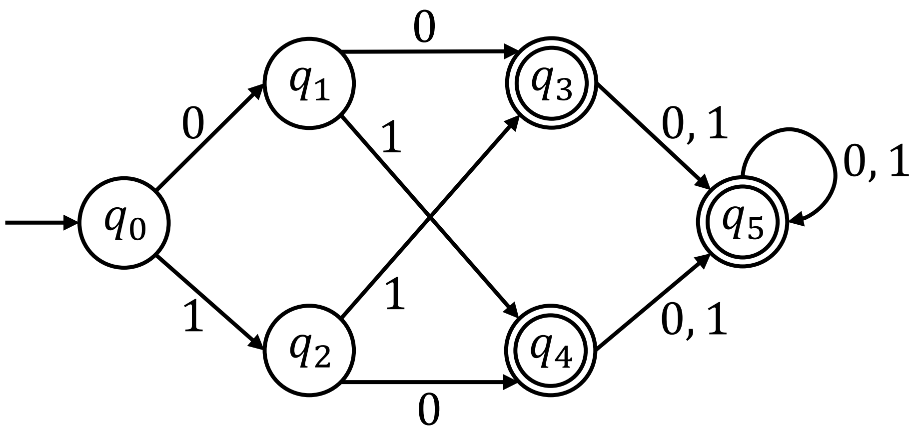
Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\boxed{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\{q_0, q_2\}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



$\{\delta(q_0, 0), \delta(q_2, 0)\} = \{q_1, q_4\}$

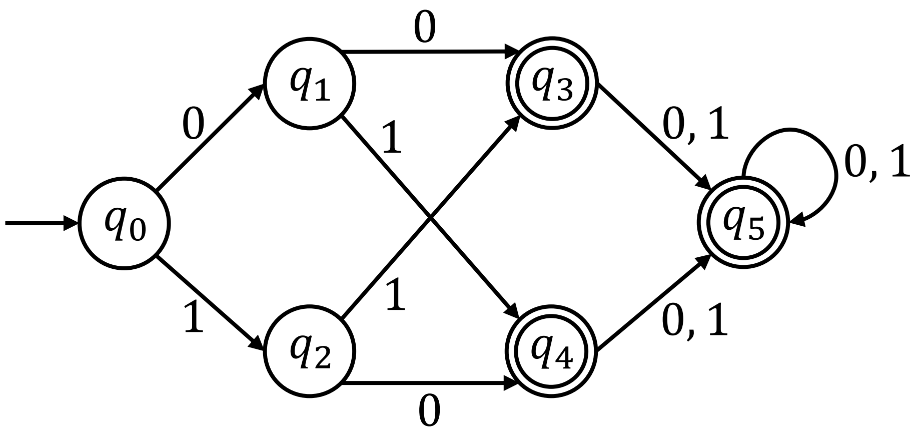
Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\boxed{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



$\{\delta(q_1, 0), \delta(q_2, 0)\} = \{q_3, q_4\}$

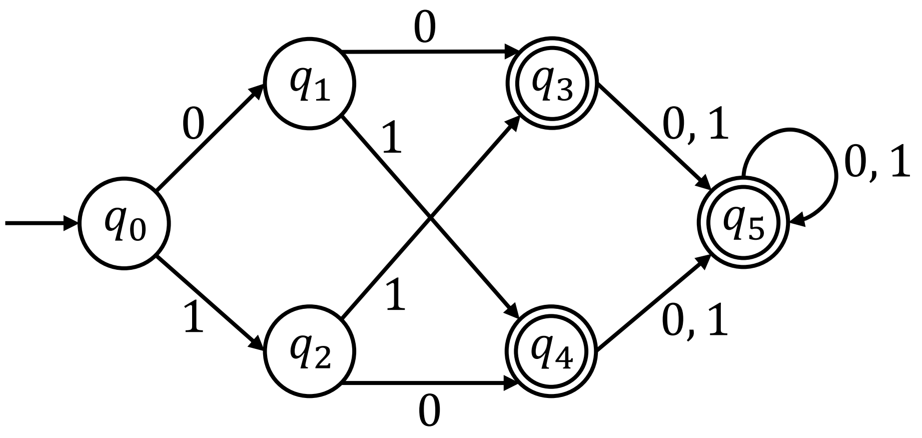
Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\cancel{\{q_0, q_1\}}$	$\boxed{\{q_1, q_2\}}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



$\{\delta(q_1, 1), \delta(q_2, 1)\} = \{q_4, q_3\}$

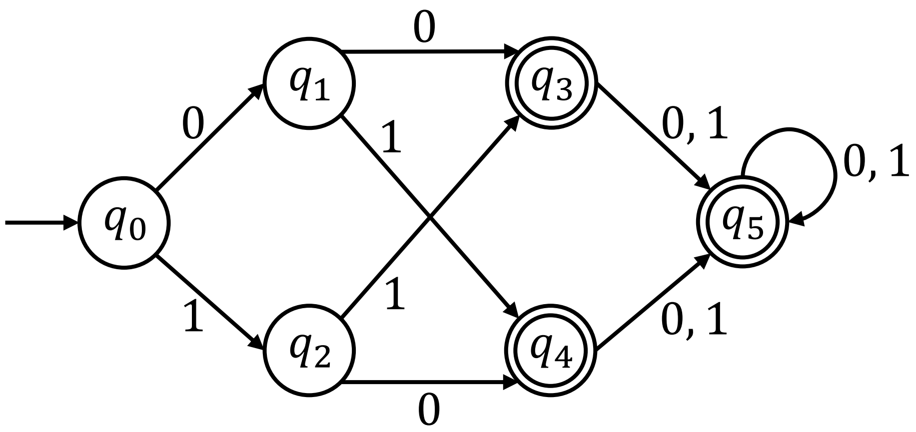
$\cancel{\{q_0, q_1\}}$	$\boxed{\{q_1, q_2\}}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

DFA State Minimization Algorithm Example



$\{\delta(q_3, 0), \delta(q_4, 0)\} = \{q_5, q_5\}$

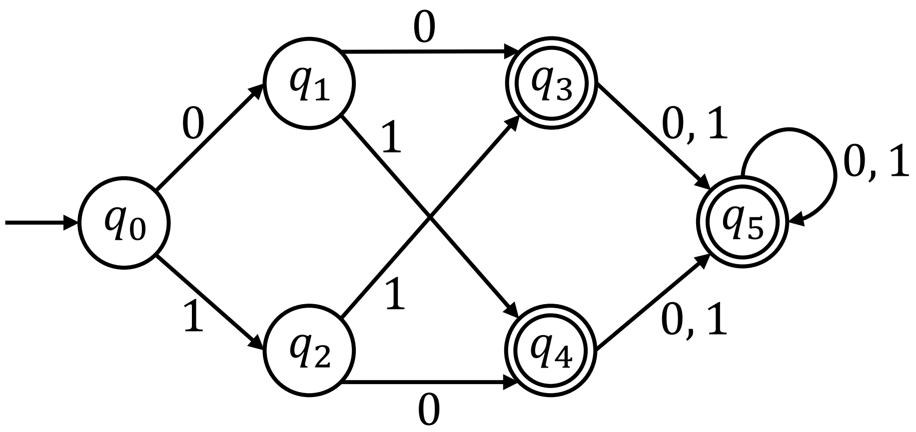
Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\boxed{\{q_3, q_4\}}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



Repeat the **until no changes occur**:

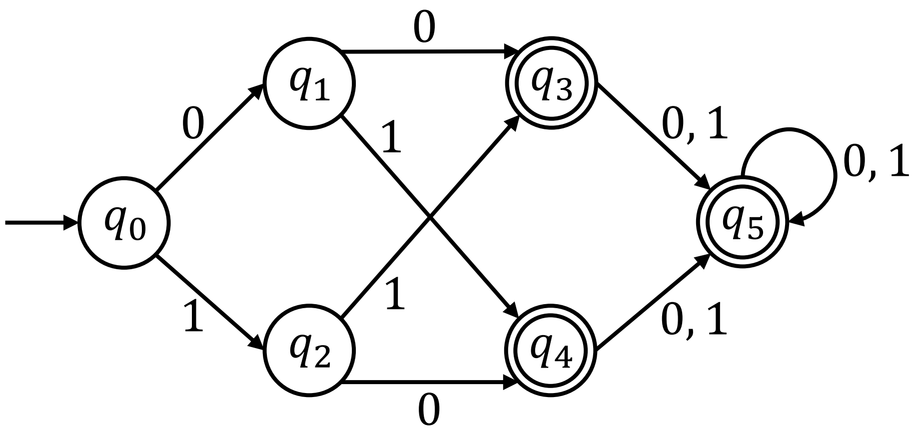
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\{\delta(q_3, 1), \delta(q_4, 1)\} = \{q_5, q_5\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\boxed{\{q_3, q_4\}}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



Repeat the **until no changes occur**:

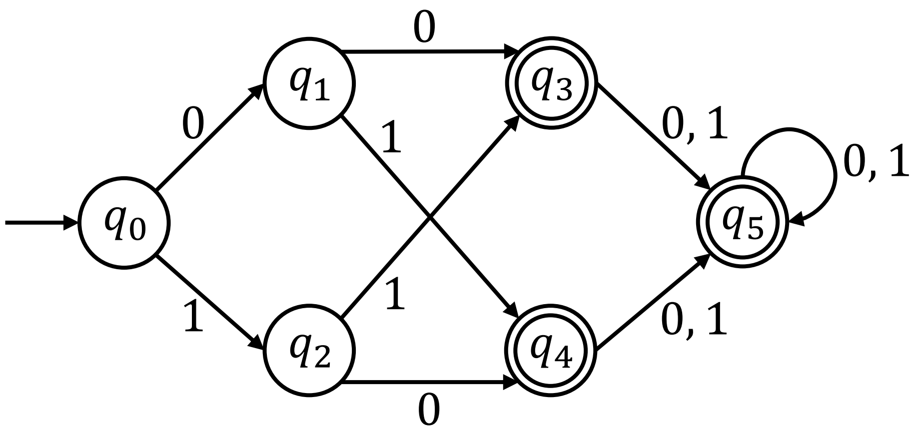
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$$\{\delta(q_3, 0), \delta(q_5, 0)\} = \{q_5, q_5\}$$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\boxed{\{q_3, q_5\}}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



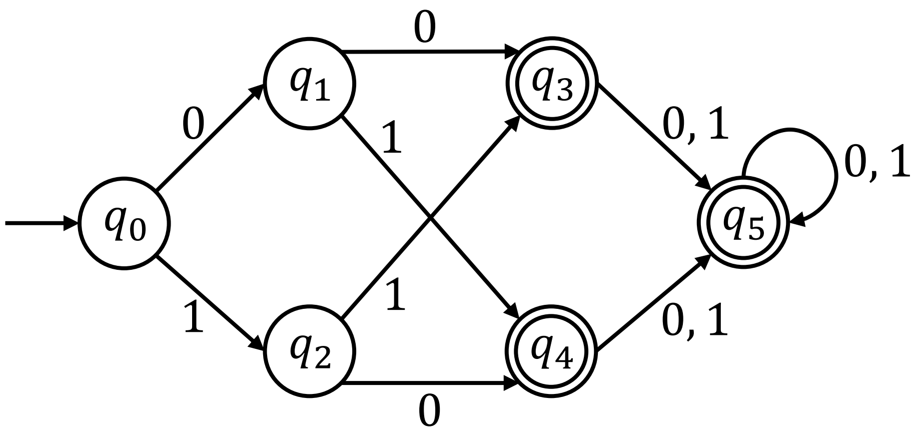
Repeat the **until no changes occur**:
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\{\delta(q_3, 1), \delta(q_5, 1)\} = \{q_5, q_5\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\boxed{\{q_3, q_5\}}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



Repeat the **until no changes occur**:

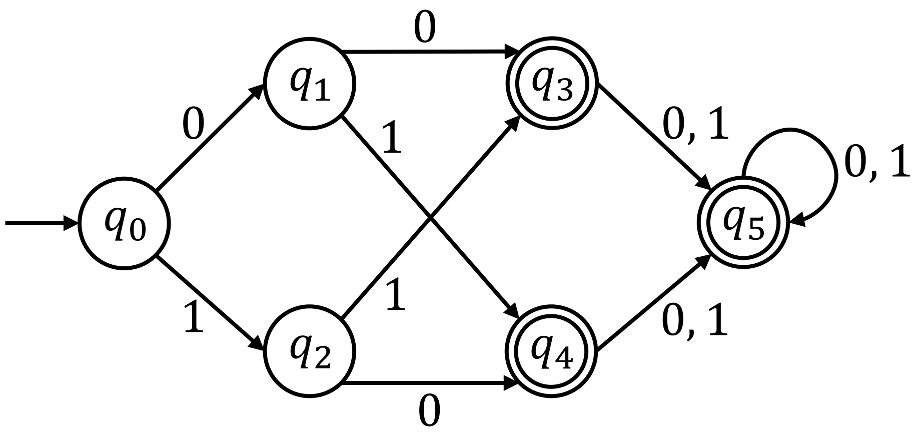
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\{\delta(q_4, 0), \delta(q_5, 0)\} = \{q_5, q_5\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\boxed{\{q_4, q_5\}}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



Repeat the **until no changes occur**:

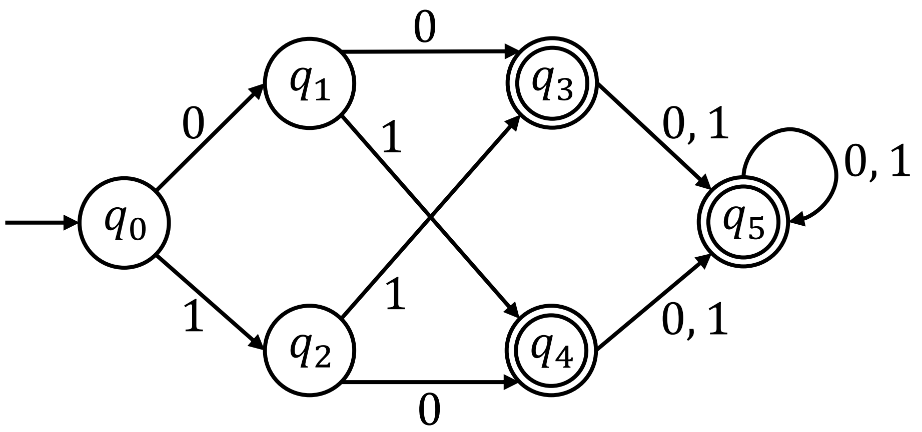
If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

$\{\delta(q_4, 1), \delta(q_5, 1)\} = \{q_5, q_5\}$

$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\boxed{\{q_4, q_5\}}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example



Repeat process again, but no changes occur

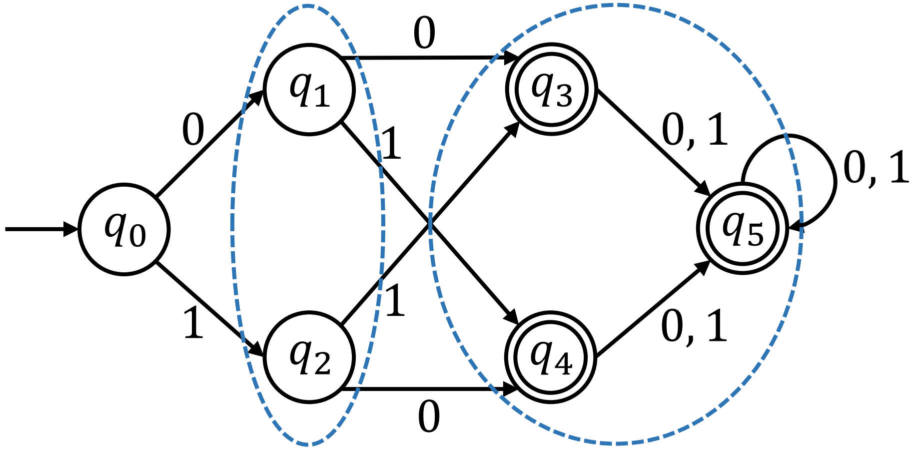
~~$\{q_0, q_1\}$~~	$\{q_1, q_2\}$	~~$\{q_2, q_3\}$~~	$\{q_3, q_4\}$	$\{q_4, q_5\}$
~~$\{q_0, q_2\}$~~	~~$\{q_1, q_3\}$~~	~~$\{q_2, q_4\}$~~	$\{q_3, q_5\}$	
~~$\{q_0, q_3\}$~~	~~$\{q_1, q_4\}$~~	~~$\{q_2, q_5\}$~~		
~~$\{q_0, q_4\}$~~	~~$\{q_1, q_5\}$~~			
~~$\{q_0, q_5\}$~~				

Repeat the **until no changes occur**:

If there exists an unmarked pair $\{\boldsymbol{i}, \boldsymbol{j}\}$ such that:

- $\{\boldsymbol{\delta}(\boldsymbol{i}, \boldsymbol{a}), \boldsymbol{\delta}(\boldsymbol{j}, \boldsymbol{a})\}$ is marked for some $\boldsymbol{a} \in \boldsymbol{\Sigma}$
- then mark $\{\boldsymbol{i}, \boldsymbol{j}\}$

DFA State Minimization Algorithm Example



Equivalent:

p_1 : $q_1 \sim q_2$

p_2 : $q_3 \sim q_4 \sim q_5$

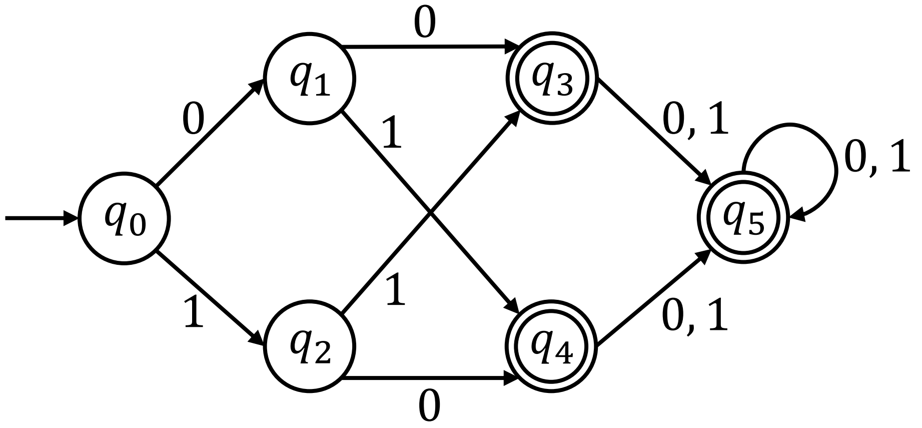
$\cancel{\{q_0, q_1\}}$	$\{q_1, q_2\}$	$\cancel{\{q_2, q_3\}}$	$\{q_3, q_4\}$	$\{q_4, q_5\}$
$\cancel{\{q_0, q_2\}}$	$\cancel{\{q_1, q_3\}}$	$\cancel{\{q_2, q_4\}}$	$\{q_3, q_5\}$	
$\cancel{\{q_0, q_3\}}$	$\cancel{\{q_1, q_4\}}$	$\cancel{\{q_2, q_5\}}$		
$\cancel{\{q_0, q_4\}}$	$\cancel{\{q_1, q_5\}}$			
$\cancel{\{q_0, q_5\}}$				

DFA State Minimization Algorithm Example

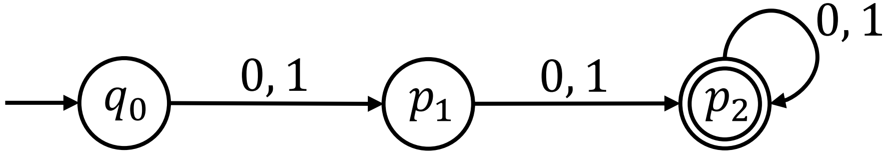
Equivalent:

$p_1 : q_1 \sim q_2$

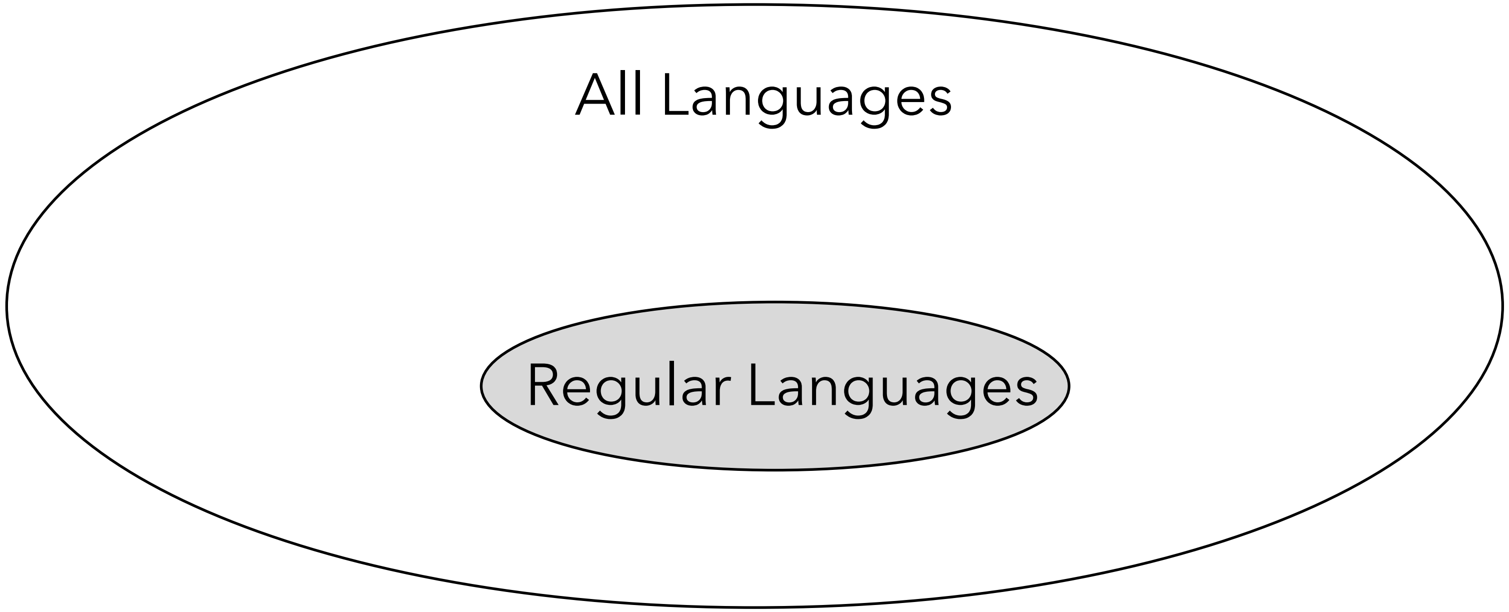
$p_2 : q_3 \sim q_4 \sim q_5$



DFA State Minimization



Non-regular Languages



- The set of **regular languages** is the set of all languages recognized by **DFAs**, **NFAs**, and **regular expressions**

- There also exist languages which cannot be recognized by DFAs, NFAs, nor regular expressions…

Exercise

Consider the following language:

$$\boldsymbol{L} = \{\, \boldsymbol{w} \in \{\mathbf{0}, \mathbf{1}\}^* \mid \boldsymbol{w} \text{ has an equal number of occurrences of } \mathbf{01} \text{ and } \mathbf{10} \,\}$$

Is $\boldsymbol{L}$ a regular language?

Exercise

Are the following languages $\boldsymbol{L_1}$, $\boldsymbol{L_2}$, or $\boldsymbol{L_3}$ regular?

- $\boldsymbol{L_1} = \{\, \boldsymbol{w} \in \{\mathbf{0}, \mathbf{1}\}^* \mid \boldsymbol{w} \text{ has an equal number of } \mathbf{0}\text{s and } \mathbf{1}\text{s} \,\}$

  ?

- $\boldsymbol{L_2} = \{\ \mathbf{0}^{\boldsymbol{n}}\mathbf{1}^{\boldsymbol{n}} \mid \boldsymbol{n} \geq \mathbf{0}\ \}$

  ?

- $\boldsymbol{L_3} = \{\ \mathbf{0}^{\mathbf{3}}\mathbf{1}^{\boldsymbol{n}} \mid \boldsymbol{n} \geq \mathbf{0}\ \}$

  Yes. $\boldsymbol{L_3} = \boldsymbol{L}(\boldsymbol{R})$ with $\boldsymbol{R} = \mathbf{0001}^*$

Non-Regular Languages

- Given a language $\boldsymbol{L}$, if there exists a finite automaton $\boldsymbol{M}$ with $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}$, then $\boldsymbol{L}$ is a **regular language**

- A language is **non-regular** if there exists **no finite automaton** that recognizes it
  - i.e. Not possible to create a finite automaton which **accepts** on every string **in the language** and **not accept** on every string **not in the language**

- Technique for proving that languages are non-regular: **the pumping lemma**

Pumping Lemma

If L is a regular language, then there is a **natural number p** (pumping length of L) such that for **every string $s \in L$** of **length at least p**, s can be **divided into $s = xyz$** satisfying the following:

1. $|y| > 0$ (i.e. $y \neq \varepsilon$)
2. $|xy| \leq p$
3. $xy^iz \in L$ for each $i \geq 0$

Notes:

- y^i means concatenation of i copies of substring y
- Conditions 1 to 3 hold **for all strings** in L that are of length at least p
- We only use the pumping lemma to prove that languages are **non-regular**

Using the Pumping Lemma

Let $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{n}}\mathbf{1}^{\boldsymbol{n}} \mid \boldsymbol{n} \geq \mathbf{0}\,\}$. Prove that $\boldsymbol{L}$ is non-regular.

Proof by contradiction using the pumping lemma:

- Assume for a contradiction that $\boldsymbol{L}$ is regular.
- Then all the properties of the pumping lemma must hold for $\boldsymbol{L}$:
  - There is some natural number $\boldsymbol{p}$
  - For every $\boldsymbol{s} \in \boldsymbol{L}$ and $|\boldsymbol{s}| \geq \boldsymbol{p}$, then there is a way to rewrite $\boldsymbol{s} = \boldsymbol{xyz}$ such that
    1. $|\boldsymbol{y}| > \mathbf{0}$ (i.e. $\boldsymbol{y} \neq \boldsymbol{\varepsilon}$)
    2. $|\boldsymbol{xy}| \leq \boldsymbol{p}$
    3. $\boldsymbol{xy}^{\boldsymbol{i}}\boldsymbol{z} \in \boldsymbol{L}$ for each $\boldsymbol{i} \geq \mathbf{0}$

Using the Pumping Lemma

$L = \{\ 0^n 1^n \mid n \geq 0\ \}$

Proof continued:

- Let p be the pumping length given by the pumping lemma
  - We don't actually know the value of p
  - In fact, there is no such p since **(spoiler)** L is not a regular language
  - We must work with the variable p and come up with a contradiction
- The PL states that conditions must hold **for every** $s \in L$ and $|s| \geq p$
  - For a contradiction, we just need to **choose one string** $\in L$ which contradicts the conditions of the PL
  - Since $|s| \geq p$, p needs to appear somewhere in our chosen string
- Choose $s = 0^p 1^p$

Using the Pumping Lemma

$\boxed{L = \{\ 0^n1^n \mid n \geq 0\,\}}$

- Choose $s = 0^p1^p$
- Since $s \in L$ and $|s| \geq p$, the PL guarantees that s can be rewritten as $s = xyz$ with
  1. $|y| > 0$ (i.e. $y \neq \varepsilon$)
  2. $|xy| \leq p$
  3. $xy^iz \in L$ for each $i \geq 0$

- To derive a contradiction, we need to show that **there is no possible way** to write $s = xyz$ such that the properties hold
- For the PL to be satisfied, there is just needs to be **one** rewriting of s
- So, we need to consider **all ways of rewriting** s and show that **all of them break**

Using the Pumping Lemma

$\boxed{L = \{\ 0^n 1^n \mid n \geq 0\ \}}$

- Choose $s = 0^p 1^p$
- Since $s \in L$ and $|s| \geq p$, the PL guarantees that s can be rewritten as $s = xyz$ with
  1. $|y| > 0$ (i.e. $y \neq \varepsilon$)
  2. $|xy| \leq p$
  3. $xy^i z \in L$ for each $i \geq 0$

$$xyz$$

$$\underbrace{0 \dots 0}_{p}\ \underbrace{1 \dots 1}_{p}$$

- Because of **property 1**, $y \neq \varepsilon$, therefore y can be the following:
  - Case 1: y consists of only 0's
  - Case 2: y consists of only 1's
  - Case 3: y consists of both 0's and 1's

Using the Pumping Lemma

$L = \{\ 0^n 1^n \mid n \geq 0\}$

Case 1: y consists of only $\mathbf{0}$**'s**

- The string $xyyz$ has more $\mathbf{0}$'s than $\mathbf{1}$'s, therefore $xyyz \notin L$
- This violates **property 3** of the PL

Case 2: y consists of only $\mathbf{1}$'s

- The string $xyyz$ has more $\mathbf{1}$'s than $\mathbf{0}$'s, therefore $xyyz \notin L$
- This violates **property 3** of the PL

Case 3: y consists of both $\mathbf{0}$'s and $\mathbf{1}$'s

- The string $xyyz$ has $\mathbf{0}$'s and $\mathbf{1}$'s out of order, therefore $xyyz \notin L$
- This violates **property 3** of the PL

Since no case of y is possible, **we cannot rewrite s as xyz** satisfying the properties of the pumping lemma. Therefore, **L is not regular**.

Pumping Lemma Example 2

Prove that $\boldsymbol{L} = \{\, \boldsymbol{w} \in \{\mathbf{0}, \mathbf{1}\}^* \mid \boldsymbol{w} \text{ has an equal number of } \mathbf{0}\text{s and } \mathbf{1}\text{s} \,\}$ is not regular.

Proof:

- Assume for a contradiction that $\boldsymbol{L}$ is regular
- Let $\boldsymbol{p}$ be the pumping length given by the pumping lemma.
- We choose $\boldsymbol{s} = \mathbf{1}^{\boldsymbol{p}}\mathbf{0}^{\boldsymbol{p}}$.
- Since $\boldsymbol{s} \in \boldsymbol{L}$ and $|\boldsymbol{s}| \geq \boldsymbol{p}$, according to the PL, we can rewrite $\boldsymbol{s} = \boldsymbol{xyz}$ satisfying
    1. $|\boldsymbol{y}| > \mathbf{0}$ (i.e. $\boldsymbol{y} \neq \boldsymbol{\varepsilon}$)
    2. $|\boldsymbol{xy}| \leq \boldsymbol{p}$
    3. $\boldsymbol{xy^iz} \in \boldsymbol{L}$ for each $\boldsymbol{i} \geq \mathbf{0}$

Pumping Lemma Example 2

$L = \{w \in \{0,1\}^* \mid w \text{ has an equal number of } 0\text{s and } 1\text{s}\}$

- We choose $s = 1^p0^p$.
- Since $s \in L$ and $|s| \geq p$, according to the PL, we can rewrite $s = xyz$ satisfying
  1. $|y| > 0$ (i.e. $y \neq \varepsilon$)
  2. $|xy| \leq p$
  3. $xy^iz \in L$ for each $i \geq 0$

$$\overset{xy}{\underbrace{1\ldots1}_{p}}\;\underbrace{0\ldots0}_{p}$$

- By **property 2**, xy must consist of only 1's
- By **property 1**, y must consist of at least one 1 (since $y \neq \varepsilon$)
- Consider $xy^0z = xz$. The string $xz \notin L$ since it contains less 1's than 0's
- We cannot rewrite $s = xyz$ satisfying all properties, so we have a contradiction.

  Therefore, L **is not regular**.

Exercises

1. Show that $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{i}}\mathbf{1}^{\boldsymbol{j}} \mid \boldsymbol{i} > \boldsymbol{j}\}$ is non-regular

2. Show that $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{i}}\mathbf{1}^{\boldsymbol{j}} \mid \boldsymbol{i} < \boldsymbol{j}\}$ is non-regular

3. Show that $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{i}}\mathbf{1}^{\boldsymbol{j}} \mid \boldsymbol{i} \leq \boldsymbol{j}\}$ is non-regular

4. Explain why $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{i}}\mathbf{1}^{\boldsymbol{j}} \mid \boldsymbol{i} \leq \boldsymbol{j} < \mathbf{121}\}$ is regular