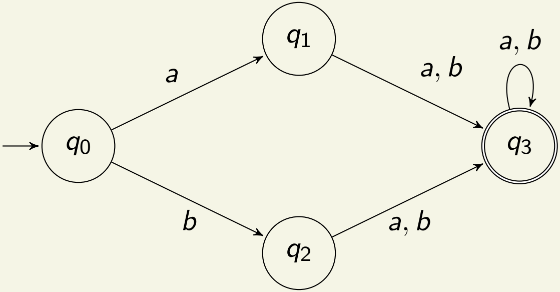
Computability and Complexity
COSC 4200

DFA Minimization

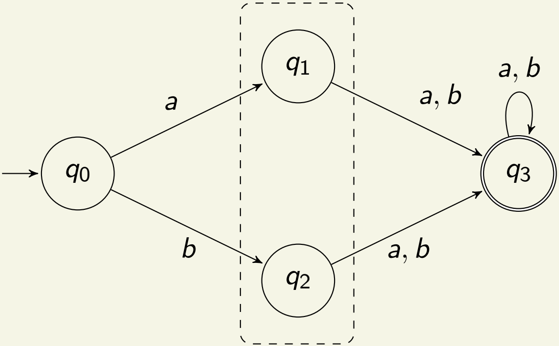
When can we simply a DFA?



q_1 and q_2 are equivalent.

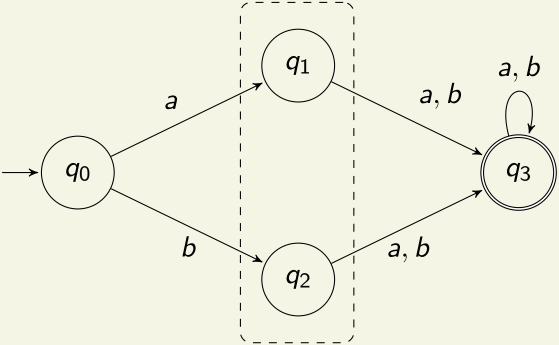
The two states are equivalent because starting from q_1 or q_2, exactly the same strings are accepted.

When can we simply a DFA?

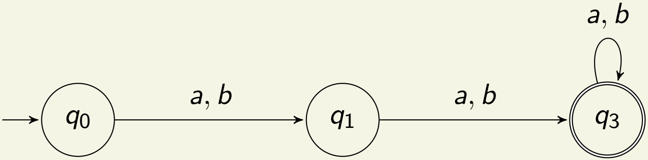


q_1 and q_2 are equivalent.

The two states are equivalent because starting from q_1 or q_2, exactly the same strings are accepted.

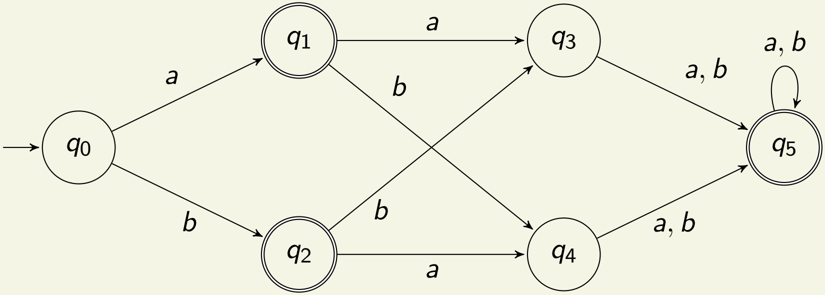


q_1 and q_2 are equivalent. We may collapse these states.

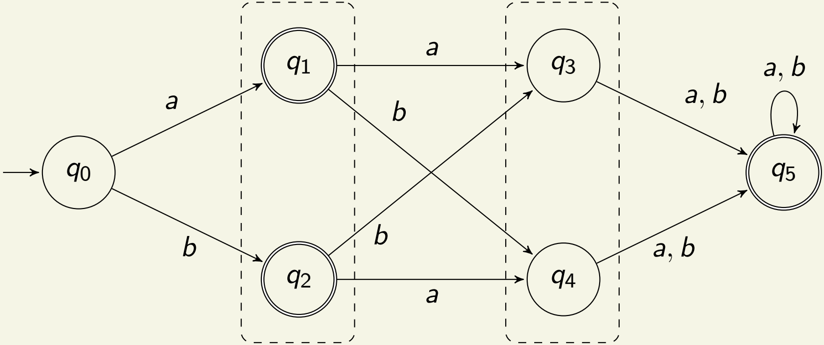


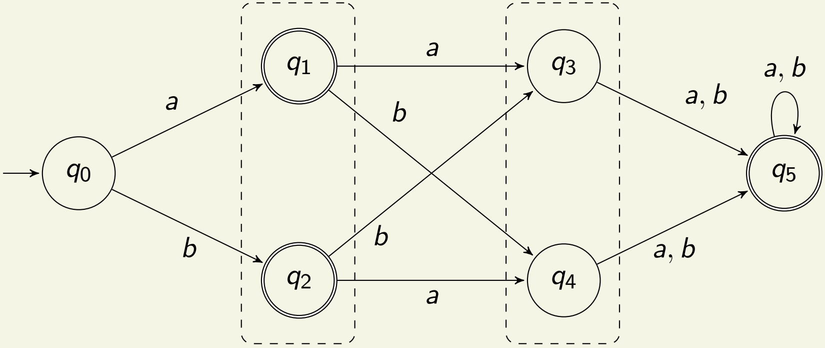
We will develop an algorithm for:

1. Determining which states are equivalent.
2. Building the collapsed automaton.

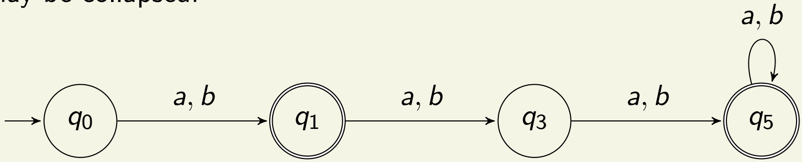


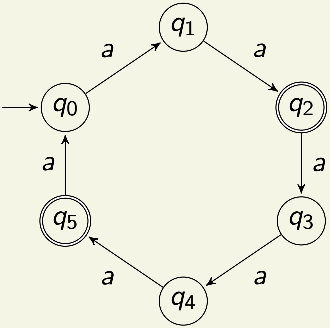
q_0
q_1
q_2
q_3
q_4
q_5
a
b
a
b
b
a
a, b
a, b
a, b


q_0
q_1
q_2
q_3
q_4
q_5
a
b
a
b
b
a
a, b
a, b
a, b

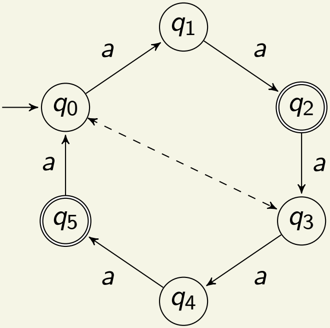


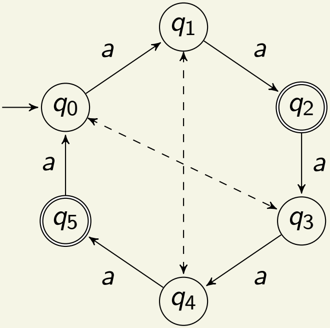
q_1 is equivalent to q_2 and q_3 is equivalent to q_4. These two pairs may be collapsed:

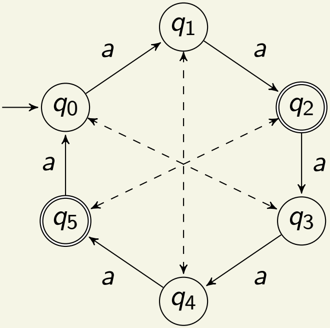


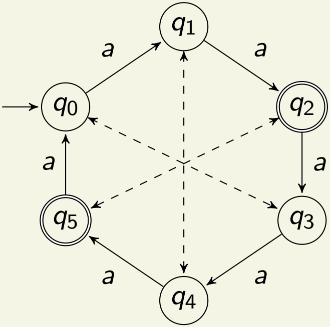


q_0
q_1
q_2
q_3
q_4
q_5
a
a
a
a
a
a

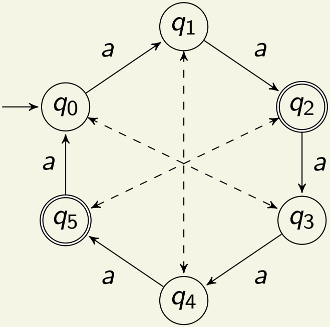

q_0
q_1
q_2
q_3
q_4
q_5
a
a
a
a
a
a


q_0
q_1
q_2
q_3
q_4
q_5
a
a
a
a
a
a

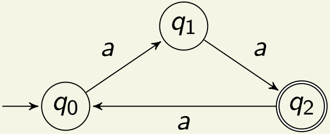

q_1
a
a
q_0
q_2
a
a
q_5
q_3
a
a
q_4



Each state is equivalent with the opposite state in the diagram: q_0 with q_3, q_1 with q_4, q_2 with q_5. Collapsing yields:



Each state is equivalent with the opposite state in the diagram: q_0 with q_3, q_1 with q_4, q_2 with q_5. Collapsing yields:



Equivalent States

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

Intuitively, two states are equivalent if they "remember" the same information. We can say that two states p and q are equivalent if for every $x \in \Sigma^*$,

$$\delta^*(p, x) \in F \Leftrightarrow \delta^*(q, x) \in F.$$

In other words, p and q are equivalent if exactly the same strings are accepted when we start in p or q. We write $p \approx q$ to denote that p is equivalent to q.

Observe that $\approx$ is an equivalence relation:

- *reflexive:* $p \approx p$ for all p
- *symmetric:* $p \approx q$ implies $q \approx p$
- *transitive:* if $p \approx q$ and $q \approx r$, then $p \approx r$

Observe that $\approx$ is an equivalence relation:

- *reflexive:* $p \approx p$ for all p
- *symmetric:* $p \approx q$ implies $q \approx p$
- *transitive:* if $p \approx q$ and $q \approx r$, then $p \approx r$

For each $p \in Q$, let

$$[p] = \{q \in Q \mid q \approx p\}$$

be the equivalence class of p under $\approx$.

Observe that $\approx$ is an equivalence relation:

- *reflexive:* $p \approx p$ for all p
- *symmetric:* $p \approx q$ implies $q \approx p$
- *transitive:* if $p \approx q$ and $q \approx r$, then $p \approx r$

For each $p \in Q$, let

$$[p] = \{q \in Q \mid q \approx p\}$$

be the equivalence class of p under $\approx$.

These equivalence classes partition Q (so each state belongs to exactly one class), and we have

$$p \approx q \Leftrightarrow [p] = [q].$$

Quotient Automaton

We use $\approx$ to define the *quotient automaton*

$$M/\!\approx\ = (Q', \Sigma, \delta', q_0', F')$$

where

- $Q' = \{[p] \mid p \in Q\}$

Quotient Automaton

We use $\approx$ to define the *quotient automaton*

$$M/{\approx} \; = (Q', \Sigma, \delta', q'_0, F')$$

where

- $Q' = \{[p] \mid p \in Q\}$
- $\delta'([p], a) = [\delta(p, a)]$ for all $[p] \in Q'$, $a \in \Sigma$

Quotient Automaton

We use $\approx$ to define the *quotient automaton*

$$M/\!\approx \; = (Q', \Sigma, \delta', q_0', F')$$

where

- $Q' = \{[p] \mid p \in Q\}$
- $\delta'([p], a) = [\delta(p, a)]$ for all $[p] \in Q'$, $a \in \Sigma$
- $q_0' = [q_0]$

Quotient Automaton

We use $\approx$ to define the *quotient automaton*

$$M/\approx \; = (Q', \Sigma, \delta', q_0', F')$$

where

- $Q' = \{[p] \mid p \in Q\}$
- $\delta'([p], a) = [\delta(p, a)]$ for all $[p] \in Q'$, $a \in \Sigma$
- $q_0' = [q_0]$
- $F' = \{[p] \mid p \in F\}$

Quotient Automaton

We use $\approx$ to define the *quotient automaton*

$$M/\!\approx\; = (Q', \Sigma, \delta', q_0', F')$$

where

- $Q' = \{[p] \mid p \in Q\}$
- $\delta'([p], a) = [\delta(p, a)]$ for all $[p] \in Q'$, $a \in \Sigma$
- $q_0' = [q_0]$
- $F' = \{[p] \mid p \in F\}$

We claim that $M/\!\approx$ is equivalent to M, that is $L(M/\!\approx) = L(M)$. First, we need to check that the transition function δ' is well-defined.

The transition function δ' is defined by

$$\delta'([p], a) = [\delta(p, a)]$$

for all $[p] \in Q'$, $a \in \Sigma$.

For δ' to be well-defined, it shouldn't matter which equivalence class representative of $[p]$ we use to apply δ. The following lemma says this is the case.

Lemma

If $p \approx q$, then $\delta(p, a) \approx \delta(q, a)$.
Equivalently, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$.

Lemma

If $p \approx q$, then $\delta(p,a) \approx \delta(q,a)$ for all $a \in \Sigma$.
Equivalently, if $[p] = [q]$, then $[\delta(p,a)] = [\delta(q,a)]$ for all $a \in \Sigma$.

Proof. Suppose $p \approx q$. Then for all $x \in \Sigma^*$,

$$\delta^*(p,x) \in F \iff \delta^*(q,x) \in F.$$

Lemma

If $p \approx q$, then $\delta(p, a) \approx \delta(q, a)$ for all $a \in \Sigma$.
Equivalently, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$ for all $a \in \Sigma$.

Proof. Suppose $p \approx q$. Then for all $x \in \Sigma^*$,

$$\delta^*(p, x) \in F \iff \delta^*(q, x) \in F.$$

Let $a \in \Sigma$. For all $y \in \Sigma^*$,

$$\delta^*(\delta(p, a), y) \in F \quad \iff \quad \delta^*(p, ay) \in F \qquad \text{property of } \delta^*$$

Lemma

If $p \approx q$, then $\delta(p, a) \approx \delta(q, a)$ for all $a \in \Sigma$.
Equivalently, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$ for all $a \in \Sigma$.

Proof. Suppose $p \approx q$. Then for all $x \in \Sigma^*$,

$$\delta^*(p, x) \in F \iff \delta^*(q, x) \in F.$$

Let $a \in \Sigma$. For all $y \in \Sigma^*$,

$$\begin{aligned} \delta^*(\delta(p, a), y) \in F &\iff \delta^*(p, ay) \in F && \text{property of } \delta^* \\ &\iff \delta^*(q, ay) \in F && \text{since } p \approx q \end{aligned}$$

Lemma

If $p \approx q$, then $\delta(p, a) \approx \delta(q, a)$ for all $a \in \Sigma$.
Equivalently, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$ for all $a \in \Sigma$.

Proof. Suppose $p \approx q$. Then for all $x \in \Sigma^*$,

$$\delta^*(p, x) \in F \iff \delta^*(q, x) \in F.$$

Let $a \in \Sigma$. For all $y \in \Sigma^*$,

$$\begin{aligned} \delta^*(\delta(p, a), y) \in F &\iff \delta^*(p, ay) \in F && \text{property of } \delta^* \\ &\iff \delta^*(q, ay) \in F && \text{since } p \approx q \\ &\iff \delta^*(\delta(q, a), y) \in F && \text{property of } \delta^*. \end{aligned}$$

Lemma

If $p \approx q$, then $\delta(p, a) \approx \delta(q, a)$ for all $a \in \Sigma$.
Equivalently, if $[p] = [q]$, then $[\delta(p, a)] = [\delta(q, a)]$ for all $a \in \Sigma$.

Proof. Suppose $p \approx q$. Then for all $x \in \Sigma^*$,

$$\delta^*(p, x) \in F \iff \delta^*(q, x) \in F.$$

Let $a \in \Sigma$. For all $y \in \Sigma^*$,

$$\begin{aligned} \delta^*(\delta(p, a), y) \in F &\iff \delta^*(p, ay) \in F && \text{property of } \delta^* \\ &\iff \delta^*(q, ay) \in F && \text{since } p \approx q \\ &\iff \delta^*(\delta(q, a), y) \in F && \text{property of } \delta^*. \end{aligned}$$

Therefore $\delta(p, a) \approx \delta(q, a)$. $\square$

The quotient automaton $M/\approx$ has the same language as the original automaton M.

Theorem

$L(M/\approx) = L(M)$.

The quotient automaton $M/\approx$ has the same language as the original automaton M.

Theorem

$L(M/\approx) = L(M)$.

The proof follows from these two lemmas:

Lemma A

$p \in F \Leftrightarrow [p] \in F'$.

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Lemma A

$p \in F \iff [p] \in F'$.

Proof. The $\Rightarrow$ direction is immediate from the definition of F'.

Lemma A

$p \in F \iff [p] \in F'$.

Proof. The $\Rightarrow$ direction is immediate from the definition of F'.

For the $\Leftarrow$ direction, we need to show that if $p \approx q$ and $p \in F$, then $q \in F$.

Lemma A

$p \in F \iff [p] \in F'$.

Proof. The $\Rightarrow$ direction is immediate from the definition of F'.

For the $\Leftarrow$ direction, we need to show that if $p \approx q$ and $p \in F$, then $q \in F$. Using $x = \epsilon$ in the definition of $p \approx q$, we have

$$p \in F \quad \iff \quad \delta^*(p, \epsilon) \in F \quad \text{definition of } \delta^*$$

Lemma A

$p \in F \iff [p] \in F'$.

Proof. The $\Rightarrow$ direction is immediate from the definition of F'.

For the $\Leftarrow$ direction, we need to show that if $p \approx q$ and $p \in F$, then $q \in F$. Using $x = \epsilon$ in the definition of $p \approx q$, we have

$$\begin{aligned} p \in F &\iff \delta^*(p, \epsilon) \in F \quad \text{definition of } \delta^* \\ &\iff \delta^*(q, \epsilon) \in F \quad \text{since } p \approx q \end{aligned}$$

Lemma A

$p \in F \iff [p] \in F'$.

Proof. The $\Rightarrow$ direction is immediate from the definition of F'.

For the $\Leftarrow$ direction, we need to show that if $p \approx q$ and $p \in F$, then $q \in F$. Using $x = \epsilon$ in the definition of $p \approx q$, we have

$$\begin{aligned} p \in F &\iff \delta^*(p, \epsilon) \in F && \text{definition of } \delta^* \\ &\iff \delta^*(q, \epsilon) \in F && \text{since } p \approx q \\ &\iff q \in F && \text{definition of } \delta^*. \end{aligned}$$

$\square$

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings.

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{rcll} \delta'^*([p], \epsilon) & = & [p] & \text{definition of } \delta'^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{rcll} \delta'^*([p], \epsilon) & = & [p] & \text{definition of } \delta'^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Assume as our inductive hypothesis the claim

$$\delta'^*([p], x) = [\delta^*(p, x)]$$

holds for some $x \in \Sigma^*$. Let $a \in \Sigma$. For the inductive step, we must show that

$$\delta'^*([p], xa) = [\delta^*(p, xa)].$$

Lemma B

For all $x \in \Sigma^*$, ${\delta'}^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{rcll} {\delta'}^*([p], \epsilon) & = & [p] & \text{definition of } {\delta'}^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Assume as our inductive hypothesis the claim

$${\delta'}^*([p], x) = [\delta^*(p, x)]$$

holds for some $x \in \Sigma^*$. Let $a \in \Sigma$. For the inductive step, we must show that

$${\delta'}^*([p], xa) = [\delta^*(p, xa)].$$

We have

$$\begin{array}{rcll} {\delta'}^*([p], xa) & = & \delta'({\delta'}^*([p], x), a) & \text{definition of } {\delta'}^* \end{array}$$

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{rcll} \delta'^*([p], \epsilon) & = & [p] & \text{definition of } \delta'^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Assume as our inductive hypothesis the claim

$$\delta'^*([p], x) = [\delta^*(p, x)]$$

holds for some $x \in \Sigma^*$. Let $a \in \Sigma$. For the inductive step, we must show that

$$\delta'^*([p], xa) = [\delta^*(p, xa)].$$

We have

$$\begin{array}{rcll} \delta'^*([p], xa) & = & \delta'(\delta'^*([p], x), a) & \text{definition of } \delta'^* \\ & = & \delta'([\delta^*(p, x)], a) & \text{inductive hypothesis} \end{array}$$

Lemma B

For all $x \in \Sigma^*$, ${\delta'}^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{rcll} {\delta'}^*([p], \epsilon) & = & [p] & \text{definition of } {\delta'}^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Assume as our inductive hypothesis the claim

$${\delta'}^*([p], x) = [\delta^*(p, x)]$$

holds for some $x \in \Sigma^*$. Let $a \in \Sigma$. For the inductive step, we must show that

$${\delta'}^*([p], xa) = [\delta^*(p, xa)].$$

We have

$$\begin{array}{rcll} {\delta'}^*([p], xa) & = & \delta'({\delta'}^*([p], x), a) & \text{definition of } \delta'^* \\ & = & \delta'([\delta^*(p, x)], a) & \text{inductive hypothesis} \\ & = & [\delta(\delta^*(p, x), a))] & \text{definition of } \delta' \end{array}$$

Lemma B

For all $x \in \Sigma^*$, $\delta'^*([p], x) = [\delta^*(p, x)]$.

Proof. The proof is by induction on strings. In the base case, we have $x = \epsilon$:

$$\begin{array}{lcll} \delta'^*([p], \epsilon) & = & [p] & \text{definition of } \delta'^* \\ & = & [\delta^*(p, \epsilon)] & \text{definition of } \delta^*. \end{array}$$

Assume as our inductive hypothesis the claim

$$\delta'^*([p], x) = [\delta^*(p, x)]$$

holds for some $x \in \Sigma^*$. Let $a \in \Sigma$. For the inductive step, we must show that

$$\delta'^*([p], xa) = [\delta^*(p, xa)].$$

We have

$$\begin{array}{lcll} \delta'^*([p], xa) & = & \delta'(\delta'^*([p], x), a) & \text{definition of } \delta'^* \\ & = & \delta'([\delta^*(p, x)], a) & \text{inductive hypothesis} \\ & = & [\delta(\delta^*(p, x), a))] & \text{definition of } \delta' \\ & = & [\delta^*(p, xa)] & \text{definition of } \delta^*. \end{array}$$

$\square$

We now use Lemmas A and B to prove the Theorem.

Theorem

$L(M/{\approx}) = L(M)$.

Proof. For any $x \in \Sigma^*$,

$$x \in L(M/{\approx}) \iff \delta'^*(q'_0, x) \in F' \qquad \text{definition of acceptance}$$

We now use Lemmas A and B to prove the Theorem.

Theorem

$L(M/{\approx}) = L(M)$.

Proof. For any $x \in \Sigma^*$,

$$
\begin{aligned}
x \in L(M/{\approx}) \quad &\iff \quad \delta'^*(q_0', x) \in F' && \text{definition of acceptance} \\
&\iff \quad \delta'^*([q_0], x) \in F' && \text{definition of } q_0'
\end{aligned}
$$

We now use Lemmas A and B to prove the Theorem.

Theorem

$L(M/\approx) = L(M)$.

Proof. For any $x \in \Sigma^*$,

$$\begin{array}{rcll} x \in L(M/\approx) & \iff & \delta'^*(q_0', x) \in F' & \text{definition of acceptance} \\ & \iff & \delta'^*([q_0], x) \in F' & \text{definition of } q_0' \\ & \iff & [\delta^*(q_0, x)] \in F' & \text{Lemma B} \end{array}$$

We now use Lemmas A and B to prove the Theorem.

Theorem

$L(M/\approx) = L(M)$.

Proof. For any $x \in \Sigma^*$,

$$
\begin{array}{rcll}
x \in L(M/\approx) & \iff & \delta'^*(q_0', x) \in F' & \text{definition of acceptance} \\
 & \iff & \delta'^*([q_0], x) \in F' & \text{definition of } q_0' \\
 & \iff & [\delta^*(q_0, x)] \in F' & \text{Lemma B} \\
 & \iff & \delta^*(q_0, x) \in F & \text{Lemma A}
\end{array}
$$

We now use Lemmas A and B to prove the Theorem.

Theorem

$L(M/{\approx}) = L(M)$.

Proof. For any $x \in \Sigma^*$,

$$
\begin{array}{rcll}
x \in L(M/{\approx}) & \iff & \delta'^*(q_0', x) \in F' & \text{definition of acceptance} \\
& \iff & \delta'^*([q_0], x) \in F' & \text{definition of } q_0' \\
& \iff & [\delta^*(q_0, x)] \in F' & \text{Lemma B} \\
& \iff & \delta^*(q_0, x) \in F & \text{Lemma A} \\
& \iff & x \in L(M) & \text{definition of acceptance.}
\end{array}
$$

$\square$

Theorem

$L(M/{\approx}) = L(M)$.

It turns out that $M/{\approx}$ cannot be collapsed any further. If we apply the quotient construction a second time, the DFA will not change.

How can we determine whether $p \approx q$?

Minimization Algorithm

1. Make a table of all pairs $\{p, q\}$, initially unmarked.

How can we determine whether $p \approx q$?

Minimization Algorithm

1. Make a table of all pairs $\{p, q\}$, initially unmarked.

2. Mark $\{p, q\}$ if $(p \in F$ and $q \notin F)$ or $(p \notin F$ and $q \in F)$.

How can we determine whether $p \approx q$?

Minimization Algorithm

1. Make a table of all pairs $\{p, q\}$, initially unmarked.
2. Mark $\{p, q\}$ if ($p \in F$ and $q \notin F$) or ($p \notin F$ and $q \in F$).
3. Repeat the following until there are no more changes:
   for each unmarked $\{p, q\}$
   if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$, then

How can we determine whether $p \approx q$?

Minimization Algorithm

1. Make a table of all pairs $\{p, q\}$, initially unmarked.
2. Mark $\{p, q\}$ if ($p \in F$ and $q \notin F$) or ($p \notin F$ and $q \in F$).
3. Repeat the following until there are no more changes:

   for each unmarked $\{p, q\}$

   if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$, then

   mark $\{p, q\}$.

How can we determine whether $p \approx q$?

Minimization Algorithm

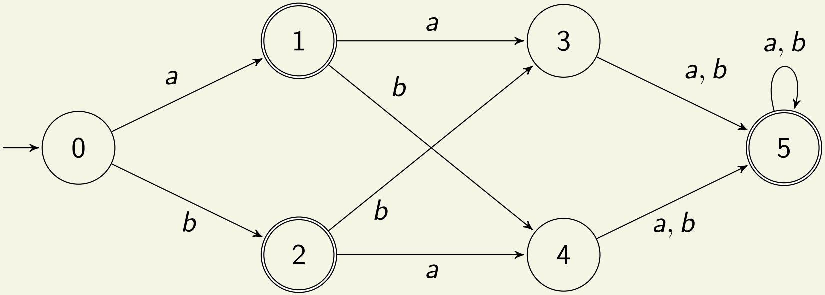
1. Make a table of all pairs $\{p, q\}$, initially unmarked.
2. Mark $\{p, q\}$ if $(p \in F$ and $q \notin F)$ or $(p \notin F$ and $q \in F)$.
3. Repeat the following until there are no more changes:

   for each unmarked $\{p, q\}$

      if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$, then

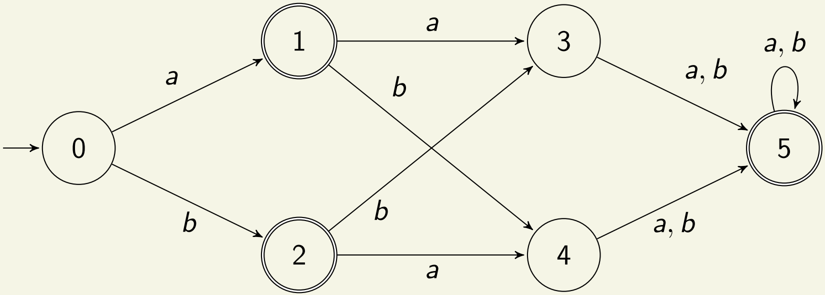
         mark $\{p, q\}$.
4. Now $p \approx q$ if and only if $\{p, q\}$ is not marked.

Example



	0	1	2	3	4
1					
2					
3					
4					
5					

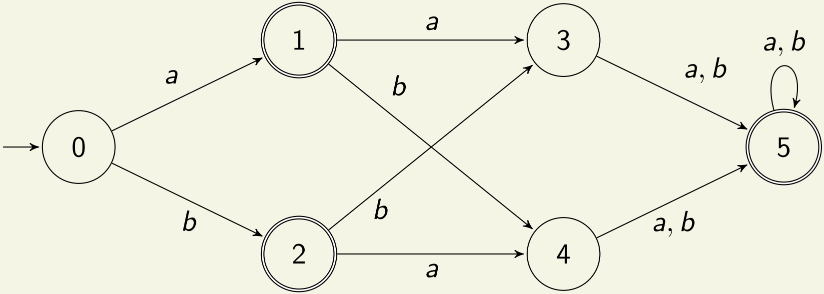
Initialization



Initialization:
Mark $\{p, q\}$ if
$(p \in F$ and $q \notin F)$ or
$(p \notin F$ and $q \in F)$.

	0	1	2	3	4
1	✓				
2	✓				
3		✓	✓		
4		✓	✓		
5	✓			✓	✓

First Pass

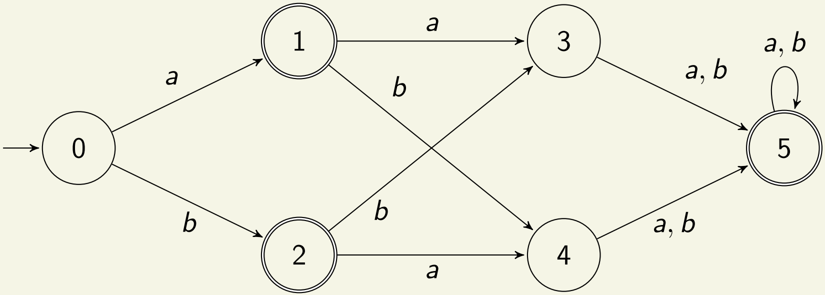


First pass:

- Mark $\{1,5\}$ because $\{1,5\} \xrightarrow{a} \{3,5\}$ is marked
- Mark $\{2,5\}$ because $\{2,5\} \xrightarrow{a} \{4,5\}$ is marked

	0	1	2	3	4
1	✓				
2	✓				
3		✓	✓		
4		✓	✓		
5	✓	✓	✓	✓	✓

Second Pass

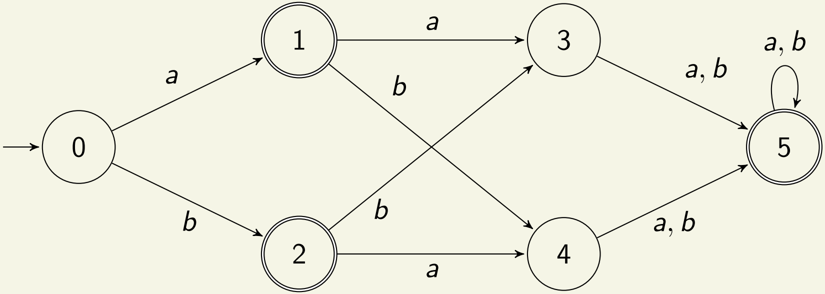


Second pass:

- Mark $\{0,3\}$ because $\{0,3\} \xrightarrow{a} \{1,5\}$ is marked
- Mark $\{0,4\}$ because $\{0,4\} \xrightarrow{b} \{2,5\}$ is marked

	0	1	2	3	4
1	✓				
2	✓				
3	✓	✓	✓		
4	✓	✓	✓		
5	✓	✓	✓	✓	✓

Third Pass



Third pass: no changes

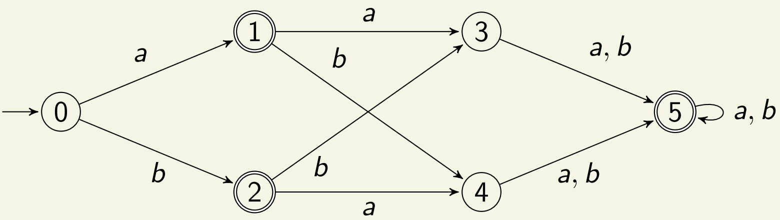
Output of algorithm:

- $1 \approx 2$, $3 \approx 4$
- $[0] = \{0\}$,
  $[1] = \{1, 2\} = [2]$,
  $[3] = \{3, 4\} = [4]$,
  $[5] = \{5\}$

	0	1	2	3	4
1	✓				
2	✓				
3	✓	✓	✓		
4	✓	✓	✓		
5	✓	✓	✓	✓	✓

Quotient Automaton

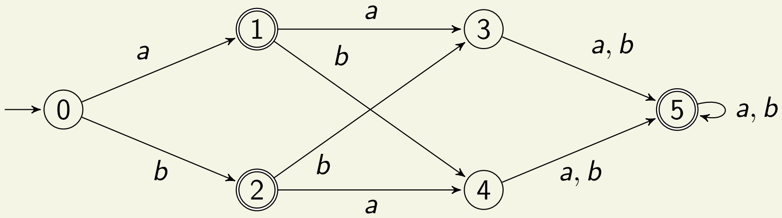
Original DFA:



$$[0] = \{0\},\ [1] = \{1,2\} = [2],\ [3] = \{3,4\} = [4],\ [5] = \{5\}$$

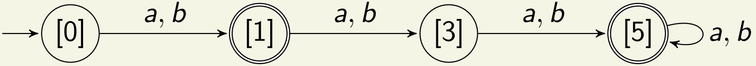
Quotient Automaton

Original DFA:



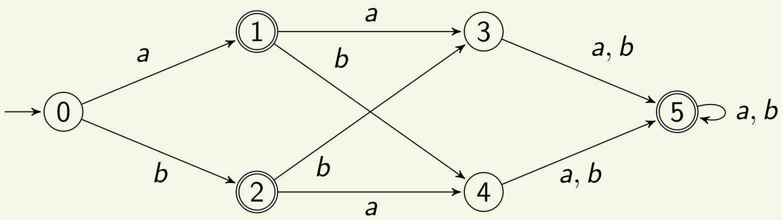
$$[0] = \{0\},\ [1] = \{1, 2\} = [2],\ [3] = \{3, 4\} = [4],\ [5] = \{5\}$$

Quotient Automaton:



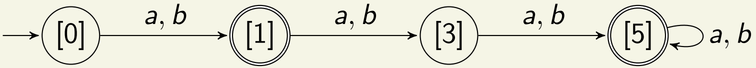
Quotient Automaton

Original DFA:



$$[0] = \{0\},\ [1] = \{1,2\} = [2],\ [3] = \{3,4\} = [4],\ [5] = \{5\}$$

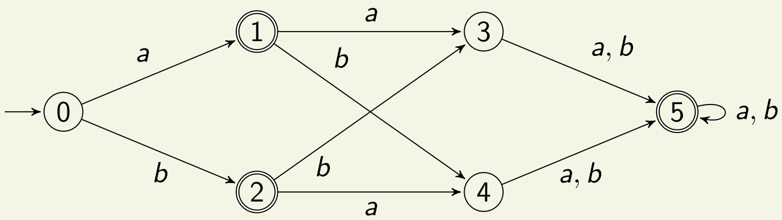
Quotient Automaton:



- $\delta'([0], a) = [\delta(0, a)] = [1]$

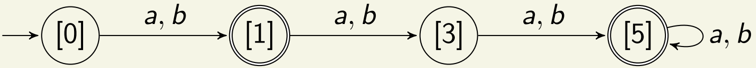
Quotient Automaton

Original DFA:



$$[0] = \{0\},\ [1] = \{1, 2\} = [2],\ [3] = \{3, 4\} = [4],\ [5] = \{5\}$$

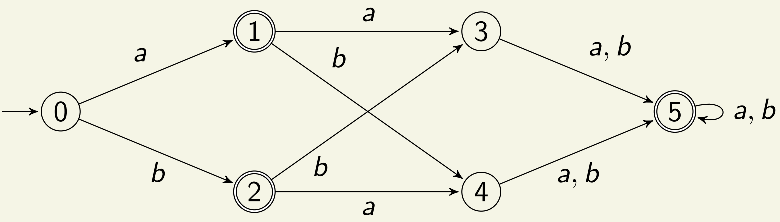
Quotient Automaton:



- $\delta'([0], a) = [\delta(0, a)] = [1]$
- $\delta'([0], b) = [\delta(0, b)] = [2] = [1]$

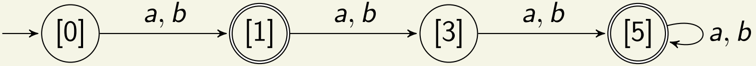
Quotient Automaton

Original DFA:



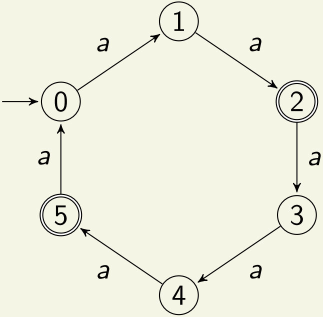
$$[0] = \{0\},\ [1] = \{1, 2\} = [2],\ [3] = \{3, 4\} = [4],\ [5] = \{5\}$$

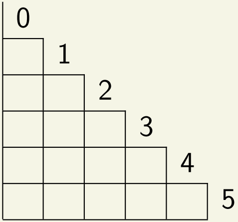
Quotient Automaton:



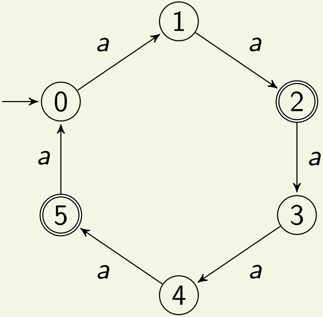
- $\delta'([0], a) = [\delta(0, a)] = [1]$
- $\delta'([0], b) = [\delta(0, b)] = [2] = [1]$
- $\delta'([1], b) = [\delta(1, b)] = [4] = [3]$

Example





Initialization



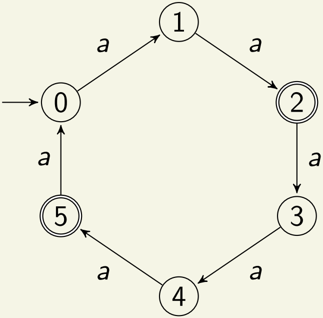
Initialization:
Mark $\{p, q\}$ if
$(p \in F$ and $q \notin F)$ or
$(p \notin F$ and $q \in F)$.

	0	1	2	3	4
1					
2	✓	✓			
3			✓		
4			✓		
5	✓	✓		✓	✓

First Pass

First pass:

- Mark $\{0,1\}$ because $\{0,1\} \xrightarrow{a} \{1,2\}$ is marked
- Mark $\{0,4\}$ because $\{0,4\} \xrightarrow{a} \{1,5\}$ is marked
- Mark $\{1,3\}$ because $\{1,3\} \xrightarrow{a} \{2,4\}$ is marked
- Mark $\{3,4\}$ because $\{3,4\} \xrightarrow{a} \{4,5\}$ is marked



	0	1	2	3	4
1	✓				
2	✓	✓			
3		✓	✓		
4	✓		✓	✓	
5	✓	✓		✓	✓

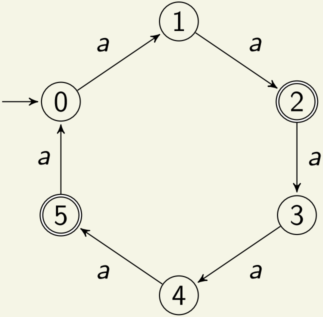
Second Pass

Second pass:

- No changes.

Output of algorithm:

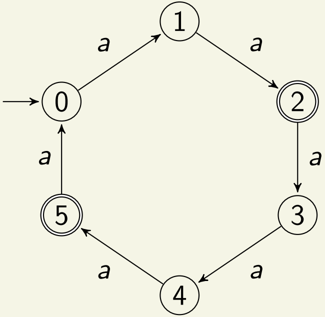
- $0 \approx 3,\ 1 \approx 4,\ 2 \approx 5$
- $[0] = \{0, 3\} = [3]$,
  $[1] = \{1, 4\} = [4]$,
  $[2] = \{2, 5\} = [5]$



	0	1	2	3	4
1	✓				
2	✓	✓			
3		✓	✓		
4	✓		✓	✓	
5	✓	✓		✓	✓

Quotient Automaton

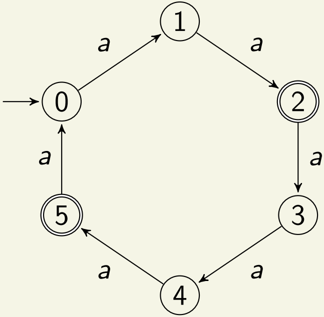
Original DFA:



$$[0] = \{0, 3\} = [3]$$
$$[1] = \{1, 4\} = [4]$$
$$[2] = \{2, 5\} = [5]$$

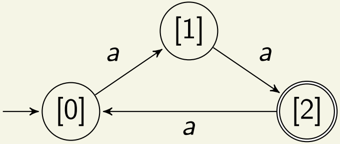
Quotient Automaton

Original DFA:



$$[0] = \{0, 3\} = [3]$$
$$[1] = \{1, 4\} = [4]$$
$$[2] = \{2, 5\} = [5]$$

Quotient Automaton:



$$\delta'([0], a) = [\delta(0, a)] = [1]$$
$$\delta'([1], a) = [\delta(1, a)] = [2]$$
$$\delta'([2], a) = [\delta(2, a)] = [3] = [0]$$