DFA State Minimization

$D = (Q, \Sigma, \delta, s, F)$

$\approx$ on Q : $p \approx q \Leftrightarrow \forall x \in \Sigma^* \left[\hat{\delta}(p, x) \in F \Leftrightarrow \hat{\delta}(q, x) \in F \right]$

Quotient Construction

$D' = D/\approx \; = (Q', \Sigma, \delta', s', F')$

$Q' = Q/\approx$

$s' = [s]$, $F' = \{ [p] \mid p \in F \}$

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

No further state collapse is possible on D'.

Algorithm to compute D' from D.

Build a 2-D table. $\longrightarrow$ indexed by the states

p, q — unmarked if $p \approx q$ $\quad$ —

$\quad\;\;$ — marked if $p \not\approx q$. $\quad$ ✓, X

$p \approx q$

$q \approx p$

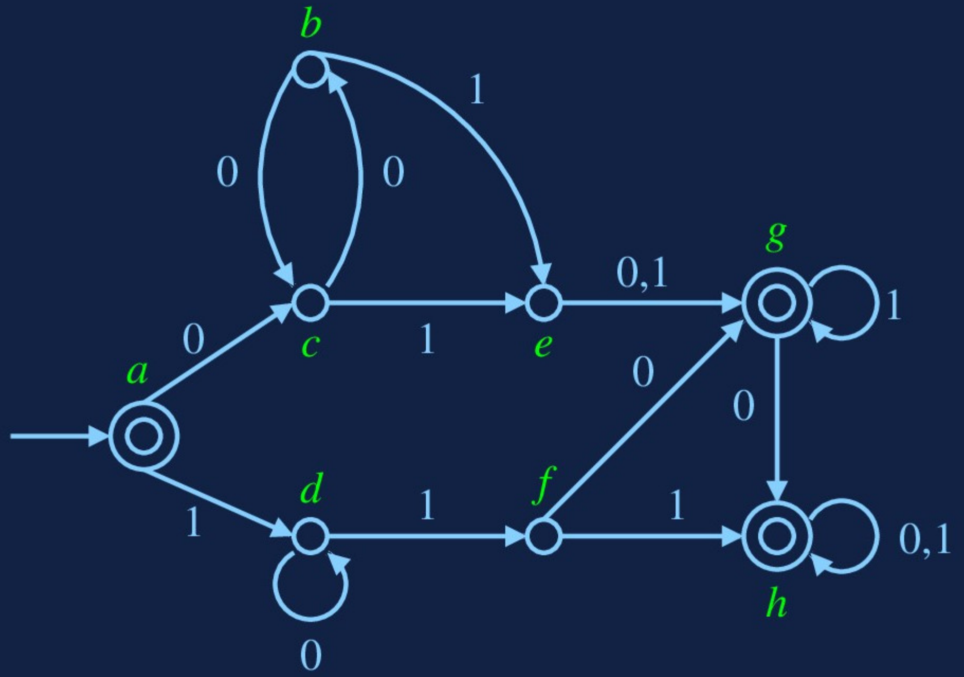
$\{p, q\}$

1. Start with all entries unmarked.

2. [<u>Initialization</u>] Mark $\{p, q\}$ if $(p \in F$ and $q \notin F)$ <u>or</u> $(p \notin F$ and $q \in F)$.

3. [Loop] Find $\{p, q\}$ and $a \in \Sigma$ such that $\{\delta(p, a), \delta(q, a)\}$ is marked.
   Mark $\{\delta(p, a), \delta(q, a)\}$
   Repeat until no further marking is possible.

4. Declare $p \approx q$ if and only if $\{p, q\}$ is unmarked.



	0	1
a	c	d
b	c	e
c	b	e
d	d	f
e	g	g
f	g	h
g	h	g
h	h	h

After initialization (end of Step 2)

	a	b	c	d	e	f	g
b	×						
c	×	–					
d	×	–	–				
e	×	–	–	–			
f	×	–	–	–	–		
g	–	×	×	×	×	×	
h	–	×	×	×	×	×	–

$\delta(a,0) = c$

$\delta(g,0) = h$

At the end of Step 3

	a	b	c	d	e	f	g
b	×						
c	×	–					
d	×	–	–				
e	×	×	×	×			
f	×	×	×	×	–		
g	×	×	×	×	×	×	
h	×	×	×	×	×	×	–

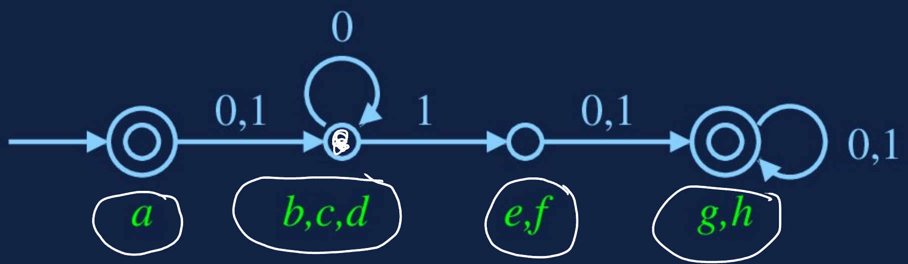
$b \approx c \approx d$

$\delta(c,0) = b$

$\delta(e,0) = g$

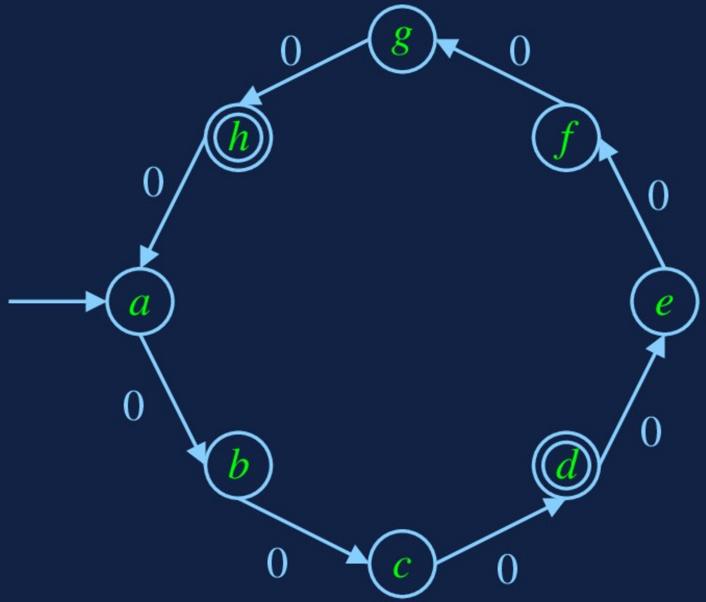
$e \approx f$

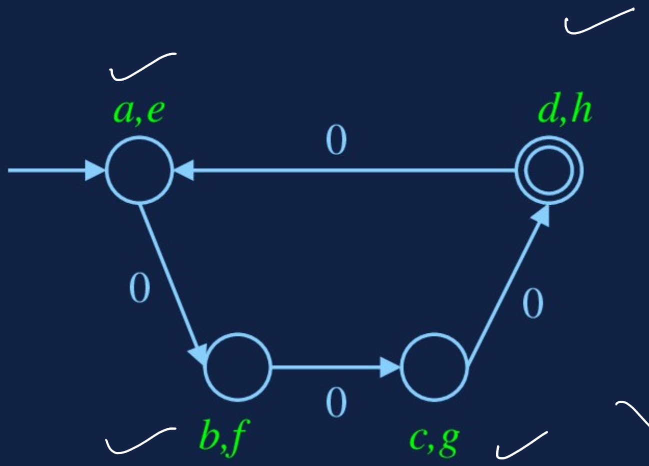
$g \approx h$



$\delta(b,0) = c$, $\delta(b,1) = e$

$\delta(d,0) = d$, $\delta(d,1) = f$





$\Sigma = \{0\}$

$\{a^n \mid n \equiv 3 \pmod 8$
or $n \equiv 7 \pmod 8\}$

$\rightarrow \{a^n \mid n \equiv 3 \pmod 4\}$

Loop

Init

a							
–	b						
–	–	c					
0	0	0	d				
–	–	–	0	e			
–	–	–	0	–	f		
–	–	–	0	–	–	g	
0	0	0	–	0	0	0	h

a							
–	b						
1	1	c					
0	0	0	d				
–	–	1	0	e			
–	–	1	0	–	f		
1	1	–	0	1	1	g	
0	0	0	–	0	0	0	h

a							
2	b						
1	1	c					
0	0	0	d				
–	2	1	0	e			
2	–	1	0	2	f		
1	1	–	0	1	1	g	
0	0	0	–	0	0	0	h

Running time

of unordered pairs $\{p, q\}$ is $\binom{n}{2}$, where $n = |Q|$.

Step 1 : $\Theta(n^2)$

Step 2 : $\Theta(n^2)$

Step 3 : polynomial in n

Efficient algorithm

Step 4 : $O(n^2)$

Claim : This algorithm works correctly.

<u>Theorem</u> : The algorithm marks $\{p, q\}$ if and only if $p \not\approx q$.

<u>Proof</u> " $\Rightarrow$ " [Induction of no. of stages of Step 3]

[Basis] $(p \in F$ and $q \notin F)$ or $(p \notin F$ and $q \in F)$

$$\Rightarrow p \not\approx q$$

[Induction] The algo marks $\{p, q\}$ at a later stage.

$\{\delta(p, a), \delta(q, a)\}$ is marked <u>for some</u> $a \in \Sigma$

$p \approx q$

$\Rightarrow \delta(p, a) \approx \delta(q, a)$

ind. hypo. $\rightarrow \delta(p, a) \not\approx \delta(q, a)$

$$\Rightarrow p \not\approx q$$

" $\Leftarrow$ " $\quad p \not\approx q$ $\quad$ To show that the algorithm marks $\{p, q\}$ at some step/stage.

$\downarrow$

$\exists x \in \Sigma^*$ s.t. $\hat{\delta}(p, x) \in F$ and $\hat{\delta}(q, x) \notin F$

or conversely.

Proceed by induction on $|x|$.

$|x| = 0 \Rightarrow (p \in F$ and $q \notin F)$ or conversely.

The algo marks $\{p, q\}$ in Step 2.

$|x| \geq 1$ $\quad x = ay$ $\quad \hat{\delta}(p, x) = \hat{\delta}(p, ay)$

$$= \hat{\delta}(\delta(p, a), y)$$

$$\left.\begin{array}{l} \hat{\delta}(\underline{\delta(p, a)}, y) \in F \\ \text{and } \hat{\delta}(\underline{\delta(q, a)}, y) \notin F \end{array}\right\} \text{ or conversely}$$

By ind $\{\delta(p, a), \delta(q, a)\}$ is marked.

- A <u>minimal</u> DFA
  - Throw away all unreachable states $\longrightarrow$ DFS/BFS
  - Do the quotient construction to collapse equivalent states. $\to$ Algo covered today

$L = \mathcal{L}(M)$
$= \mathcal{L}(N)$

M, N have no unreachable states.

$M/\approx_M = M'$

$N/\approx_N = N'$

M' and N' are essentially the <u>same</u> DFA.

This minimal DFA is a property of L (the language)