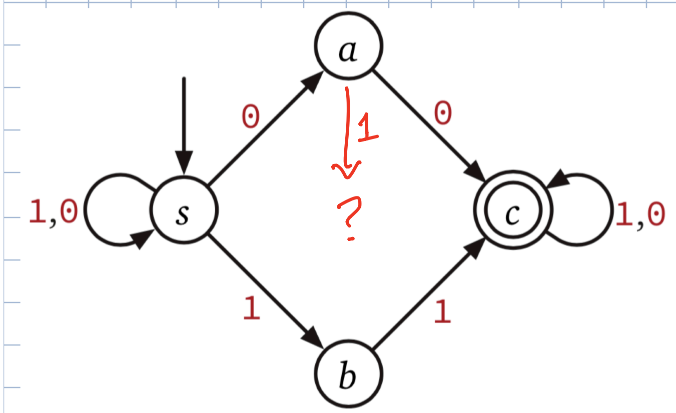


∃ NFAs

NFA accepts $w = a_1 a_2 \cdots a_n$

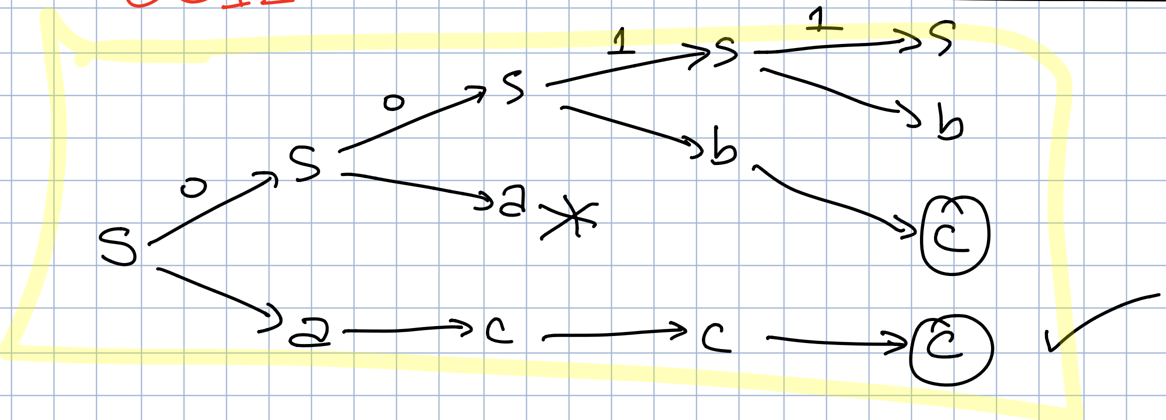
$\Longleftrightarrow$

There is a sequence

$s = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots \xrightarrow{a_n} q_n$

where q_n is accepting

0011



Q - states

Σ - input alphabet

$s \in Q$ start

$A \subseteq Q$ accepting

$\delta : Q \times \Sigma \to 2^Q$

$\delta(s, 0) = \{s, a\}$

$\delta(a, 1) = \emptyset$

$\delta^* : Q \times \Sigma^* \to 2^Q$

$$\delta^*(q, w) = \begin{cases} \{q\} & w = \varepsilon \\ \bigcup_{p \in \delta(q,a)} \delta^*(p, x) & w = ax \end{cases}$$

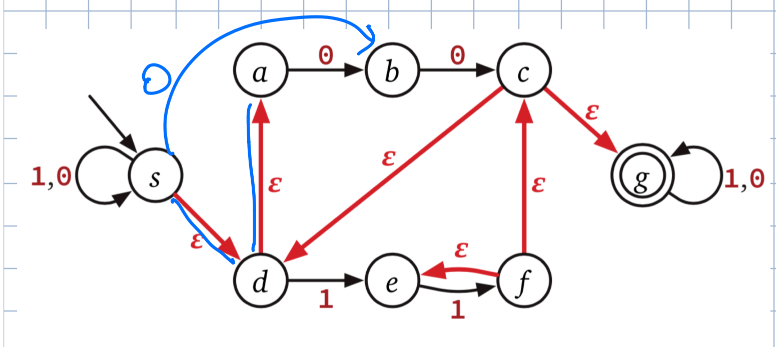
$$L(N) = \{ w \in \Sigma^* \mid \delta^*(s, w) \cap A \neq \emptyset \}$$

Every DFA "is" an NFA

1. NFA $\Rightarrow$ DFA
2. reg. exp. $\Rightarrow$ NFA
3. NFA $\Rightarrow$ reg. exp.

$\Longrightarrow$ Regular $\Longleftrightarrow$ Automatic

ε-transitions



An NFA with ε-transitions

ε-NFA accepts w iff

$\exists$ seq. of transitions

$s = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 - - - \xrightarrow{a_L} q_L$

where $w = a_1 \cdot a_2 \cdot \cdots \cdot a_L$

and each $a_i = \varepsilon$ or $a_i \in \Sigma$

0011

$s \xrightarrow{0} s \xrightarrow{0} s \xrightarrow{\varepsilon} d \xrightarrow{1} e \xrightarrow{1} f \xrightarrow{\varepsilon} c \xrightarrow{\varepsilon} g$

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$

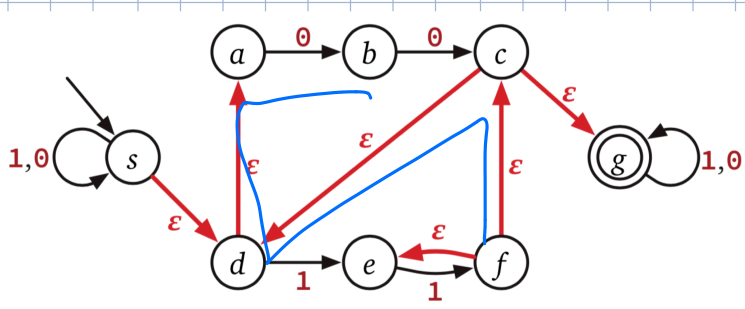
ε-reach(q) = all states reachable from q by ε-transitions

$\varepsilon\text{-reach}(s) = \{s, d, a\}$

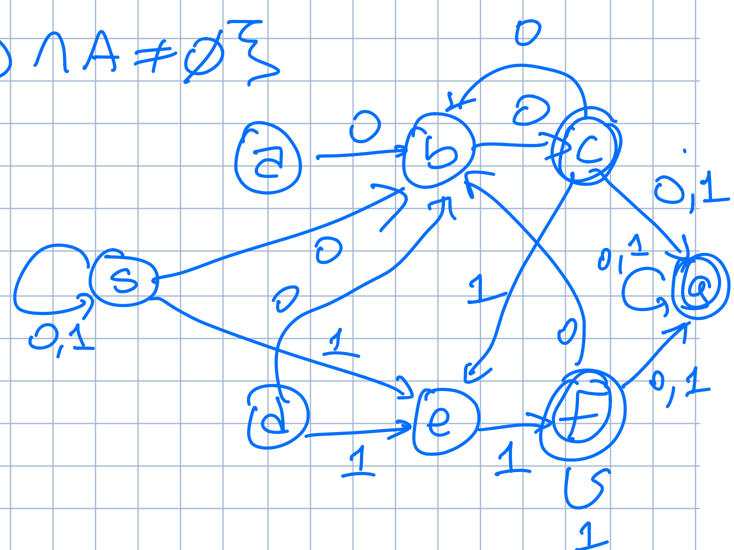
$\delta' : Q \times \Sigma \to 2^Q$

$$\delta'(q, a) = \bigcup_{p \in \varepsilon\text{-reach}(q)} \delta(p, a)$$

$A' = \{q \mid \varepsilon\text{-reach}(q) \cap A \neq \emptyset\}$



An NFA with ε-transitions



NFA → DFA : Subset construction

NFA

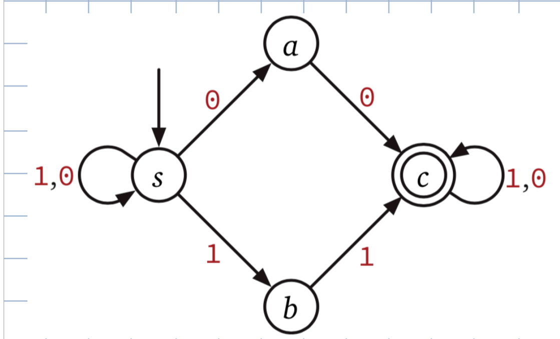
Q
Σ
s
A
$\delta : Q \times \Sigma \to 2^Q$

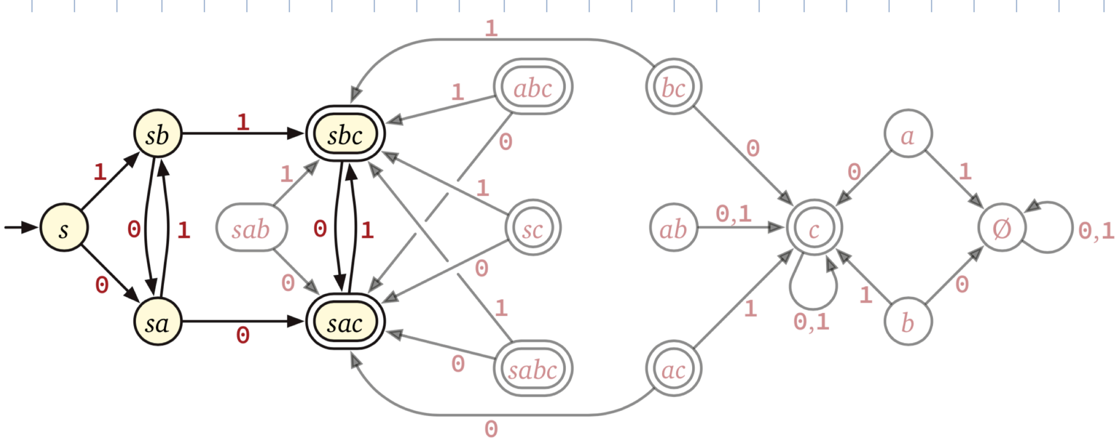
Equivalent DFA:

$$Q' = 2^Q$$
$$s' = \{s\}$$
$$A' = \{S \subseteq Q \mid A \cap S \neq \emptyset\}$$

$$\delta'(q', a) = \bigcup_{q \in q'} \delta(q, a)$$

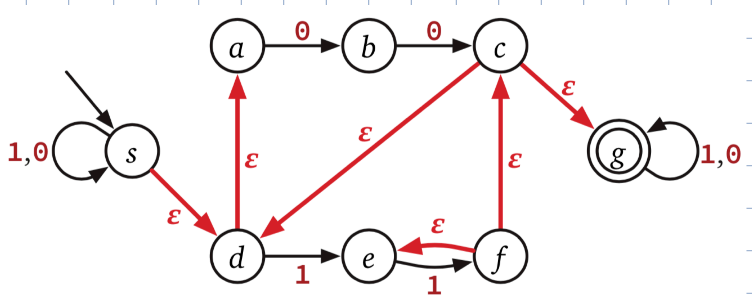
$\boxed{\text{DFA-state} = \text{set of NFA states}}$





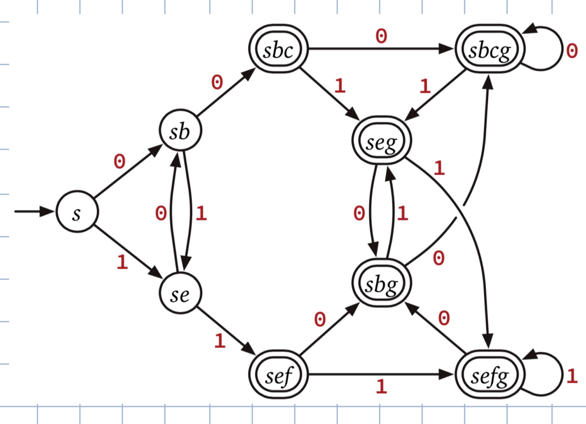
Incremental subset construction

q'	ε-reach	$\delta'(q',0)$	$\delta'(q',1)$	A?
s	sad	sb	se	NO
sb	sabd			
se				



An NFA with ε-transitions

q'	ε-reach(q')	$q' \in A'$?	$\delta'(q', 0)$	$\delta'(q', 1)$
s	sad		sb	se
sb	$sabd$		sbc	se
se	$sade$		sb	sef
sbc	$sabcdg$	✓	$sbcg$	seg
sef	$sacdefg$	✓	sbg	$sefg$
$sbcg$	$sabcdg$	✓	$sbcg$	seg
seg	$sadeg$	✓	sbg	$sefg$
sbg	$sabdg$	✓	$sbcg$	seg
$sefg$	$sacdefg$	✓	sbg	$sefg$



Regular expression $\longrightarrow$ NFA (Thompson)

Given regexp R (tree), computes NFA M s.t. $L(M) = L(R)$

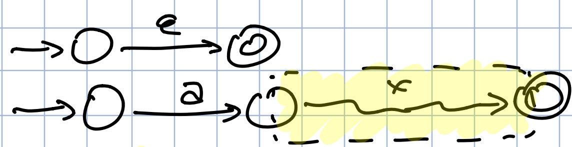
and M has unique accept state $\neq$ start

- $R = \emptyset$

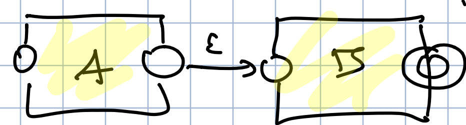
- $R = w$

$w = \varepsilon$:

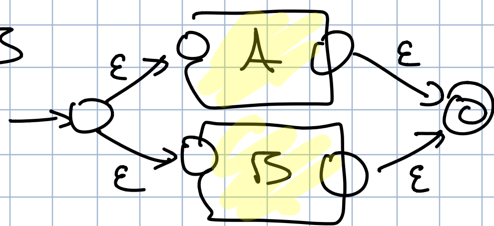
$w = ax$:



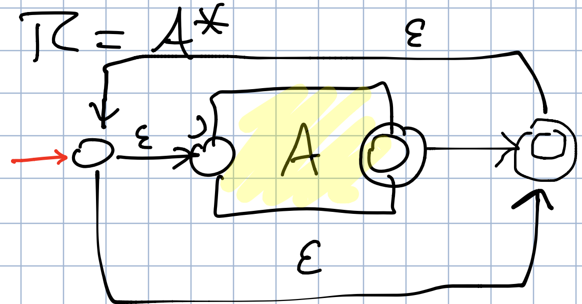
- $R = A \cdot B$



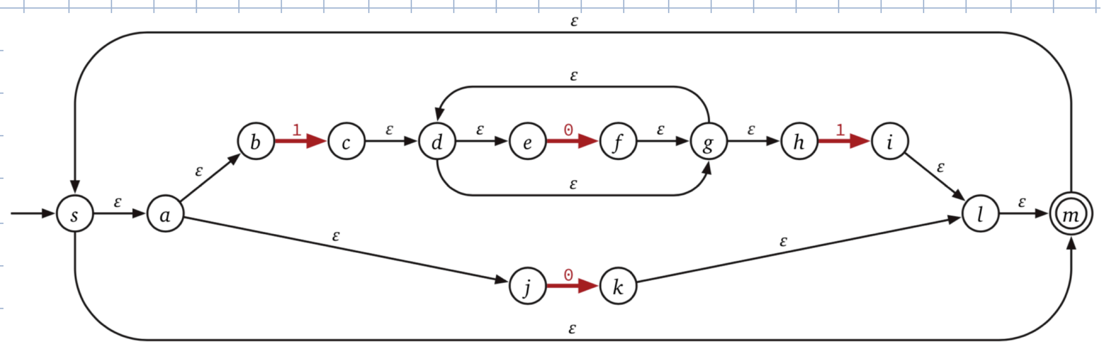
- $R = A + B$



- $R = A^*$

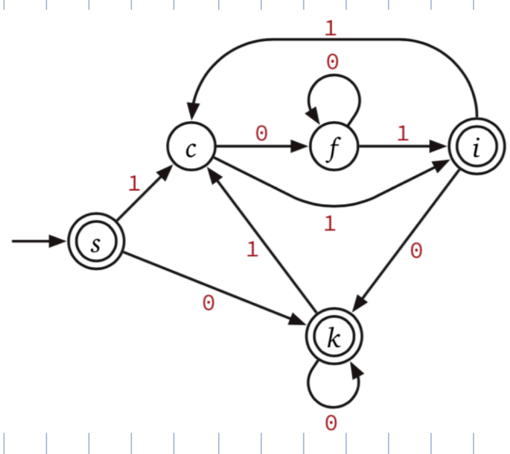


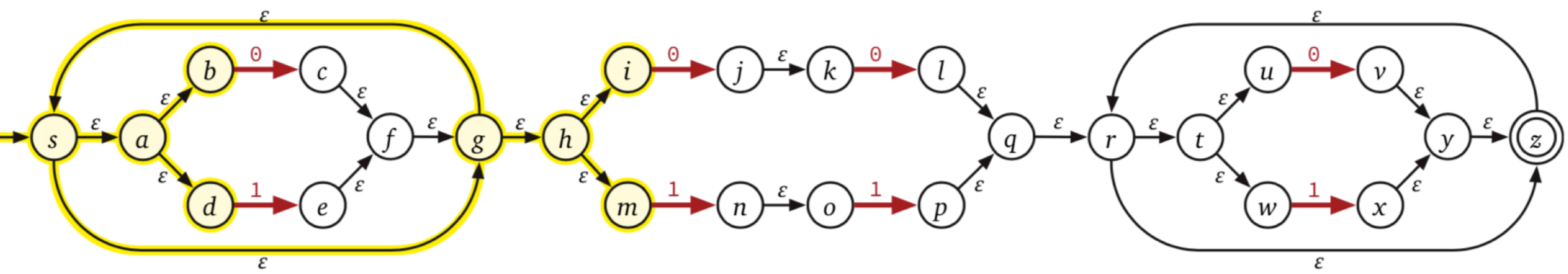
$\boxed{(10^*1 + 0)^*}$



The NFA constructed by Thompson's algorithm for the regular expression $(0 + 10^*1)^*$.

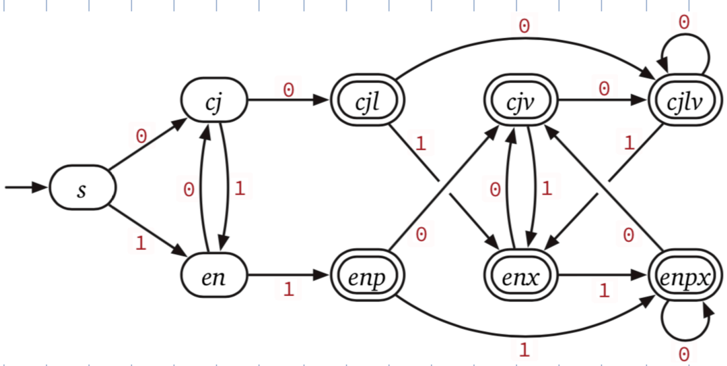
q'	ε-reach(q')	$q' \in A'$?	$\delta'(q', 0)$	$\delta'(q', 1)$
s	$sabjm$	✓	k	c
k	$sabjklm$	✓	k	c
c	$cdegh$		f	i
f	$degh$		f	i
i	$sabjilm$	✓	k	c



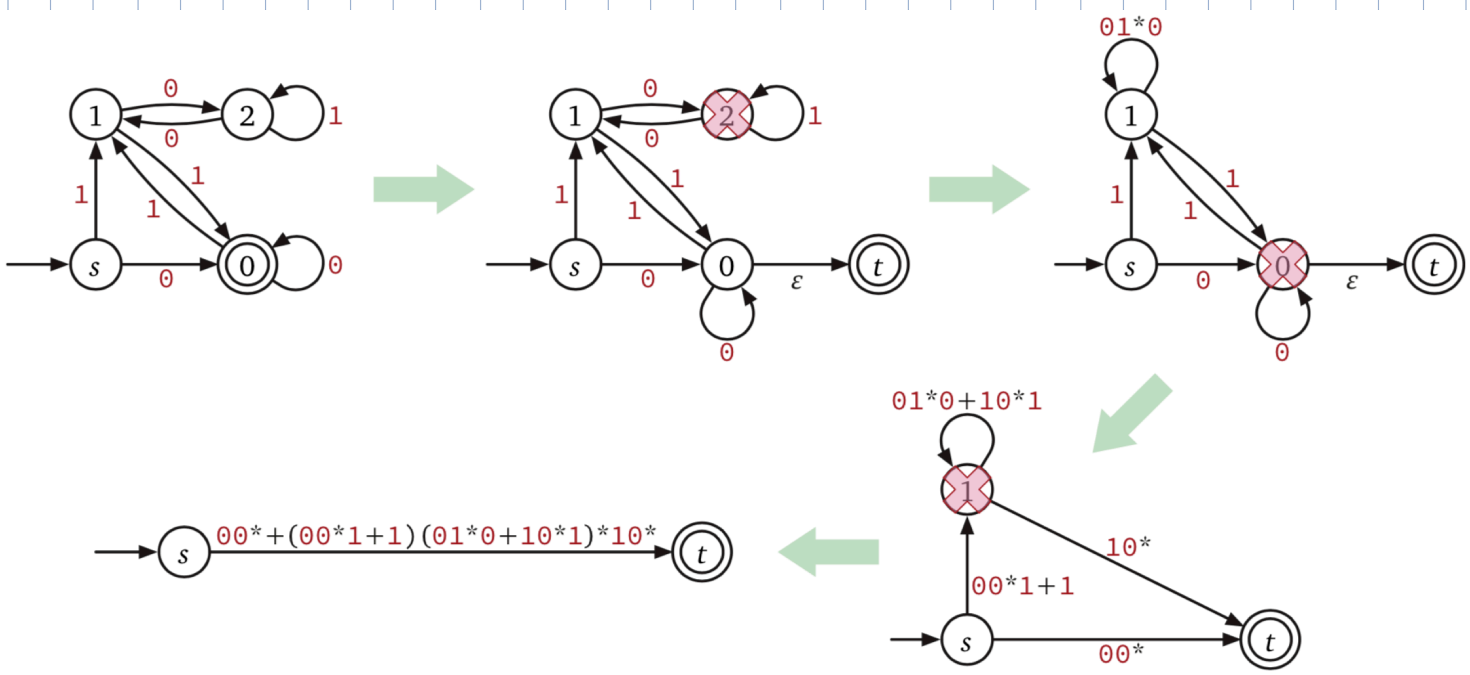


Thompson's NFA for the regular expression $(0+1)^*(00+11)(0+1)^*$,
with the ε-reach of the start state s highlighted.

q'	ε-reach(q')	$q' \in A'$?	$\delta'(q', 0)$	$\delta'(q', 1)$
s	$sabdghim$		cj	en
cj	$sabdfghijkm$		cjl	en
en	$sabdfghmno$		cj	enp
cjl	$sabdfghijklmqrtuwz$	✓	$cjlv$	enx
enp	$sabdfghmnopqrtuwz$	✓	cjv	$enpx$
$cjlv$	$sabdfghijklmqrtuvwyz$	✓	$cjlv$	enx
enx	$sabdfghmnopqrtuwxyz$	✓	cjv	$enpx$
cjv	$sabdfghijkmrtuvwyz$	✓	$cjlv$	enx
$enpx$	$sabdfghmnopqrtuwxyz$	✓	cjv	$enpx$



NFA → Regular expression (Kleene, Han & Wood)



Converting a DFA into an equivalent regular expression using Han and Wood's algorithm.