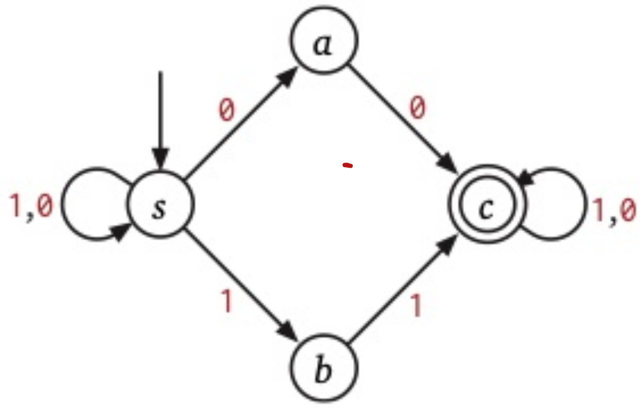
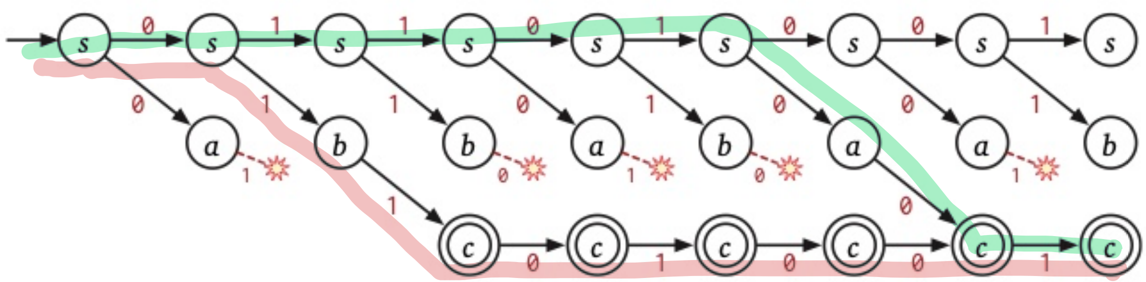
GPS 2 vanishing — fixed

HW parties

Nondeterministic Finite-state Automata



$s \xrightarrow{0} s \xrightarrow{1} s \xrightarrow{1} s \xrightarrow{0} s \xrightarrow{1} s \xrightarrow{0} a \xrightarrow{0} c \xrightarrow{1} c$ Magic!



- Threads
- Magic oracle
- Verification / Proof
- Backtracking ⇝ Dynamic Programming

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

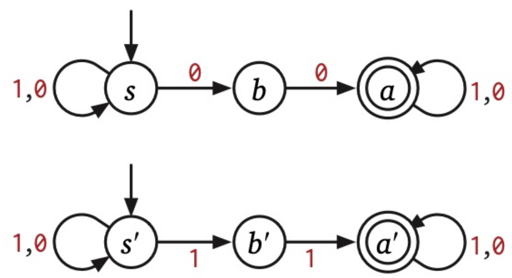
$\delta^* : Q \times \Sigma^* \to 2^Q$

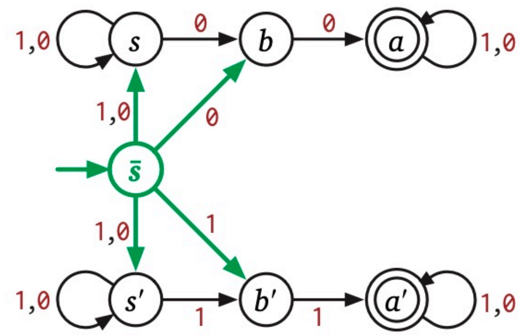
Accept w iff $\delta^*(s,w) \cap A \neq \emptyset$

Kleene's Theorem:

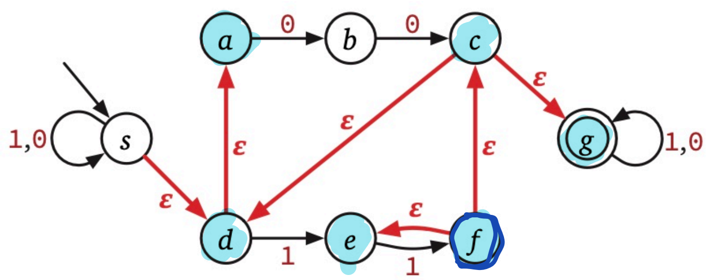
$$\mathrm{DFA} \underset{}{\overset{\text{trivial}}{\rightleftarrows}} \mathrm{NFA} \rightleftarrows \mathrm{Regular}$$

S   ε



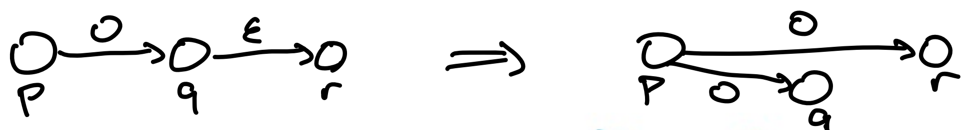


NFA can have multiple start states



$$s \xrightarrow{0} s \xrightarrow{1} s \xrightarrow{1} s \xrightarrow{0} s \xrightarrow{1} s \xrightarrow{\varepsilon} d \xrightarrow{\varepsilon} a \xrightarrow{0} b \xrightarrow{0} c \xrightarrow{\varepsilon} g \xrightarrow{1} g$$

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

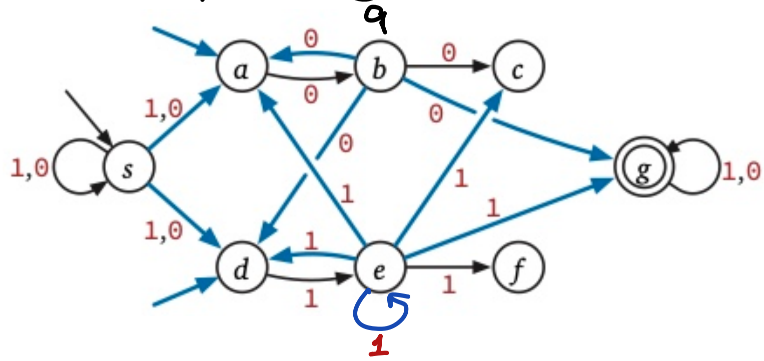


$Q' := Q$

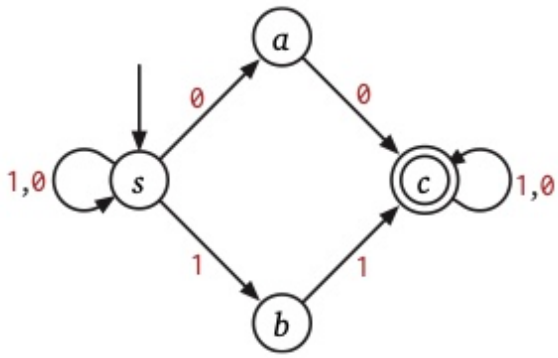
$S' := \varepsilon\text{-}reach(s)$

$A' := A$

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



NFA → DFA



$w = 10010110$

Subset construction

$\{s\} \xrightarrow{0} \{a,s\} \xrightarrow{1} \{b,s\} \xrightarrow{1} \{b,c,s\} \xrightarrow{0} \{a,c,s\} \xrightarrow{0} \{c,s,a\}$

For any subset $P \subseteq Q$,

define $\delta(P,a) = \bigcup_{p \in P} \delta(p,a)$

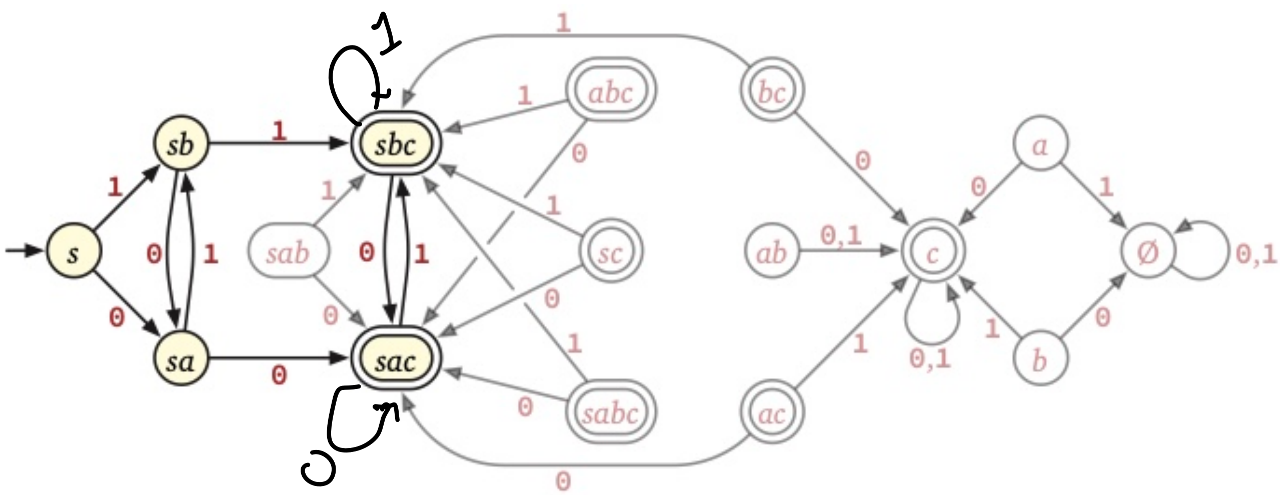
Define new DFA $M' = (Q', s', A', \delta')$

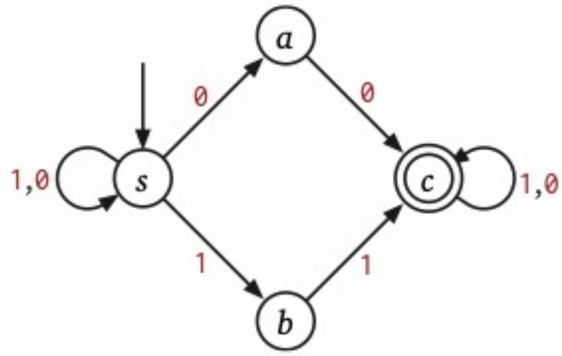
$Q' = 2^Q$

$s' = \{s\}$

$A' = \{P \subseteq Q \mid P \cap A \neq \emptyset\}$

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

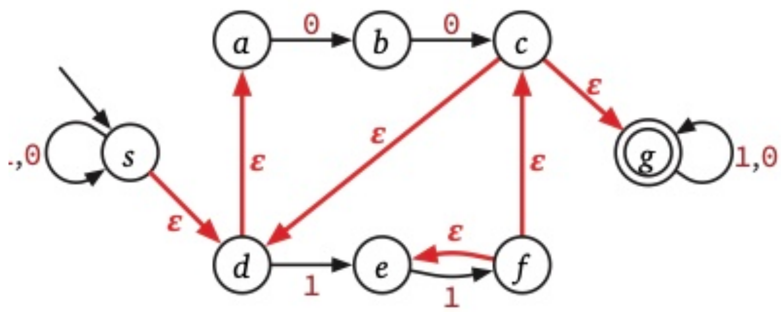




Breadth-First search

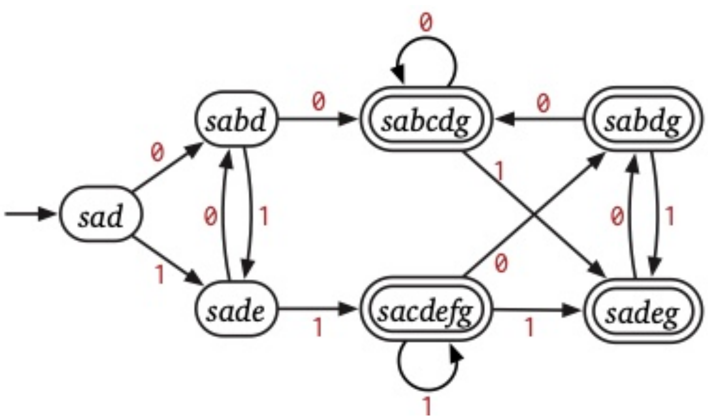
queue

q'	$\delta(q',0)$	$\delta(q',1)$
s	as	bs
as	acs	bs
bs	as	bcs
acs	acs	bcs
bcs	acs	bcs



ε-reach(s)

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

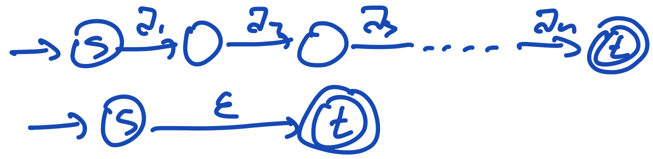


Thompson's Algorithm   Reg Exp → NFA

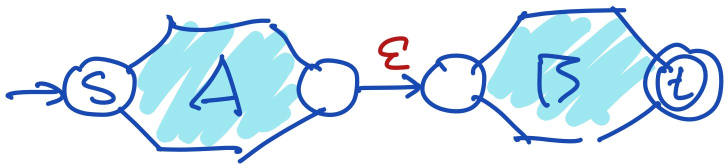
Given reg. expr. R

- $R = \emptyset$

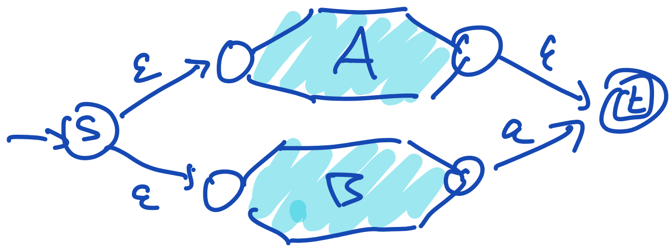
- $R = w = a_1 a_2 \ldots a_n$
  $w = \varepsilon$



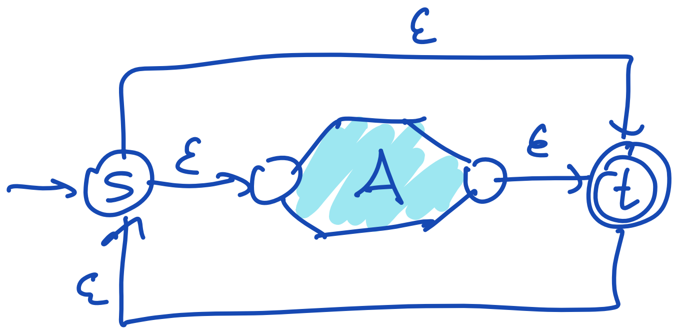
- $R = A \cdot B$

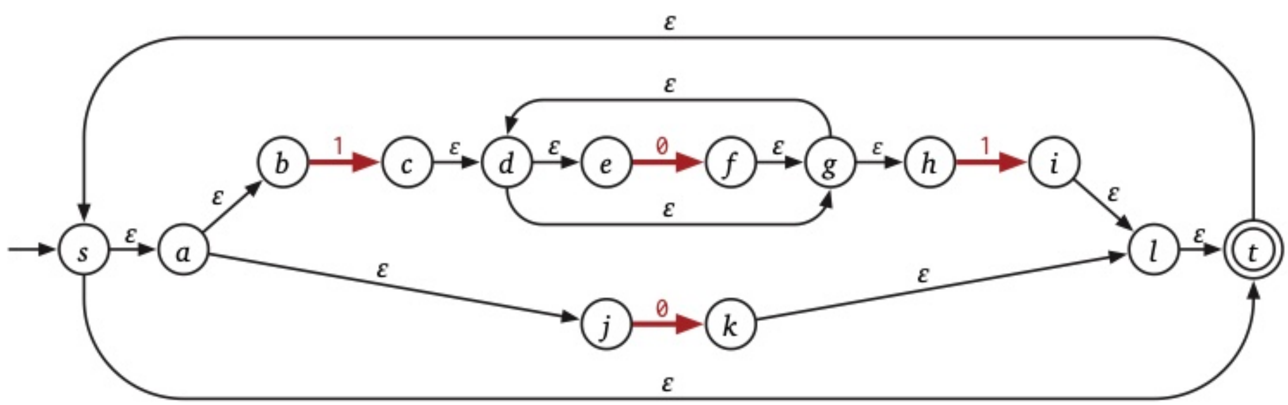


- $R = A + B$



- $R = A^*$





$(10^*1 + 0)^*$

q'	$q' \in A'$?	$\delta'(q', 0)$	ε-reach	$\delta'(q', 1)$	ε-reach
sabjm	✓	k	*sabjklt*	c	*cdegh*
sabjklt	✓	k	*sabjklt*	c	*cdegh*
cdegh		f	*defgh*	i	*sabijlt*
defgh		f	*defgh*	i	*sabijlt*
sabijlt	✓	k	*sabjklt*	c	*cdegh*