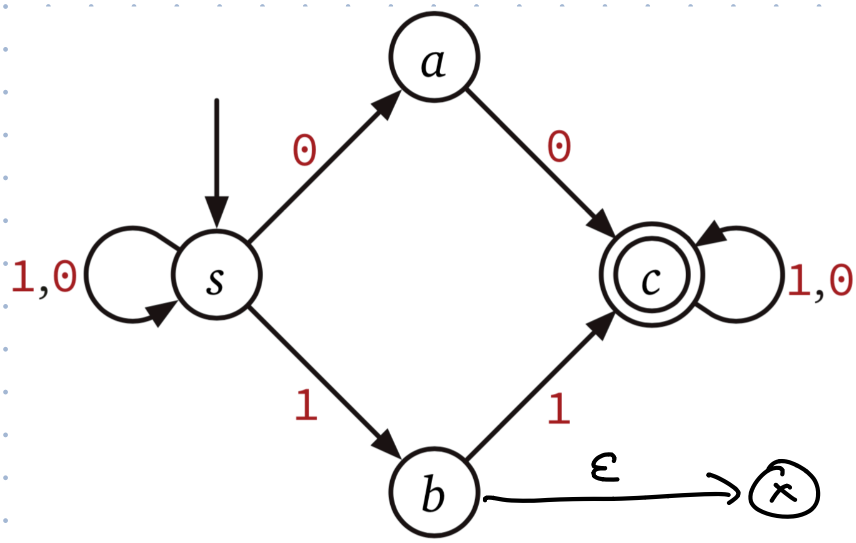
Nondeterministic Finite-state Automata



NFA accepts w if $\exists$ a walk $s \xrightarrow{w_1} q_1 \xrightarrow{w_2} \cdots \xrightarrow{w_n} q_n$ where $q_n \in A$

- Magic oracle
- Parallel threads
- Verification

Accepts 0101100

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

$\delta : Q \times \Sigma \to 2^Q$ $\qquad$ $\delta(s, 0) = \{s, a\}$

---

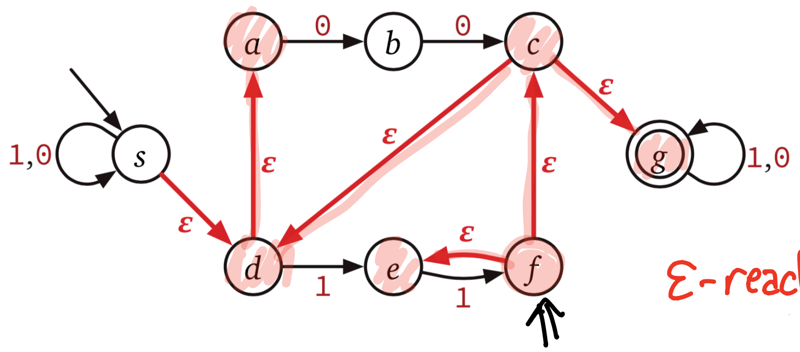
trivial: DFA → NFA

subset: NFA → DFA

dynamic programming: NFA → reg. expr.

parse tree: reg. expr. → NFA

ε-transitions



ε-reach(f) = {a, c, d, e, f, g}

0101100

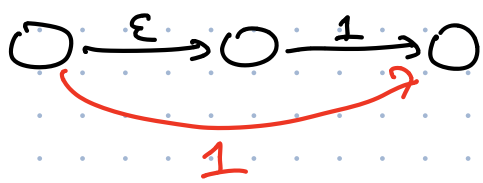
$$s \xrightarrow{0} s \xrightarrow{1} s \xrightarrow{0} s \xrightarrow{\varepsilon} d \xrightarrow{1} e \xrightarrow{1} f \xrightarrow{\varepsilon} c \xrightarrow{\varepsilon} g \xrightarrow{0} g \xrightarrow{0} g$$

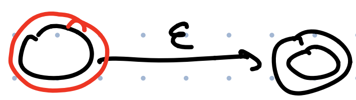
1

0

0

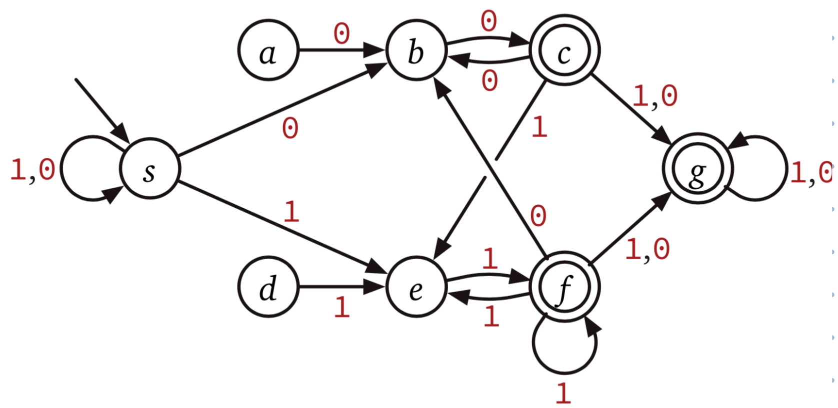
$$\text{NFA} \underset{}{\overset{\text{trivial}}{\rightleftarrows}} \varepsilon\text{-NFA}$$





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

$$Q' := Q$$
$$s' = s$$
$$A' = \{q \in Q \mid \varepsilon\text{-}reach(q) \cap A \neq \varnothing\}$$
$$\delta'(q,a) = \delta(\varepsilon\text{-}reach(q), a) = \bigcup_{p \in \varepsilon\text{-reach}(q)} \delta(p, a)$$



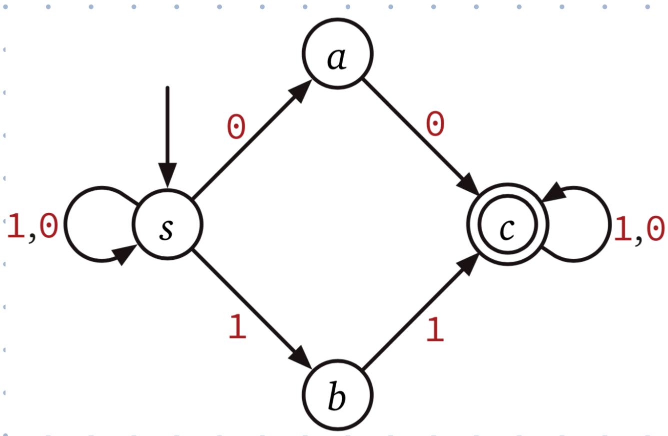
Subset construction: NFA $\longrightarrow$ DFA

$N = \text{NFA} = (Q, s, A, \delta)$ $\qquad \delta : Q \times \Sigma \to 2^Q$

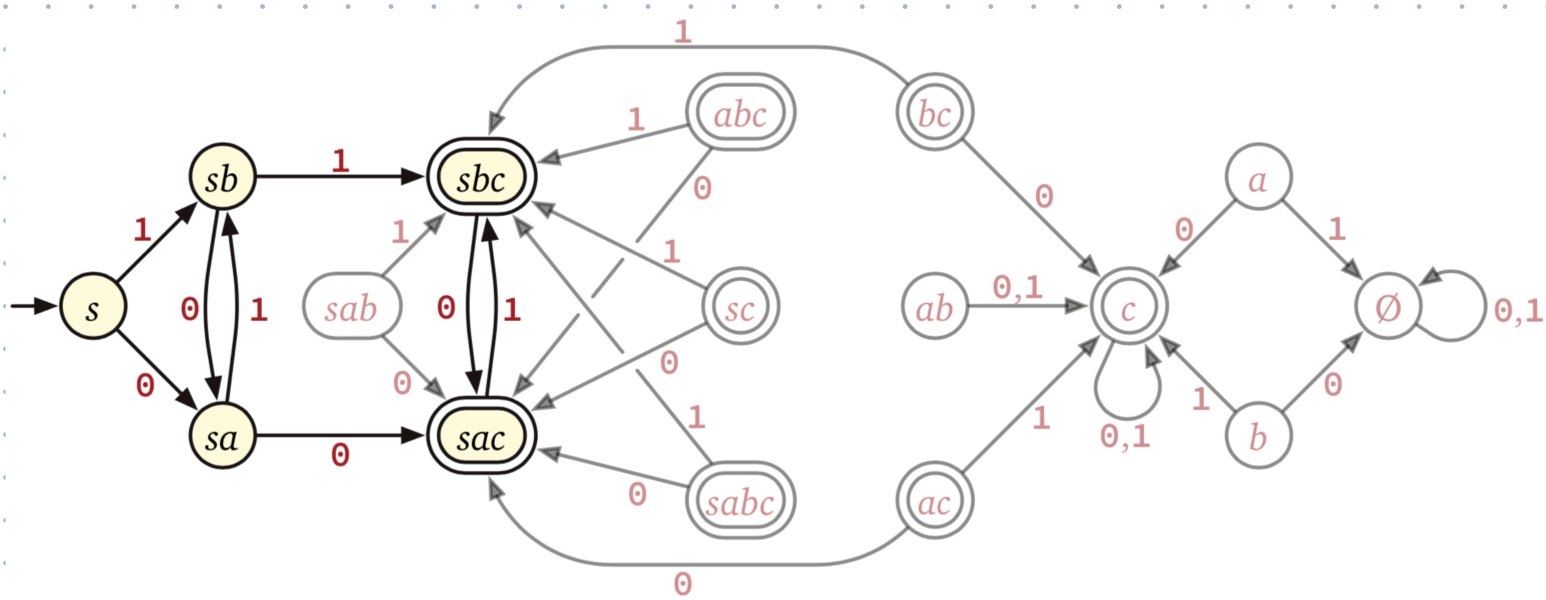
$M = \text{DFA} = (Q', s', A', \delta')$ $\qquad \delta' : Q' \times \Sigma \to Q'$

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

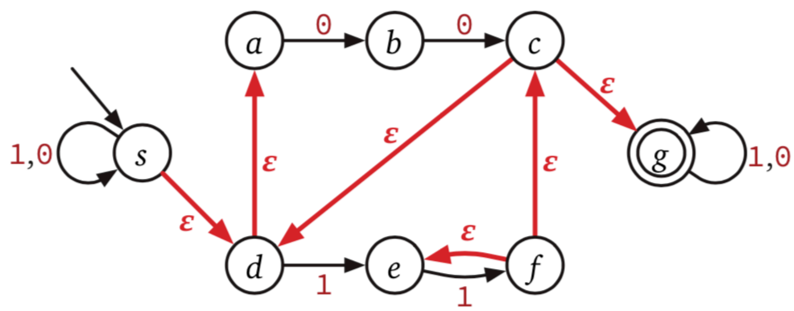
$$\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$$



$Q' = \{\emptyset, \{a\}, \{a,b\}, \{s,c\} \ldots$
$\ldots \{s,a,b,c\}\}$

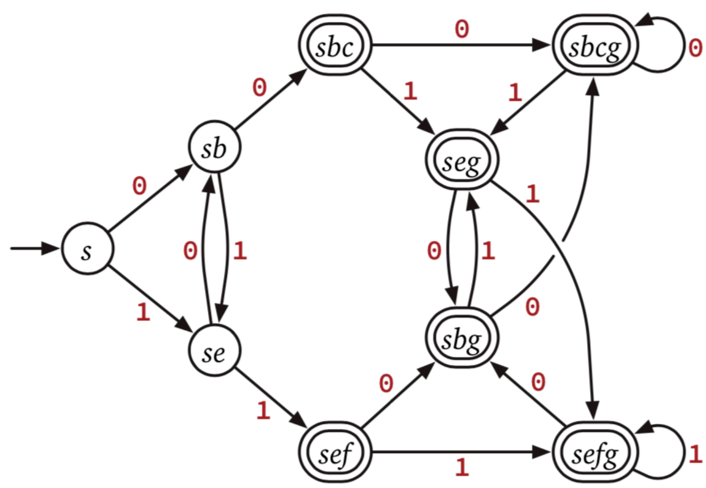


Incremental Subset Construction



S	ε-reach	Acc?	δ(−, 0)	δ(−, 1)
s	sda	×	sb	se
sb	sdab	×	sbc	se
se				
sbc				

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	$sefg$



Thompson's Algorithm: Regular Expression $\to$ εNFA

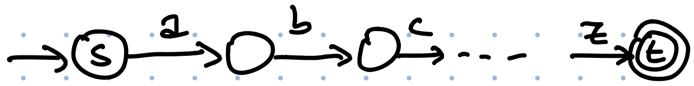
Given a reg. exp. R compute NFA N

s.t $L(R) = L(N)$

<u>5 cases:</u>

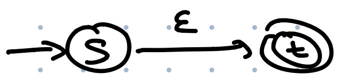
$R = \emptyset$

$R = w = abc\cdots z$

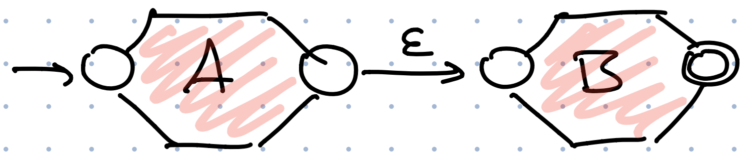


$w = a$

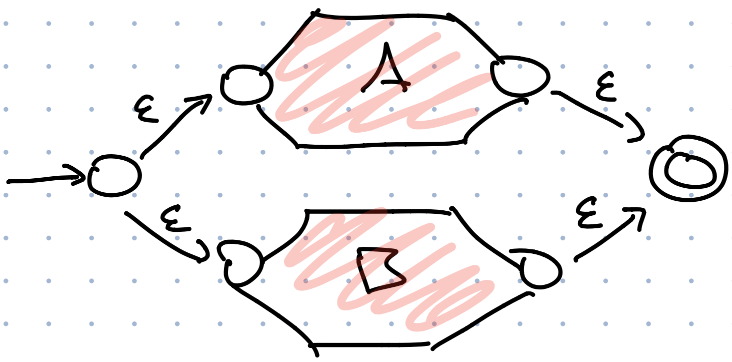
$w = \varepsilon$



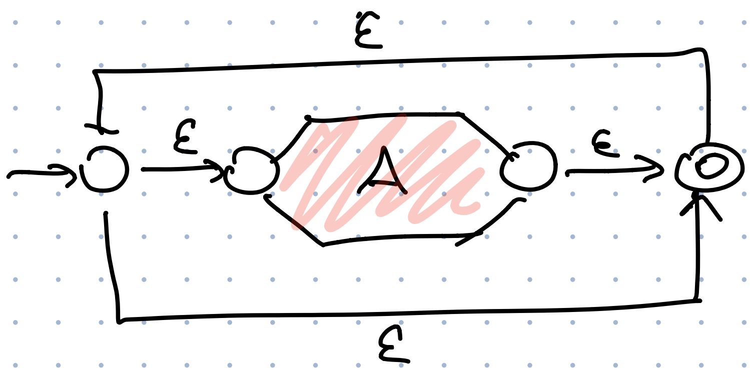
$R = A \cdot B$



$R = A + B$

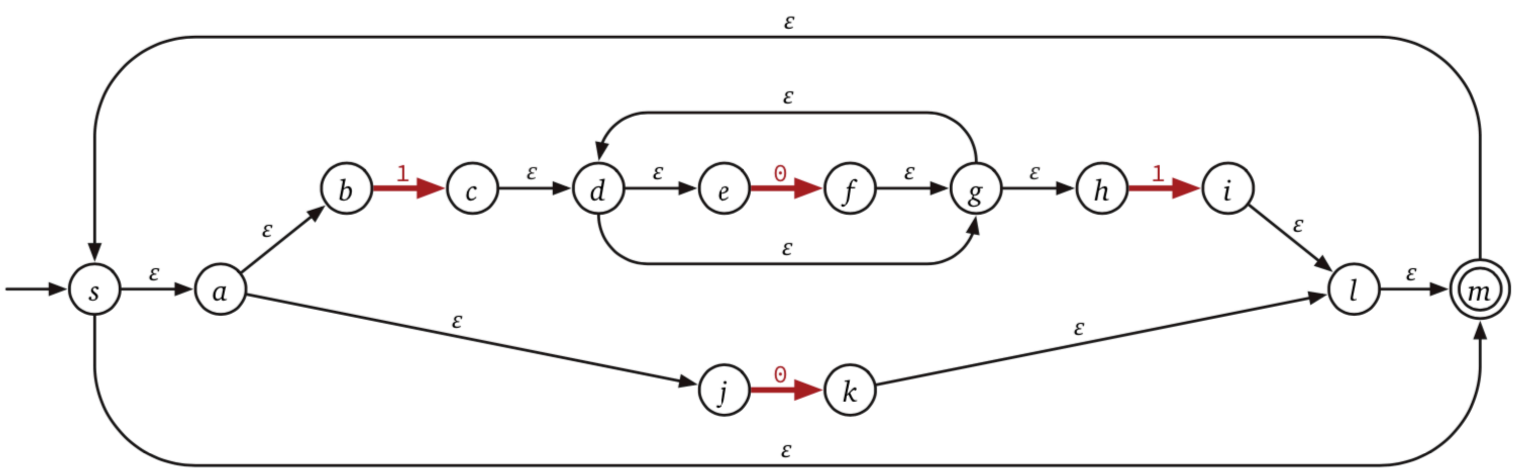


$R = A^*$

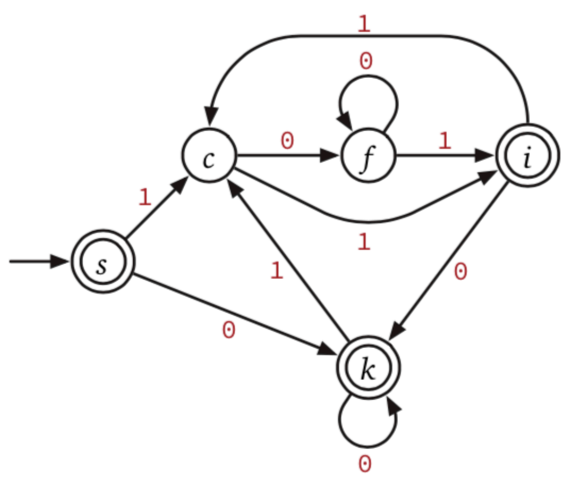


Binary strings with even # of 1s

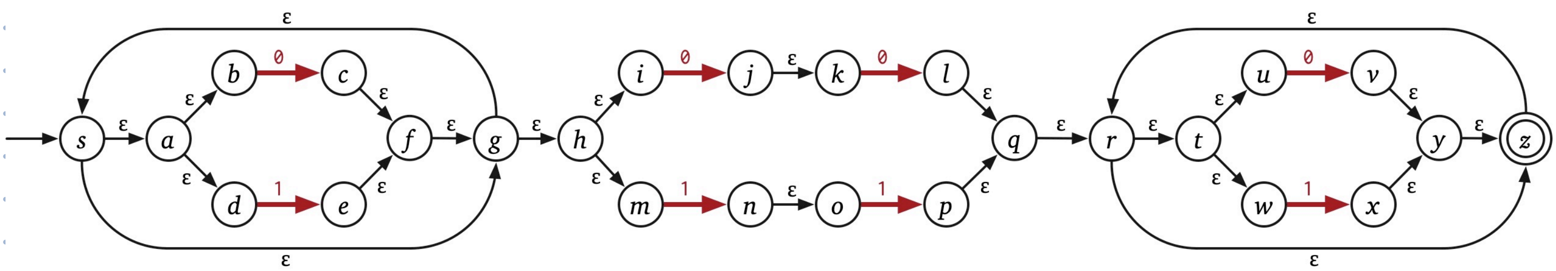
$(10^*1 + 0)^*$



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	$defgh$		f	i
i	$sabjilm$	✓	k	c



$(0+1)^*(00+11)(0+1)^*$



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	$enpx$