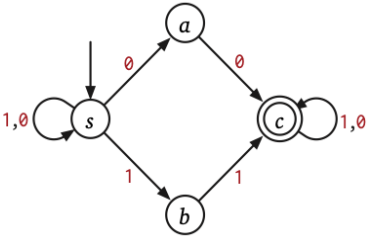
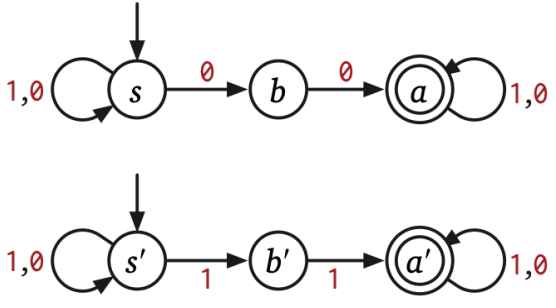
- GPS 3: Mon 15th
  Homework 3: Tue 16th
- Grades for HW1

NFAs

Has 00 or 11.



One NFA with 2 start states

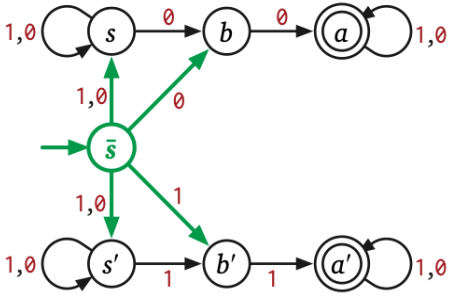


Formally, you have start states

$$S \subseteq Q$$

all states

Accept iff $\bigcup_{s \in S} \delta^*(s, w) \cap A \neq \emptyset$



To go back to one:

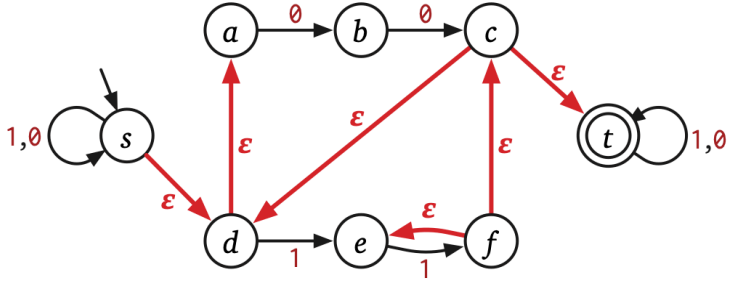
Make new state $\bar{s}$

Copy transitions out of S to leave $\bar{S}$ instead.

$\bar{S}$ accepts iff $S \cap A \neq \emptyset$

$\bar{s}$ is the start state.

ε-transitions



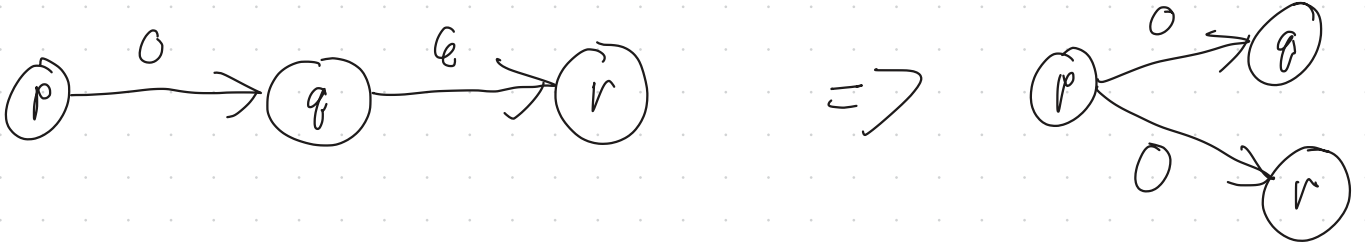
Can follow ε-transitions without reading a character.

Formally, $\varepsilon\text{-reach}(q)$: $(q \in Q)$

$q \in \varepsilon\text{-reach}(q)$ → states reachable from q via ε-transitions

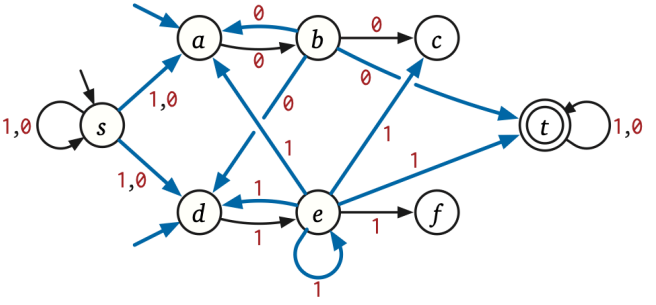
$$\delta^*(p, w) := \begin{cases} \varepsilon\text{-reach}(p) & \text{if } w = \varepsilon \\ \bigcup_{r \in \varepsilon\text{-reach}(p)} \bigcup_{q \in \delta(r, a)} \delta^*(q, x) & \text{if } w = a x \end{cases}$$

Given NFA $M = (\Sigma, Q, S, A, \delta)$ with ε-transitions, can make one without $M' = (\Sigma, Q', S', A', \delta')$



$Q' = Q$

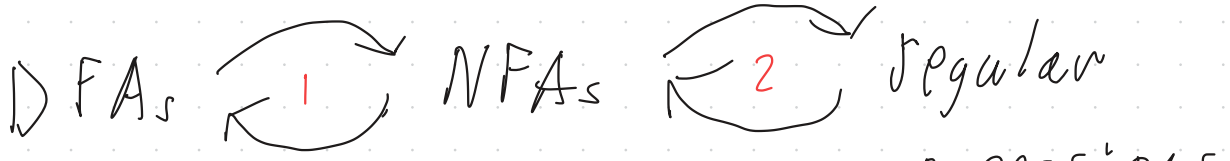
$S' = \bigcup_{s \in S} \varepsilon\text{-reach}(s)$



$A' = A$

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

Kleene's Theorem



DFAs $\Rightarrow$ NFAs ✓

NFAs $\Rightarrow$ DFAs

Maintain subsets of states

Given NFA $M = (\Sigma, Q, s, A, \delta)$, use a subset construction

Make a DFA $M' = (\Sigma, Q', s', A', \delta')$

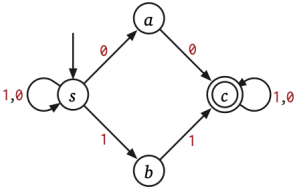
$Q' = 2^Q$

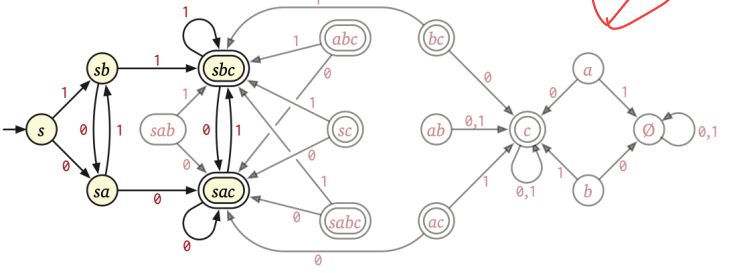
$s' = \{s\}$

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

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

lots of unreachable states!



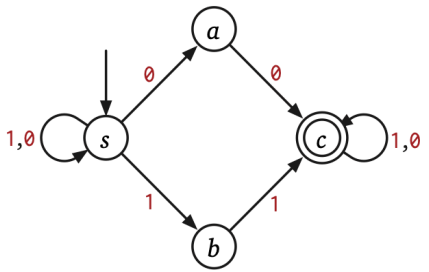


BFS: Table with $2+|\Sigma|$ columns and one row per reachable DFA state

columns: DFA state | is accepting | result of each symbol

start with row for s

while there is a not filled out row, compute transitions and add new DFA states as new rows

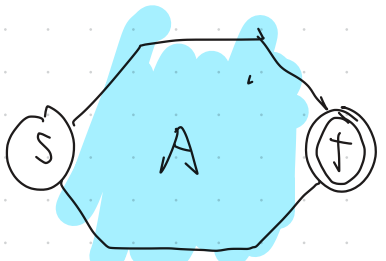


q'	$q' \in A'$?	$\delta'(q', 0)$	$\delta'(q', 1)$
s	no	sa	sb
sa	no	sac	sb
sb	no	sa	sbc

sac	yes	sac	s b c
sbc	yes	sac	sbe

Regular expression ⇒ NFA (with ε-transitions)

↑ Thompson's algorithm ['68]: Makes an NFA with ε-transitions with
- one start state s
- accept state t
- $s \neq t$



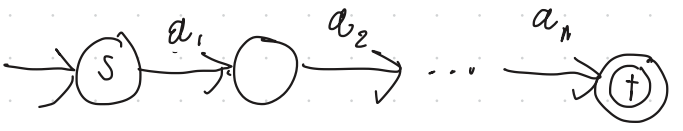
← NFA built for (sub) regular expression A

Let R be a regular expression...

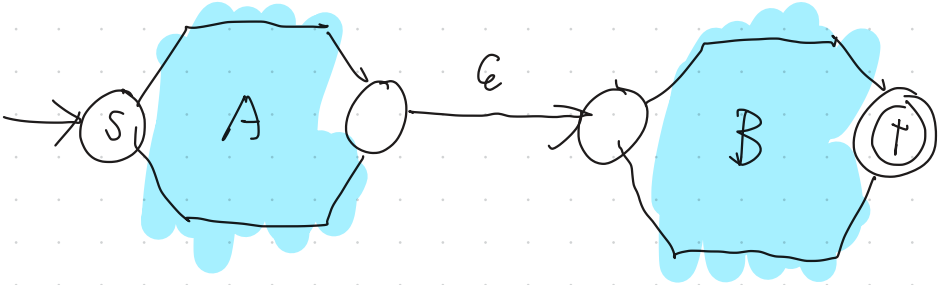
If $R = \emptyset$:

If $R = \varepsilon$:

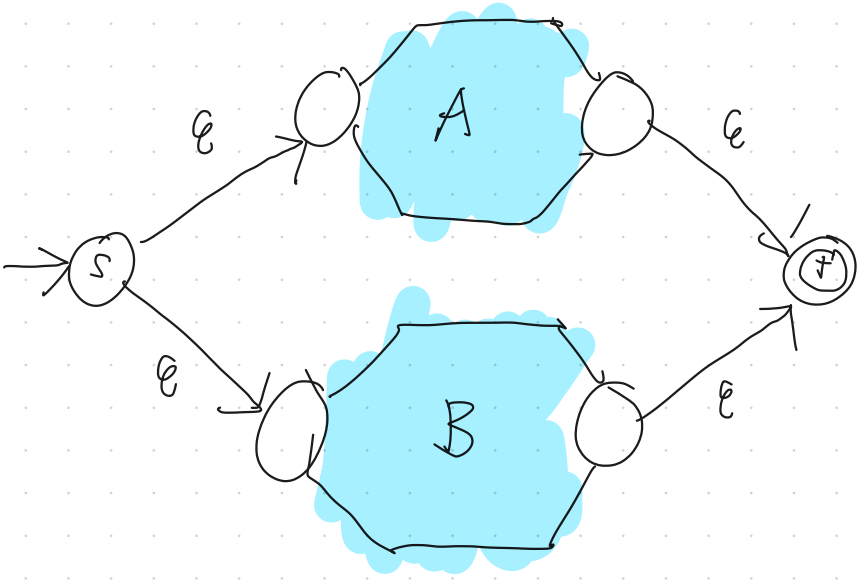
If $R = w = a_1 a_2 \dots a_n$



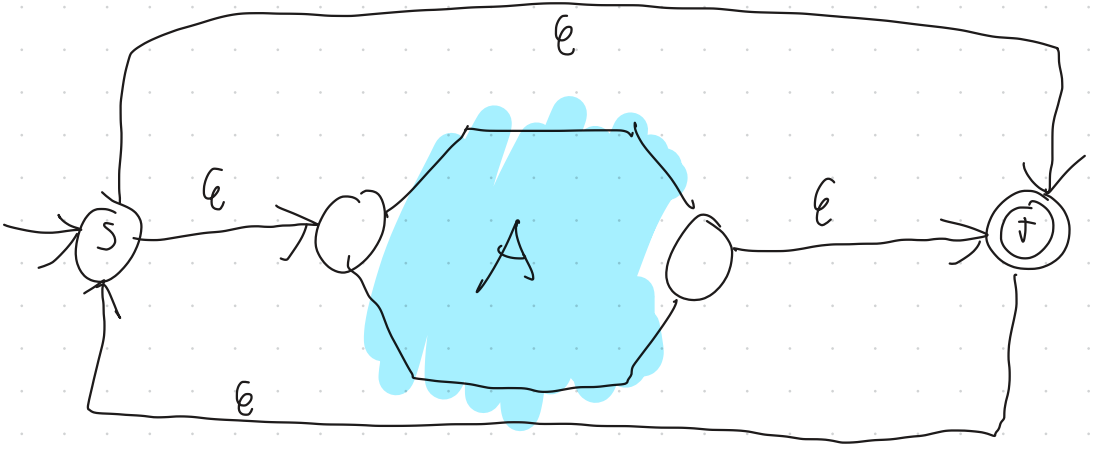
If $R = AB$:



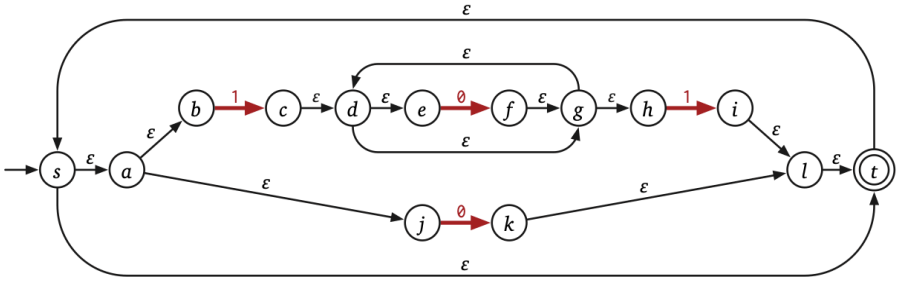
If $R = A + B$:



If $R = A^*$



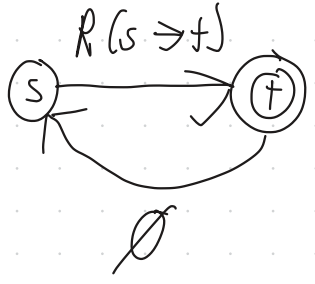
$(1 0^* 1 + 0)^*$ ⇒



NFA ⇒ regular expression...

expression automaton: finite-state machine where every $p \to q$ has a regular expression $R(p \to q)$ (can be $\emptyset$)

can go from p to q if $w = xy$ and
x matches $R(p \Rightarrow q)$. Continue with y.

If 

$\Rightarrow$ $R(s \Rightarrow t)$

Assume our machine has $s \neq t$ and

$R(q \Rightarrow s) = R(t \Rightarrow q) = \emptyset$

for all q

Pick $q \in Q \setminus \{s, t\}$.

Make a new "transition function"

$$R'(p \to r) := R(p \to r) + R(p \to q) R(q \to q)^* R(q \to r)$$