Announcements:

- no class 2/22
- gradescope checkin
- prob pres grades

Today:

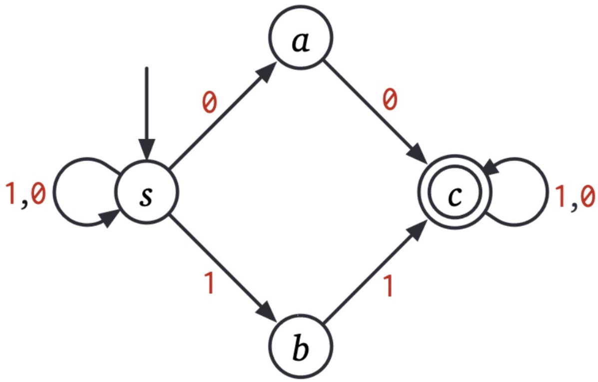
goal 1: prove Kleene's Theorem

regular <=> automatic

can be generated by a regular expression

can be recognized by a DFA

goal 2: language transformations



what's wrong w/ this DFA?

it's a Nondeterministic Finite Automaton (NFA)

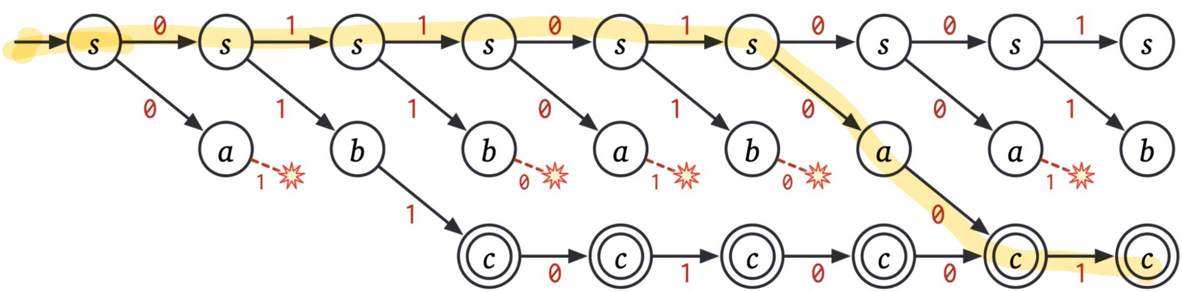
01101001

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

we say an NFA accepts w if there is some accepting path for w.

ways we can conceptualize NFA:

- clairvoyance/magic
- parallel threads



Running our example NFA on the input string 01101001.

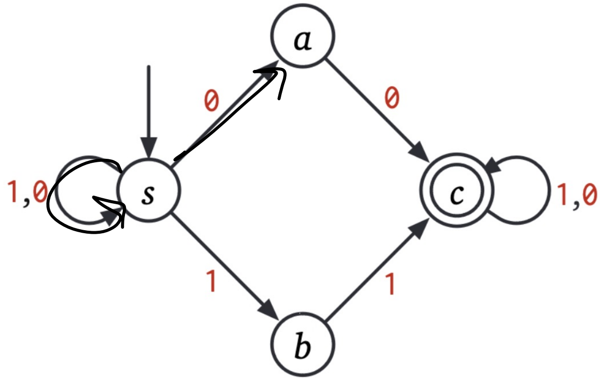
- verification / proof

Q: set of states
$\{s, a, b, c\}$

$s \in Q$: s

$A \subseteq Q$ $\{c\}$

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



$\downarrow$
the power set of Q

$P(Q) =$ the power set of Q

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

$= $ all subsets of Q

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

w/ partners:

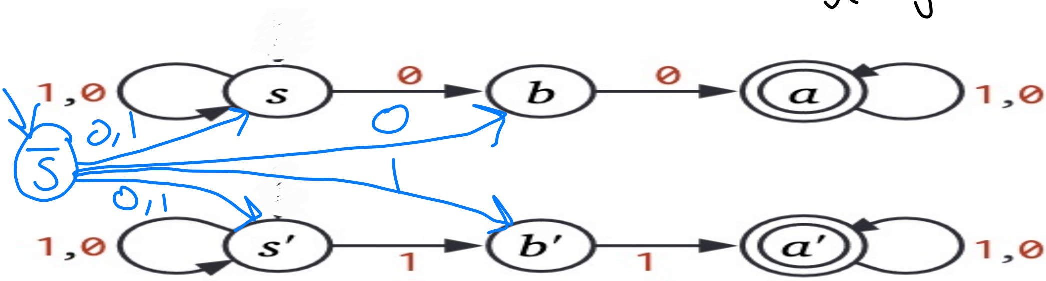
- what language does this NFA accept?

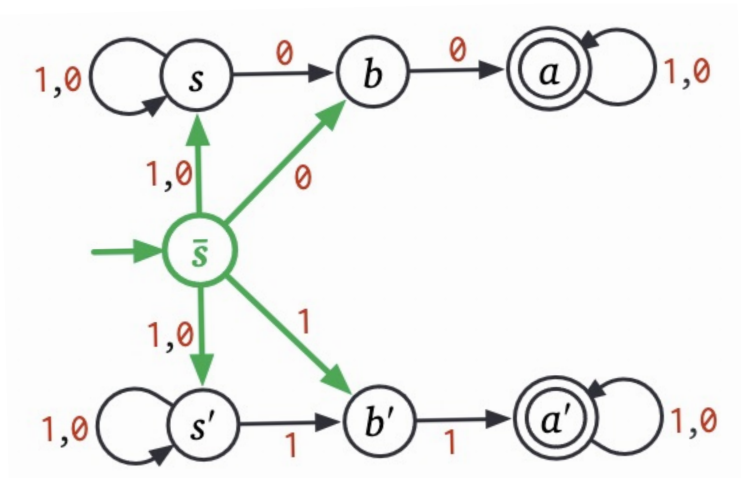
- note that the smallest DFA for this language has 8 states. How would you prove this fact?

More NFA options:

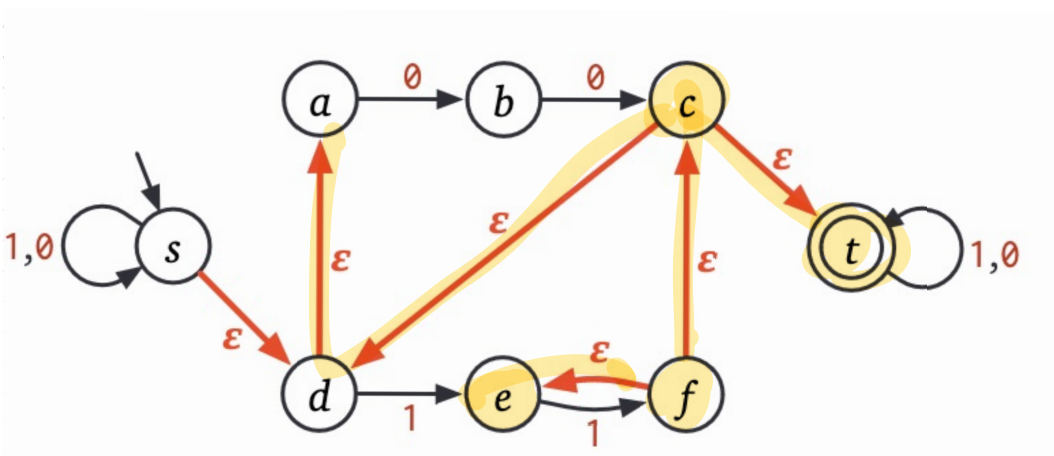
- multiple start states

single NFA





- ε-transitions



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

$\varepsilon\text{-reach}(f) = \{e, c, t, d, a\}$

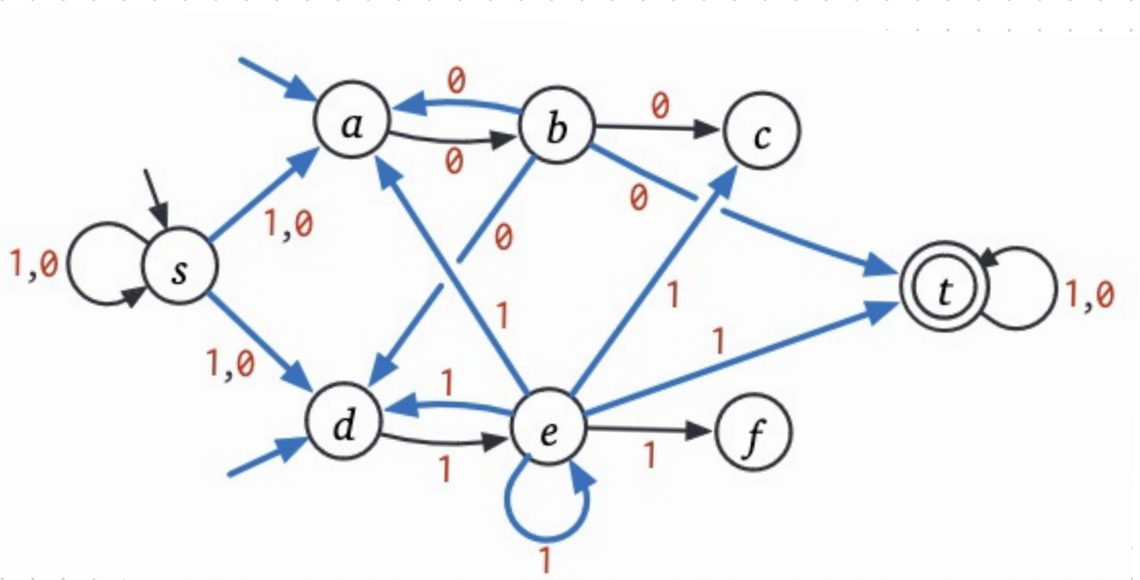
$$p \xrightarrow{0} q \xrightarrow{\varepsilon} r \quad \Rightarrow \quad \begin{tikzcd} p \arrow[r, "0"] \arrow[dr, "0"'] & q \\ & r \end{tikzcd}$$

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

$Q' = Q$

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

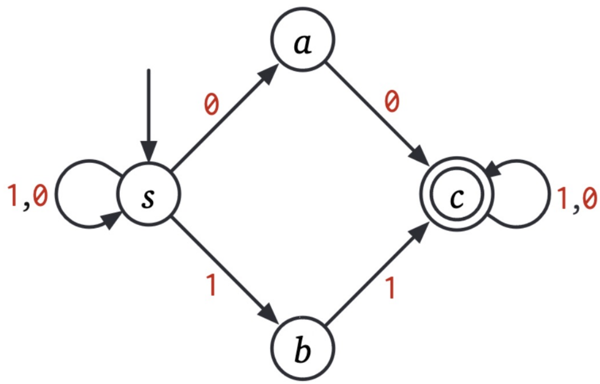
$\delta'(q, a) = \varepsilon\text{-reach}(\delta(q, a))$



regular $\Leftrightarrow$ automatic

$$\text{reg exp} \rightleftarrows \text{NFA} \xrightarrow{\text{subset const.}} \text{DFA}, \qquad \text{DFA} \xrightarrow{\text{trivial}} \text{NFA}$$

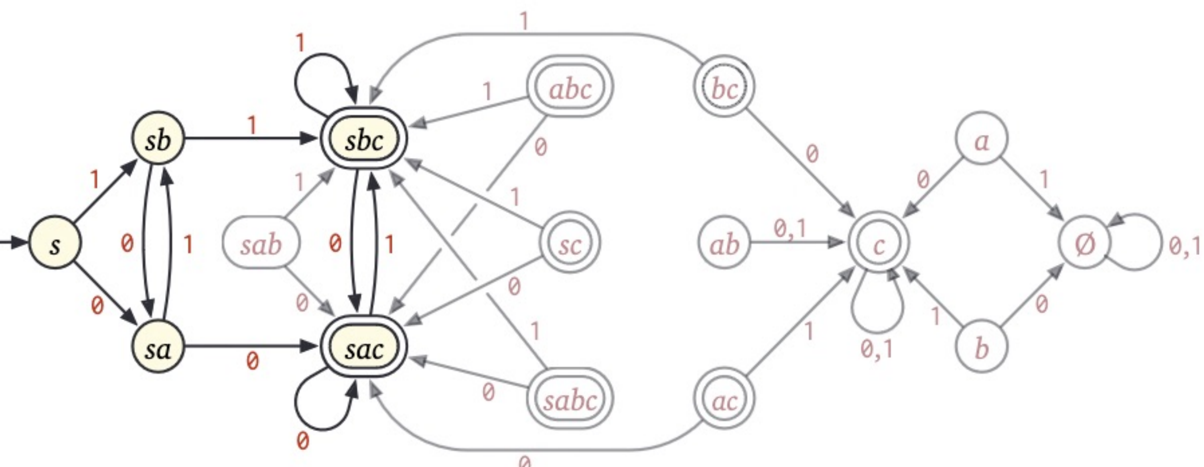
NFA $\Rightarrow$ DFA (subset construction)



w = 01101001

$$\left[\ \{s\} \xrightarrow{0} \{s,a\} \xrightarrow{1} \{s,b\} \xrightarrow{1} \{s,b,c\} \xrightarrow{0} \cdots\ \right]$$

- think of each set as a state of DFA



Given NFA $M = (Q, \Sigma, A, \delta)$, define DFA M' as follows:

$Q' = 2^Q$

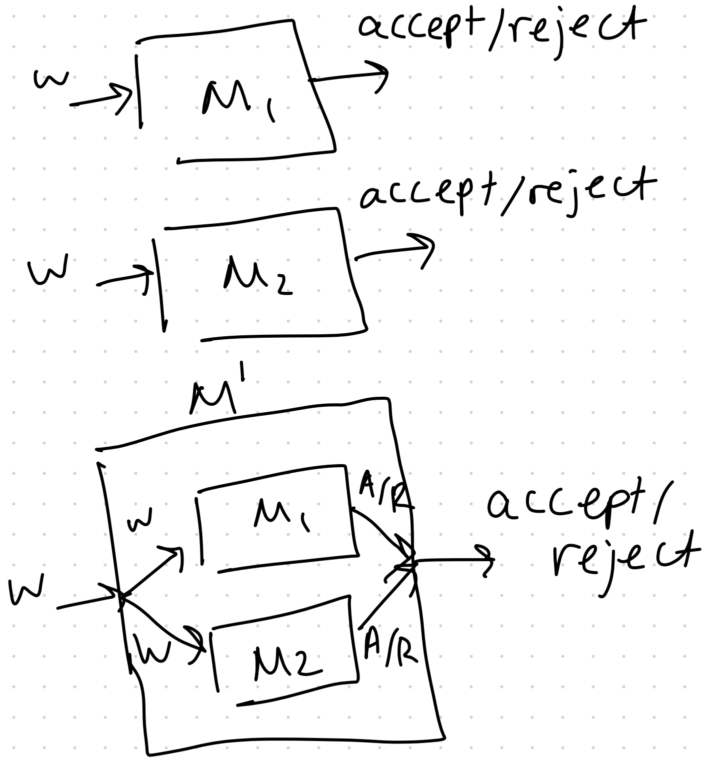
$s' = \{s\}$

$A' = \{q' \subseteq Q : q' \cap A \neq \emptyset\}$

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

Part 2: Language Transformations.

recall product construction:



$Q' = Q_1 \times Q_2$

$s' = (s_1, s_2)$

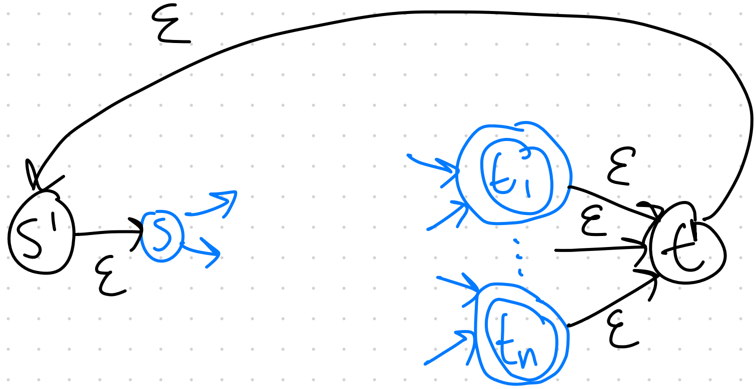
A = depends

$\delta'((p,q), a)$
$= (\delta_1(p,a), \delta_2(q,a))$

Suppose I have a DFA M accepting L. How do I build an NFA M' accepting L^*?

$L = \{ \varepsilon, 1, 101 \}$

$L^* = \{ \varepsilon, 11, 1101, 10110110111, \ldots \}$



$$Q' = Q \cup \{s', t'\}$$

$$s' = s'$$

$$A' = \{t'\}$$

$$\delta'(q,a) = \begin{cases} \{\delta(q,a)\} & \text{if } q \in Q \\ \{s\} & \text{if } q = s' \\ \{t'\} & \text{if } q \in A \end{cases}$$

let $$\text{flip}(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ 1 \cdot \text{flip}(x) & \text{if } w = 0x \\ 0\,\text{flip}(x) & \text{if } w = 1x \end{cases}$$

$$\text{flip}(1001) = 0110$$

let $\text{FLIP}(L) = \{\text{flip}(w) : w \in L\}$

$$\text{FLIP}(0^*1^*) = 1^*0^*$$

Claim: if L is regular, then $\text{FLIP}(L)$ is regular.

Proof:

Let M be a DFA accepting L.

relabel transitions.