Announcements

- Recitations start tomorrow
- Problem Set #1 out
- Problem Set #0

Maximum:

1. raw total as before
2. top 25 out of 30 nonbonus probs, scale up to 300 points and add any bonus points after

---

NFA ← DFA

Thm: Every reg lang can be recognized by an NFA.

Proof: State diagram of a DFA is already an NFA.

If $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA,

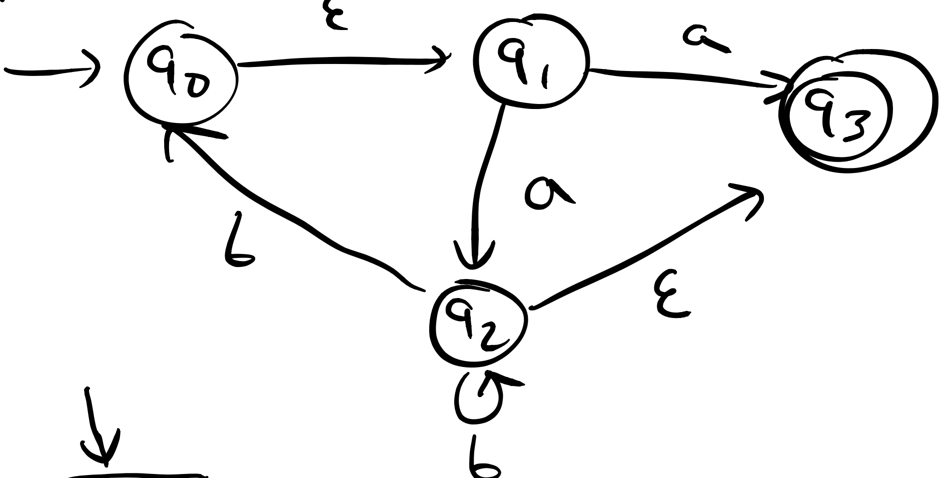
we make an NFA $N = (Q, \Sigma, \delta', q_0, F)$

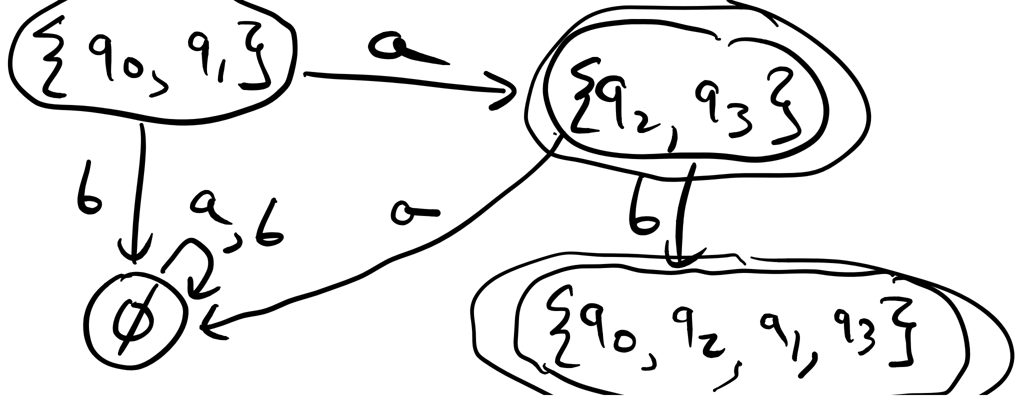
$$\delta'(q, a) = \underline{\{q'\}}$$

if $\delta(q, a) = q'$

So $L(M) = L(N)$. QED

Example:





Thm: Every language recognized by an NFA is regular.

(NFA $\to$ DFA)

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

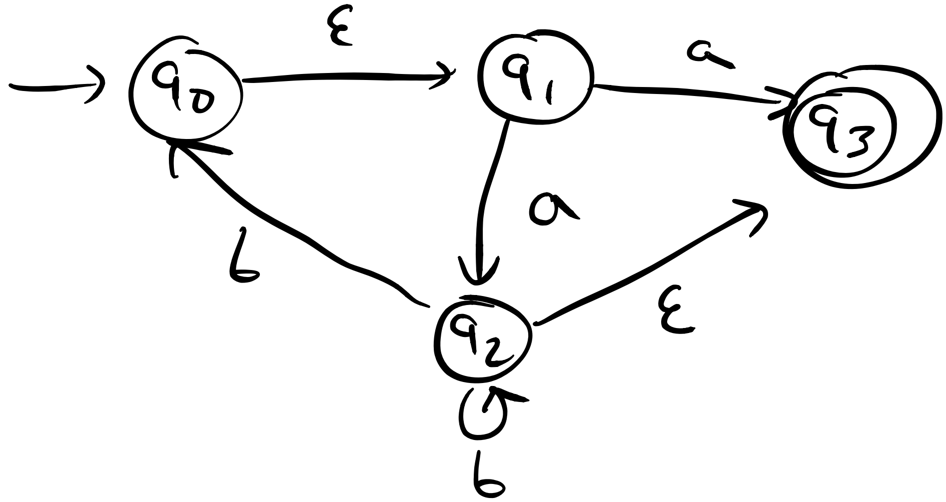
We construct a DFA $D = (Q', \Sigma, \delta', q_0', F')$ as follows:

(Powerset construction)

- $Q' = \mathcal{P}(Q)$

Def'n: The epsilon-closure (ε-closure) of a set of states S, $E(S)$, the smallest subset $X \subseteq Q$ s.t.

1. $S \subseteq X$
2. if $s \in X$ and $t \in \delta(s, \varepsilon)$ then $t \in X$.



$$E(\{q_0\}) = \{q_0, q_1\}$$

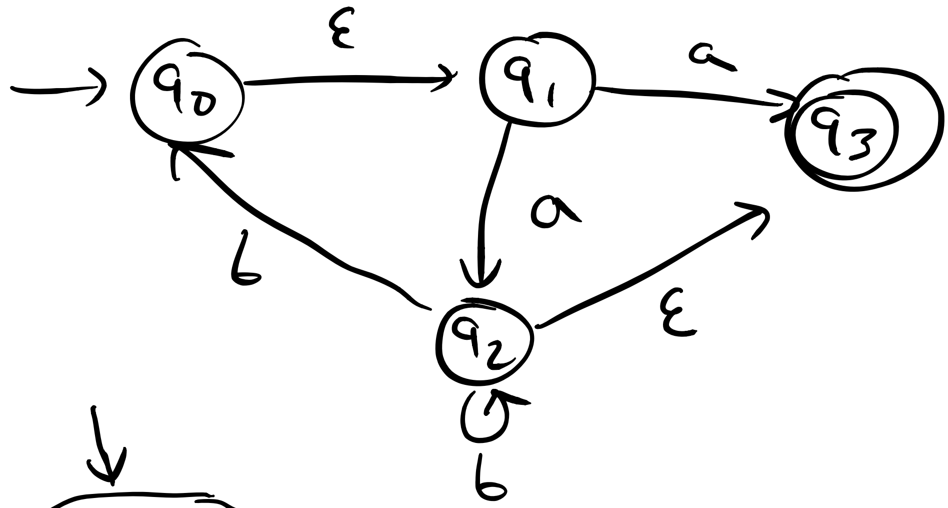
$$E(\{q_0, q_2\}) = \{q_0, q_2, q_1, q_3\}$$

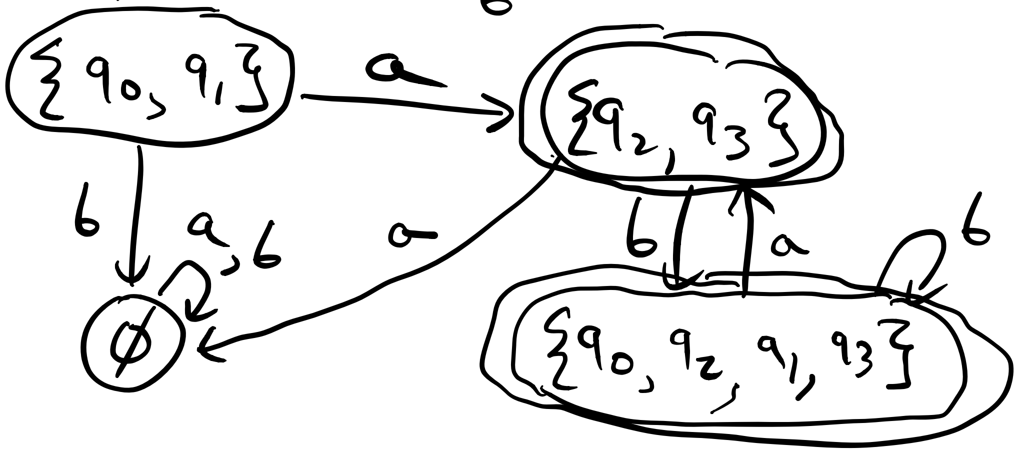
We construct a DFA $D = (Q', \Sigma, \delta', q_0', F')$ as follows:

(Powerset construction)

- $Q' = \mathcal{P}(Q)$
- $q_0' = E(\{q_0\})$
- $F' = \{X \subseteq Q : X \cap F \neq \emptyset\}$.
- $\delta'(S, a) = E\left(\bigcup_{s \in S} \delta(s, a)\right)$

for all $a \in \Sigma_\varepsilon$ and $S \in Q$





Starts with a, and does not have the substring aa, or is of the form ab^*

Regular Expressions

Let Σ be an alphabet.

Then we say R is a <u>regular expression</u> if:

1. $R = a$ for some $a \in \Sigma$
2. $R = \varepsilon$
3. $R = \emptyset$
4. $R = (R_1 \cup R_2)$ where R_1, R_2 are regexes
5. $R = (R_1)(R_2)$ ——————
6. $R = (R_1^*)$ where R_1 is a regex

Then define the language of a regex R:

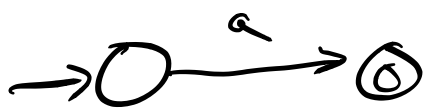
1. if $R = a$, then $L(R) = \{a\}$
2. if $R = \varepsilon$, then $L(R) = \{\varepsilon\}$
3. if $R = \emptyset$, then $L(R) = \emptyset$
4. if $R = R_1 \cup R_2$, then $L(R) = L(R_1) \cup L(R_2)$
5. if $R = R_1 R_2$, then $L(R) = L(R_1)\, L(R_2)$
6. if $R = R_1^*$, then $L(R) = (L(R_1))^*$

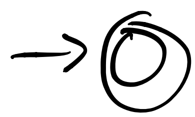
Example: a regex for all strings in $\{0,1\}^*$ that have the substring 101

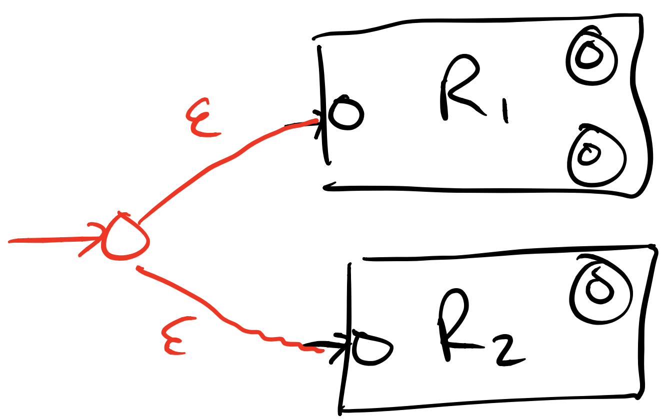
$$(0 \cup 1)^* 101 (0 \cup 1)^*$$

Thm The languages of regexes are the same as the regular languages

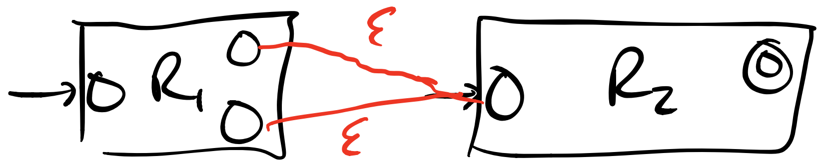
(Regex $\to$ NFA)

1. if $R = a$ 

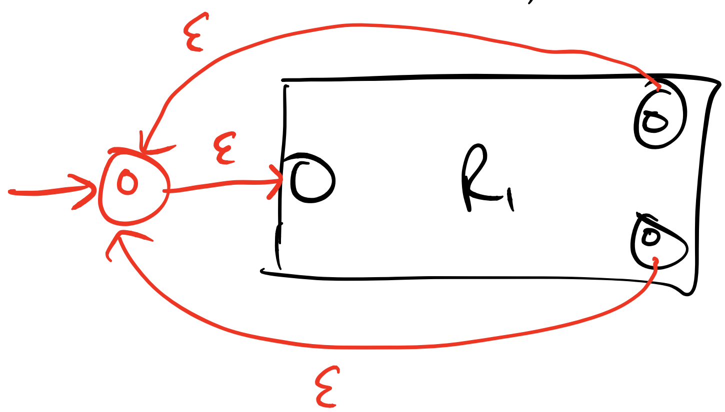
2. if $R = \varepsilon$ 
3. if $R = \emptyset$ 
4. if $R = R_1 \cup R_2$



5. if $R = R_1 R_2$

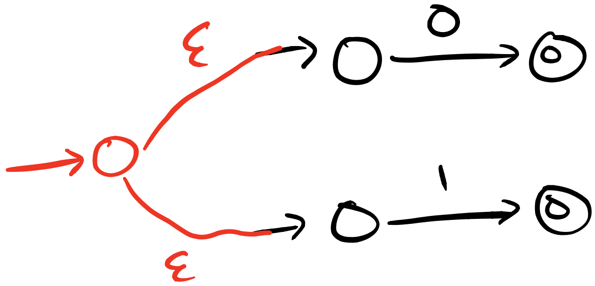


6. if $R = (R_1^*)$

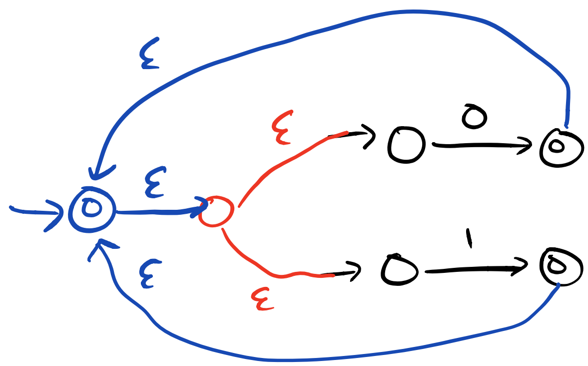


$(0 \cup 1)^* 101$ make an NFA

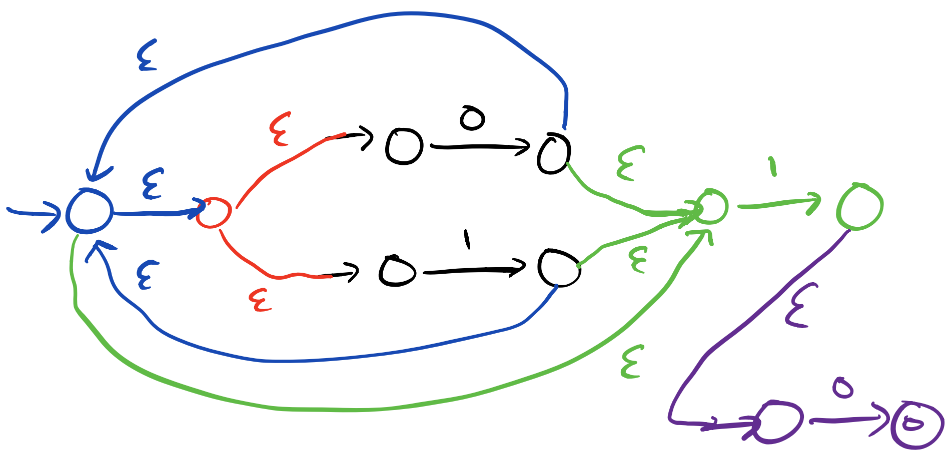
$0 \cup 1$:

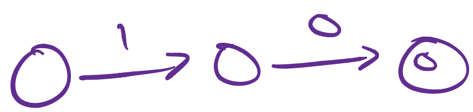


$(0 \cup 1)^*$



$(0 \cup 1)^* 1 0$



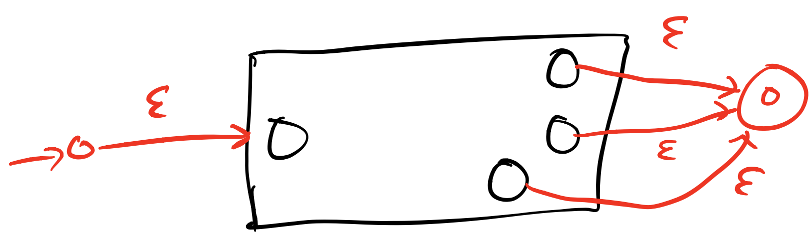


where R is a regex

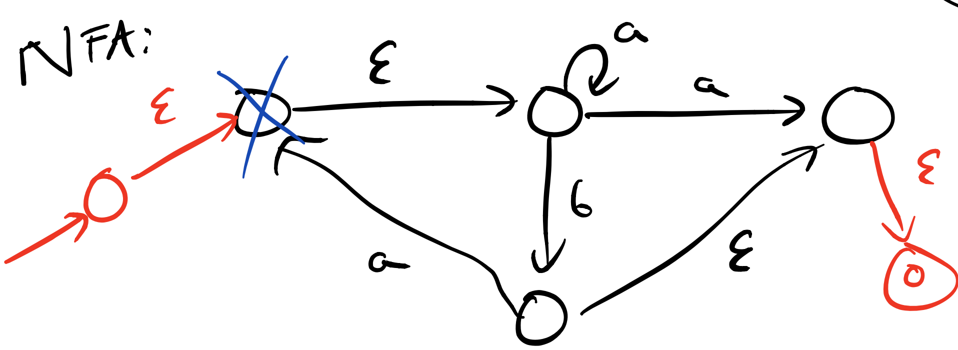
So the language of the NFA is $L(R)$.

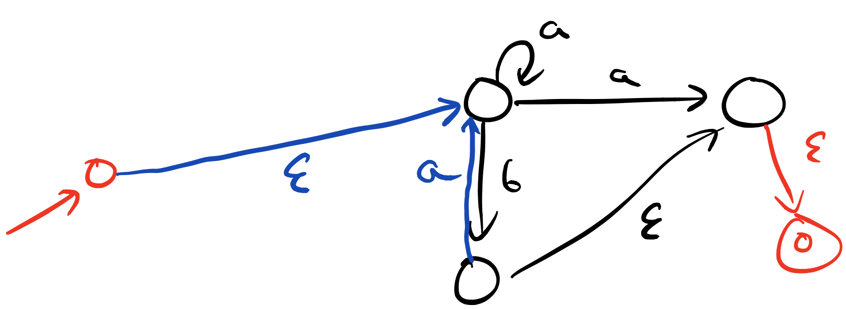
Def'n: A generalized NFA (GNFA) is an NFA but has a single start state, single final state, has no transition leaving the final state, has no transition entering the start state, and every transition is a regex.

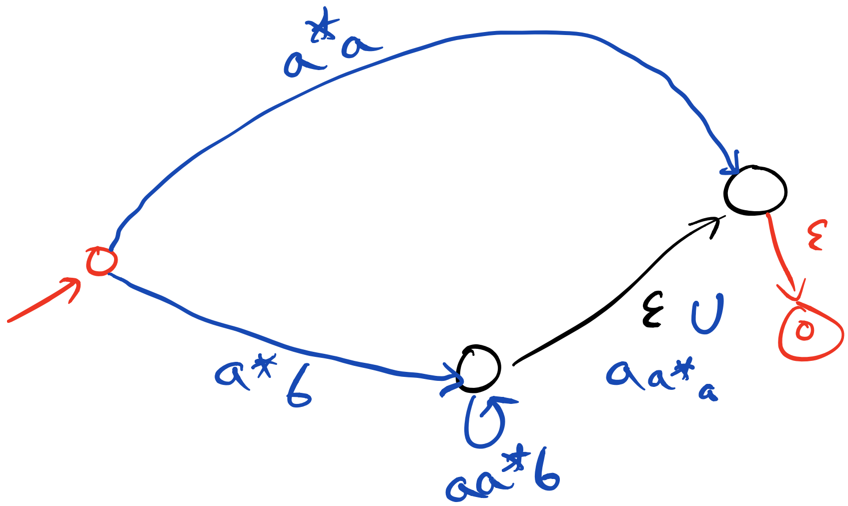
Q: Can every NFA be converted into one with a single final state?

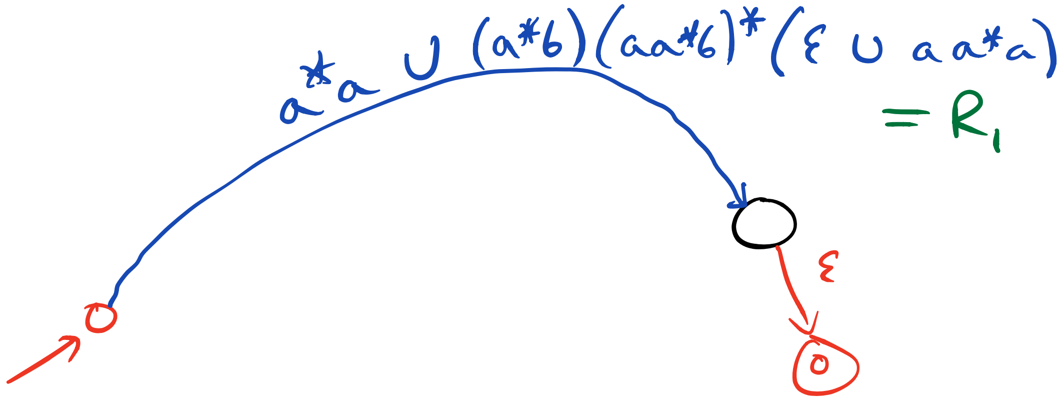


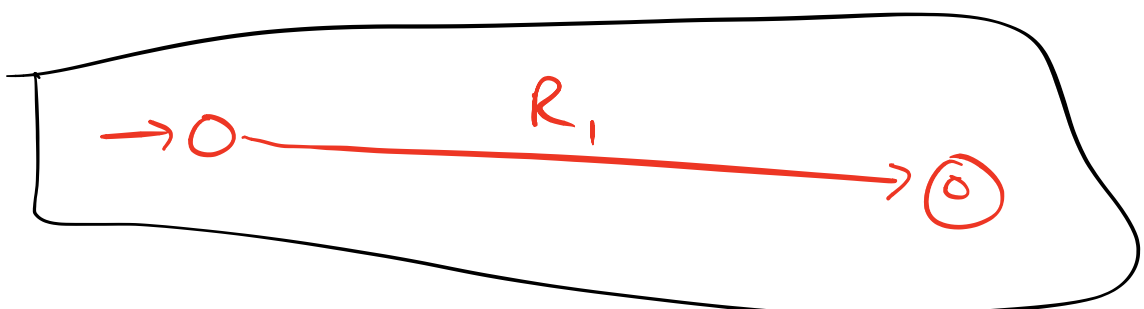
NFA:











The regex is R_1.

Problem Solving Session

Let $L \subseteq \{0,1\}^*$ be defined as follows:

- $10 \in L$ $\quad \#_0(10) = \#_1(10) = 1$
- for any $x \in L$, $x01x \in L$.

Prove that every $x \in L$ has equal number of 0's and 1's.

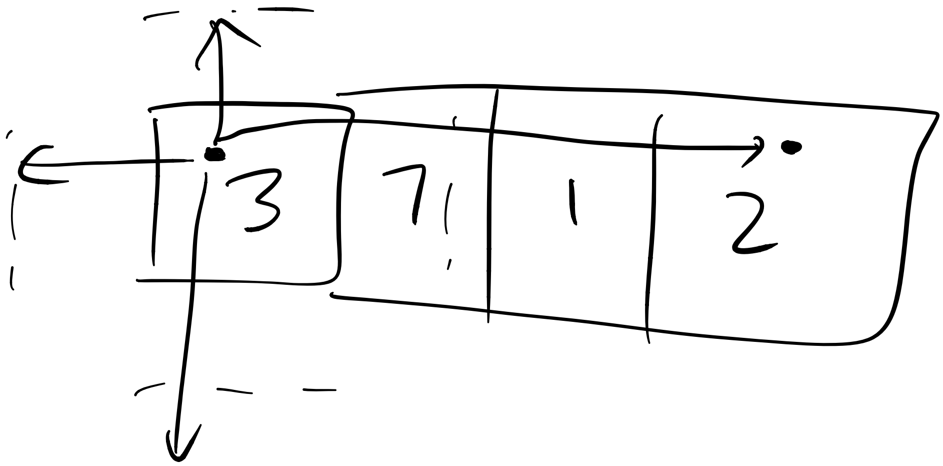
(Formally)
Define $\#_0(w)$ and $\#_1(w)$:

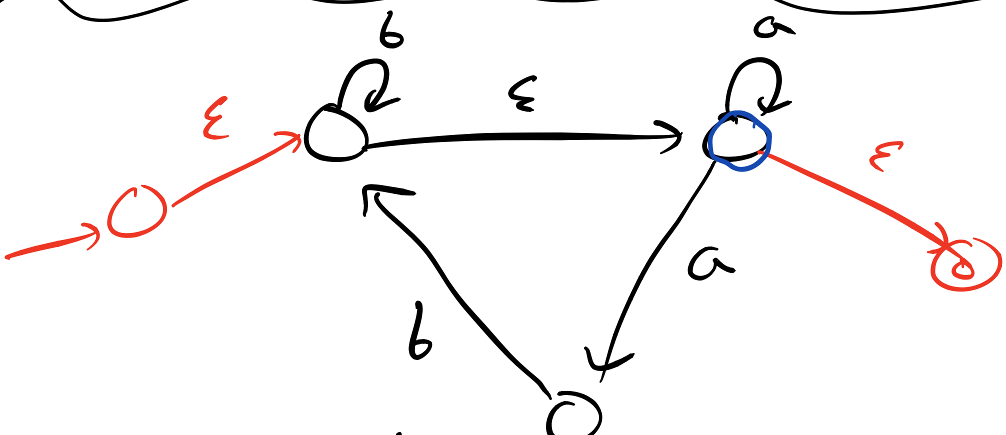
$$\#_0(w) = \begin{cases} 0 & \text{if } w = \varepsilon \\ 1 + \#_0(x) & \text{if } w = 0x \text{ and } x \in \{0,1\}^* \\ \#_0(x) & \text{if } w = 1x \text{ and } x \in \{0,1\}^* \end{cases}$$

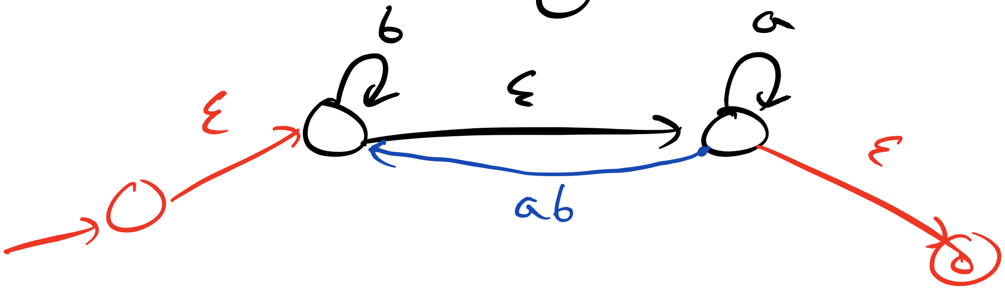
Prove: if $w = xy$
then $\#_0(w) = \#_0(x) + \#_0(y)$

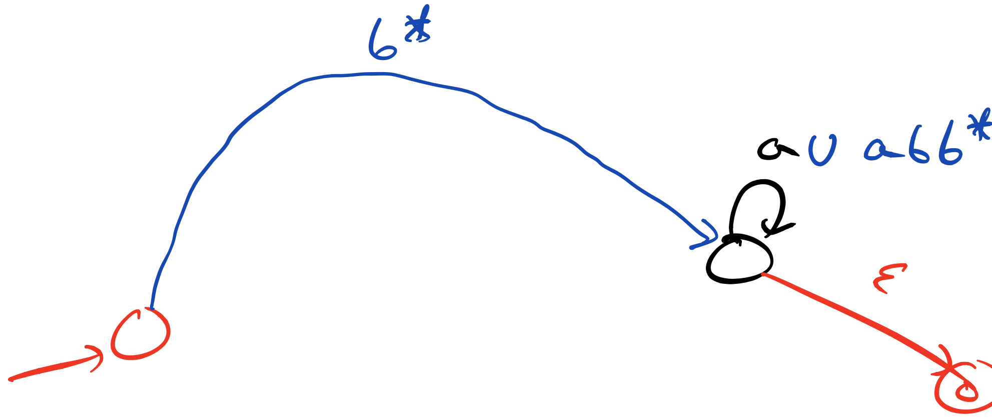
Assume $x \in L$ has $\#_0(x) = \#_1(x)$
then does $x01x$ have $\#_0(x01x) = \#_1(x01x)$?

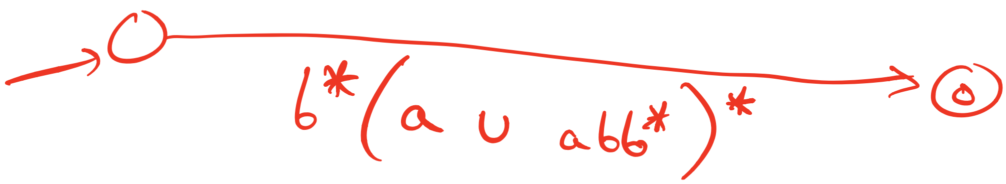
Jeff Erickson











Regex is $b^*(a \cup abb^*)^*$