<u>Kleene's Theorem</u> A language can be described by an FSA iff it can be described by a reg. expression.

$$\begin{tikzcd} \text{Any regex } \alpha \arrow[r, "\text{converted}"] & \text{NFA } N \arrow[r, "\text{convert}"] & \text{FSA } M \arrow[ll, bend right, "\text{convert?}"'] \end{tikzcd}$$

$$L(\alpha) \quad = \quad L(N) \quad = \quad L(M)$$

note: each FSA is trivially an NFA.

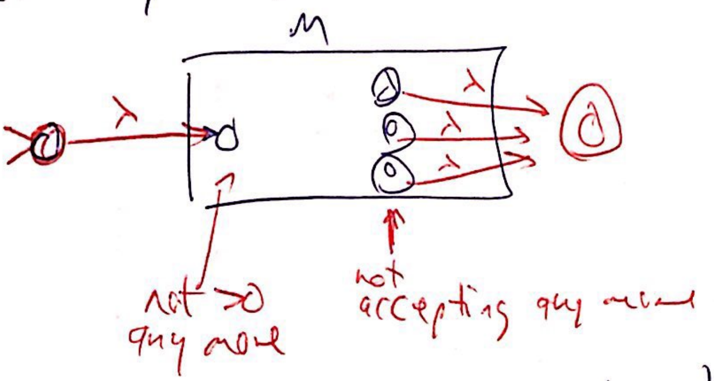
<u>idea:</u> generalize FSA to a regex eating automaton

where like: $\bigcirc \xrightarrow{a \cup b^*} \bigcirc$ any transition on such an arc consumes a string matching the regex from remaining input

goal: convert M to $\to\bigcirc \xrightarrow{\alpha} \circledcirc$ then $L(M) = L(\alpha)$

Convert M to $\to \bigcirc \xrightarrow{\alpha} \circledcirc$ by:

[1] Normalize start and accept states by adding one of each, removing the old ones:

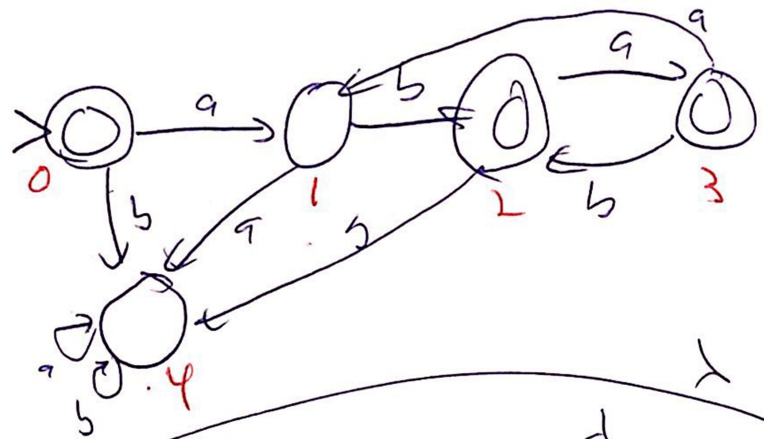


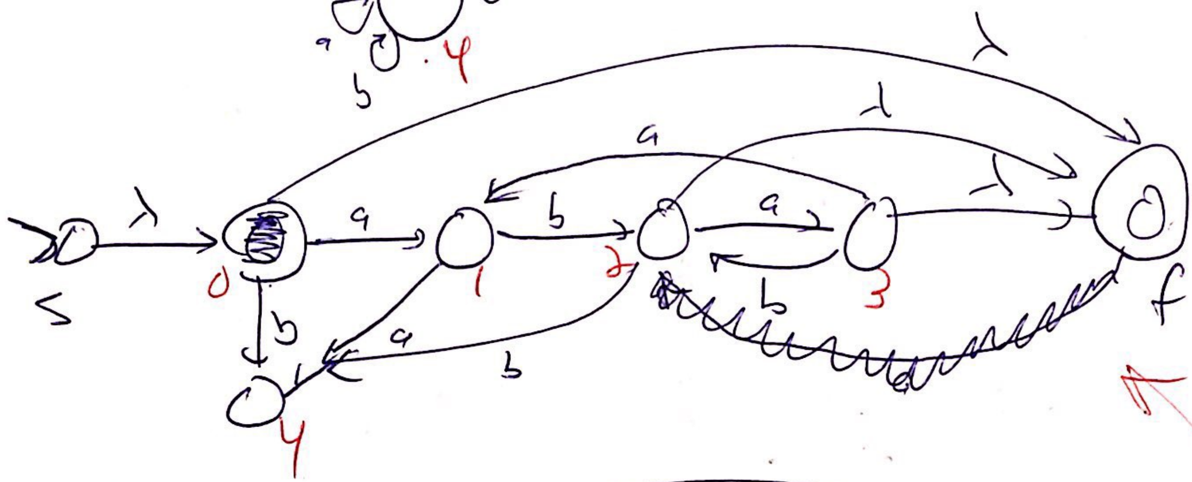
[2] Repeatedly delete ~~eac~~ each state except s, f.

a) copy everything else except deleted state d and its arcs.

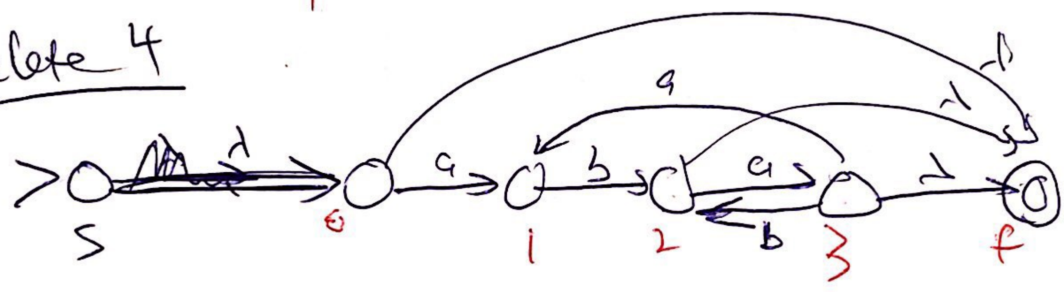
b) consider each pair (p,q) of remaining states and write a regex describing all strings that could travel $p \to d \to q$ before deletion.

M:





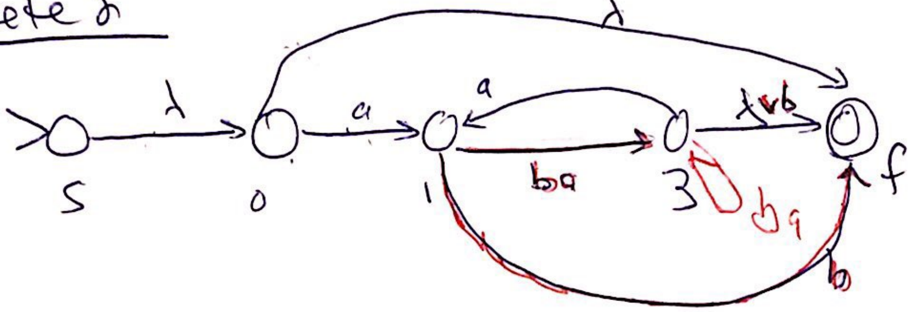
delete 4



No pair p, q can go p → 4 → q

So no new arcs

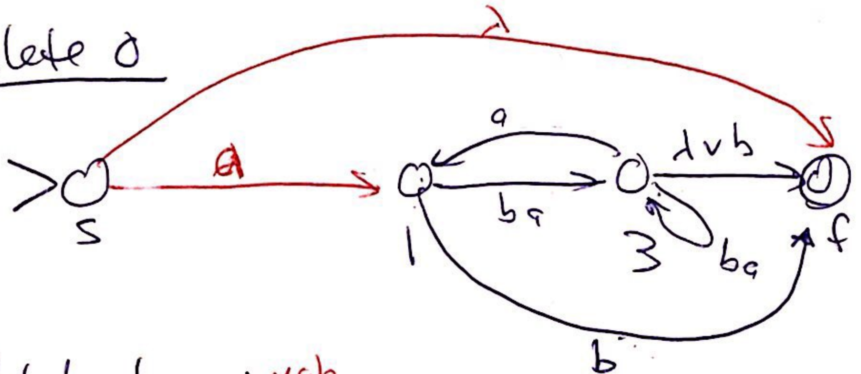
delete 2



pairs		regex
p→2	2→q	p→2→q
1	3	ba
1	f	b
3	3	ba
3	f	b

add to automaton (may use v)

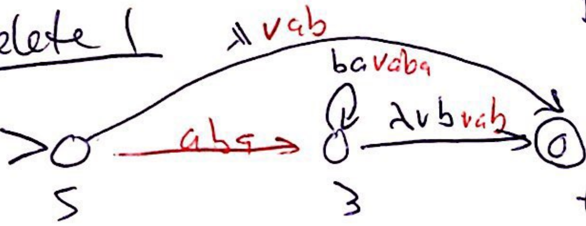
delete 0



pairs

p→0	0→q	regex
S	1	a
S	f	λ

delete 1



pairs

p→1	1→q	regex
S	3	aba
S	f	ab
3	3	aba
3	f	ab

<u>delete 3</u> (first case of deleting a state w/a self-loop) $p \to \overset{\circlearrowleft}{3} \to q$

$$> s \xrightarrow{\lambda \vee ab \vee aba(ba \vee aba)^*(\lambda \vee b \vee ab)} (f)$$

<u>paths</u>

s→3	3→f	regex
s	f	$aba(ba \vee aba)^*(\lambda \vee b \vee ab)$

$s \to 3 \quad \to f$

2)

self loop effect

$L(M) = L(\alpha)$ for $\alpha =$

$$\lambda \vee ab \vee aba(ba \vee aba)^*(\lambda \vee b \vee ab)$$

Leveraging the generality and correctness of the above process, any FSA M has regex α s.t $L(M) = L(\alpha)$

□ Kleene's Thm.

A language that can't be described by FSA or regex but is decidable:

$L = \{a^n b^n \mid n \in \mathbb{N}\}$ where a^n abbreviates $\underbrace{aaa \cdots a}_{n \text{ times}}$

$\Longrightarrow$ Large enough n will force any particular m into a repeated state, thus "forgetting" how many a's it has seen. So, no M has $L(M) = a^n b^n$