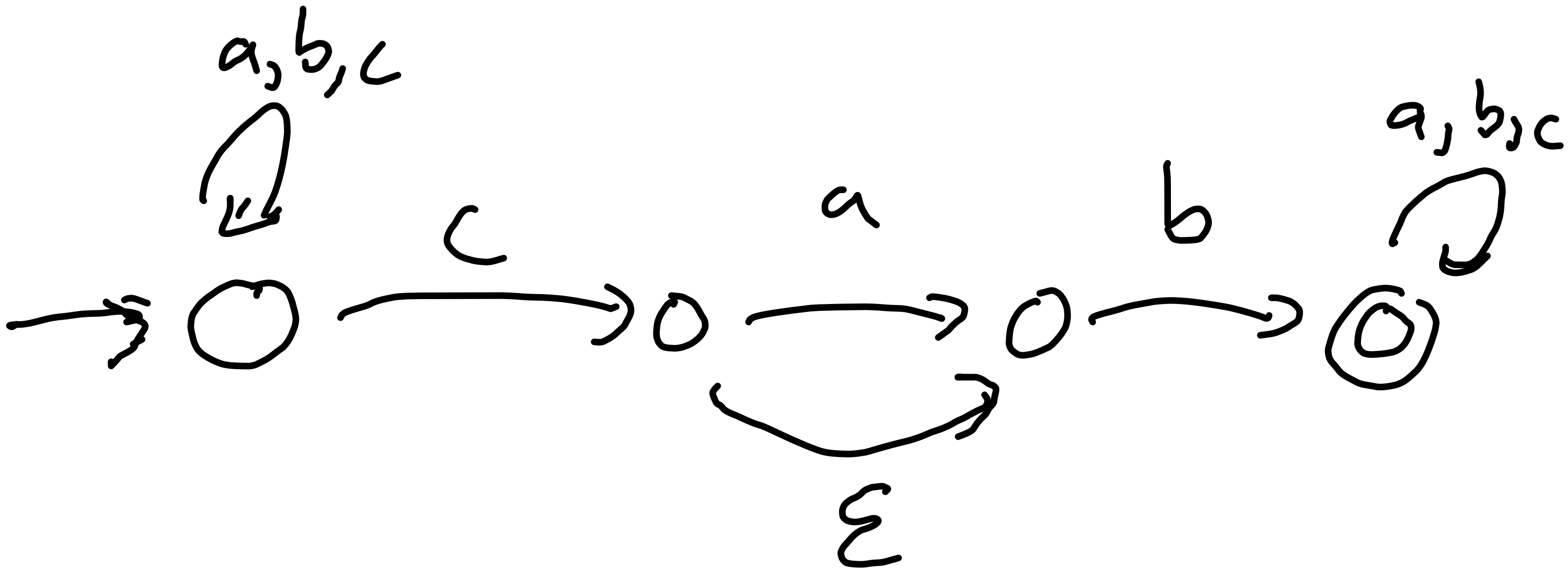
Non-deterministic finite automata (NFAs)

Like DFAs

2 extra features

- zero, multiple transitions for a (state, input) pair

- ε-transitions



$\{ \ldots \text{cab} \ldots, \ldots \text{cb} \ldots \}$

NFA a description of
possible moves / behaviors

To run:

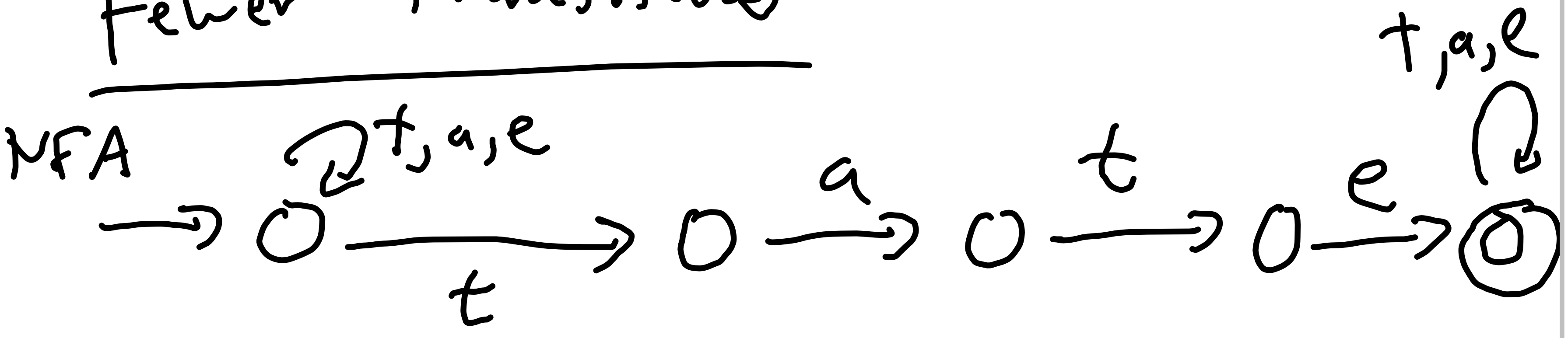
Compile NFA $\rightarrow$ DFA

A string w is accepted by an NFA M if There is some sequence of moves on input w That gets M to an accept state

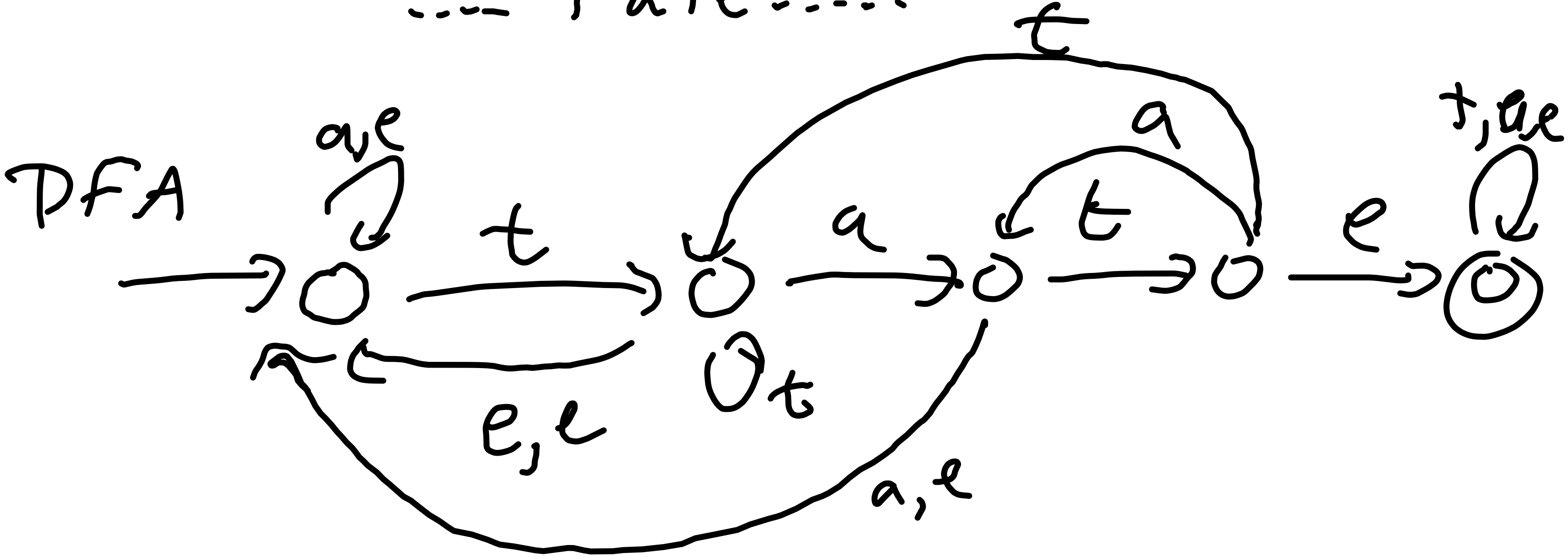
Nicer notation

- no need for reject transitions
- fewer states / fewer transitions
- easier to compose
- NFA's can guess parameters

Fewer Transitions

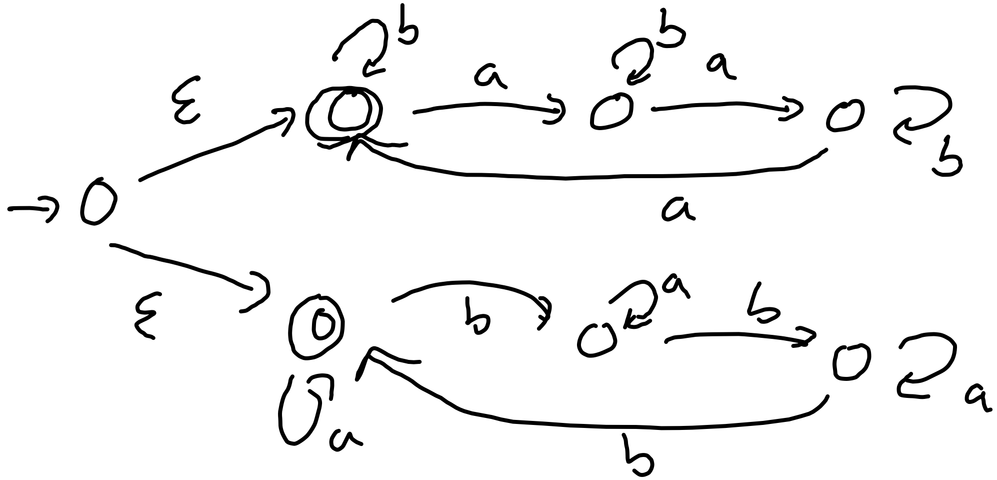


..... tate .....

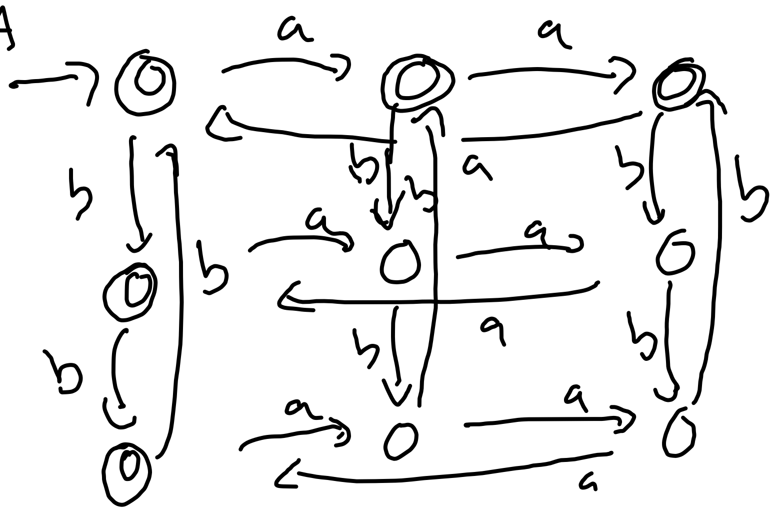


states exploding

$$\{w : w \text{ contains } 3n \text{ a's or } 3m \text{ b's}\} \subseteq \{a,b\}^*$$

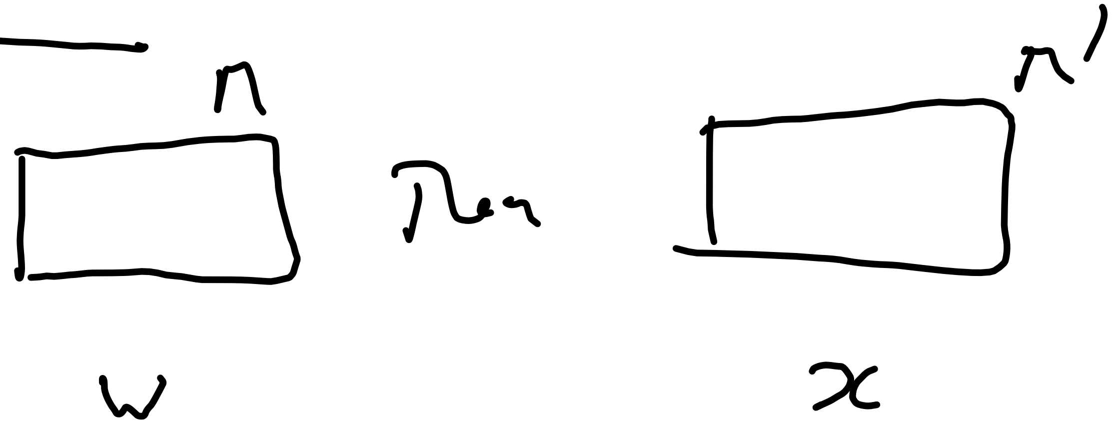


DFA



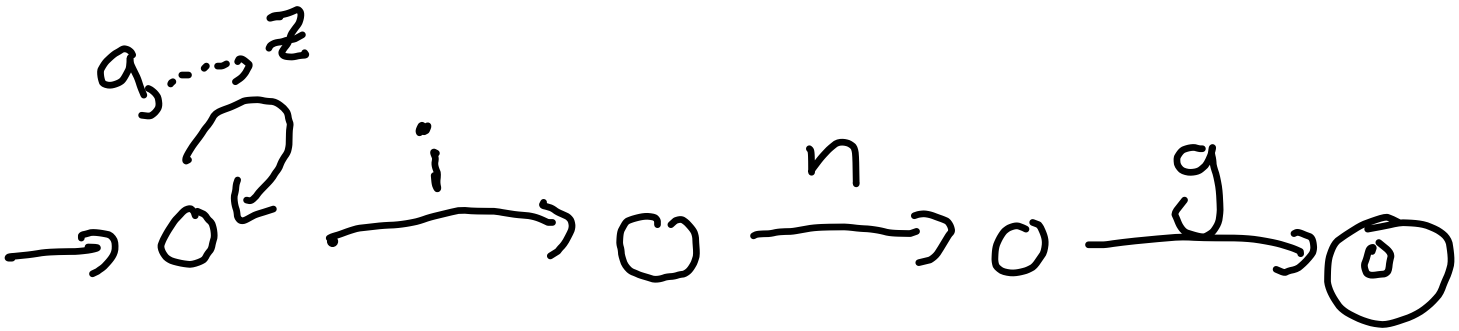
guessing

(1)

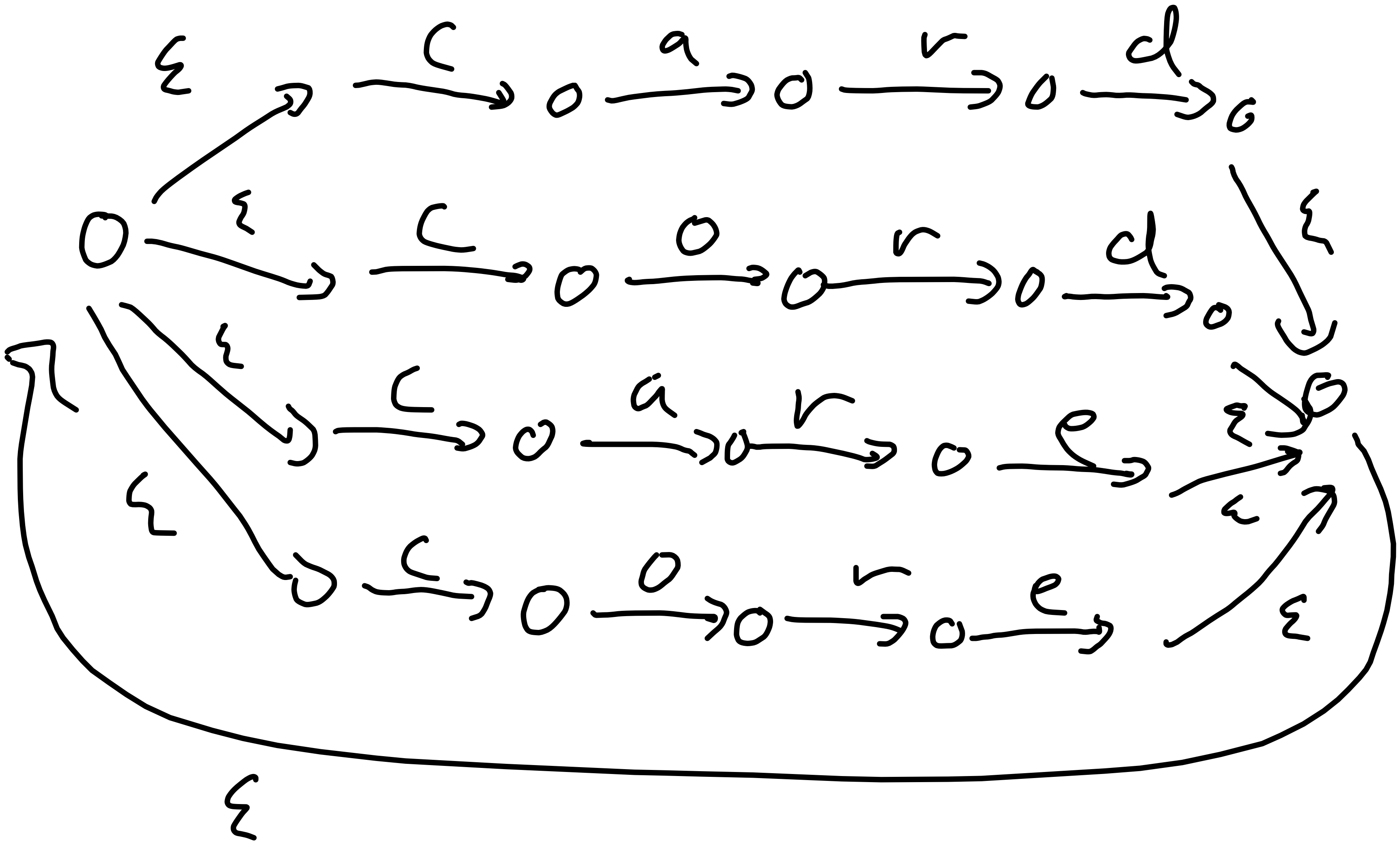


wx

(2)



card, cord, core, care ...



Notation

$$\Sigma_\varepsilon = \Sigma \cup \{\epsilon\}$$

$\mathcal{P}(A)$ = power set of A

= {subsets of A}

= 2^A

if $|A| = n$

$|\mathcal{P}(A)| = 2^n$

An NFA is a 5-tuple

$$\langle Q, \Sigma, \delta, q_0, F \rangle$$

where

- Q is a finite set of states
- Σ is a finite alphabet
- q_0 is the start state
- $F \subseteq Q$ are the accept states
- $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$

  i.e. output is a set of states

Let $M = \langle Q, \Sigma, \delta, q_0, F \rangle$
be an NFA
and let w be a string

Then M accepts w if
there exist

- a sequence of states
  $r_0, \ldots, r_m$

and

- a sequence of items
  from Σ_ε
  $w_1, \ldots, w_m$

Such That

- $w = w_1 w_2 \ldots w_m$

could be an input char in w

or could be an ε

- $r_0 = q_0$
- $r_m \in F$
- $r_{i+1} \in \delta(r_i, w_{i+1})$

$\forall i \in [0, m-1]$

$r_i \rightarrow r_{i+1}$ on input w_{i+1}

$$\begin{tikzcd} {} \arrow[r] & 1 \arrow[loop above, "{a,b}"] \arrow[r, "a"] & 2 \arrow[r, "b"] & 3 \arrow[d, bend right, "\varepsilon"'] \arrow[d, "b"] \\ & & & 4 \end{tikzcd}$$

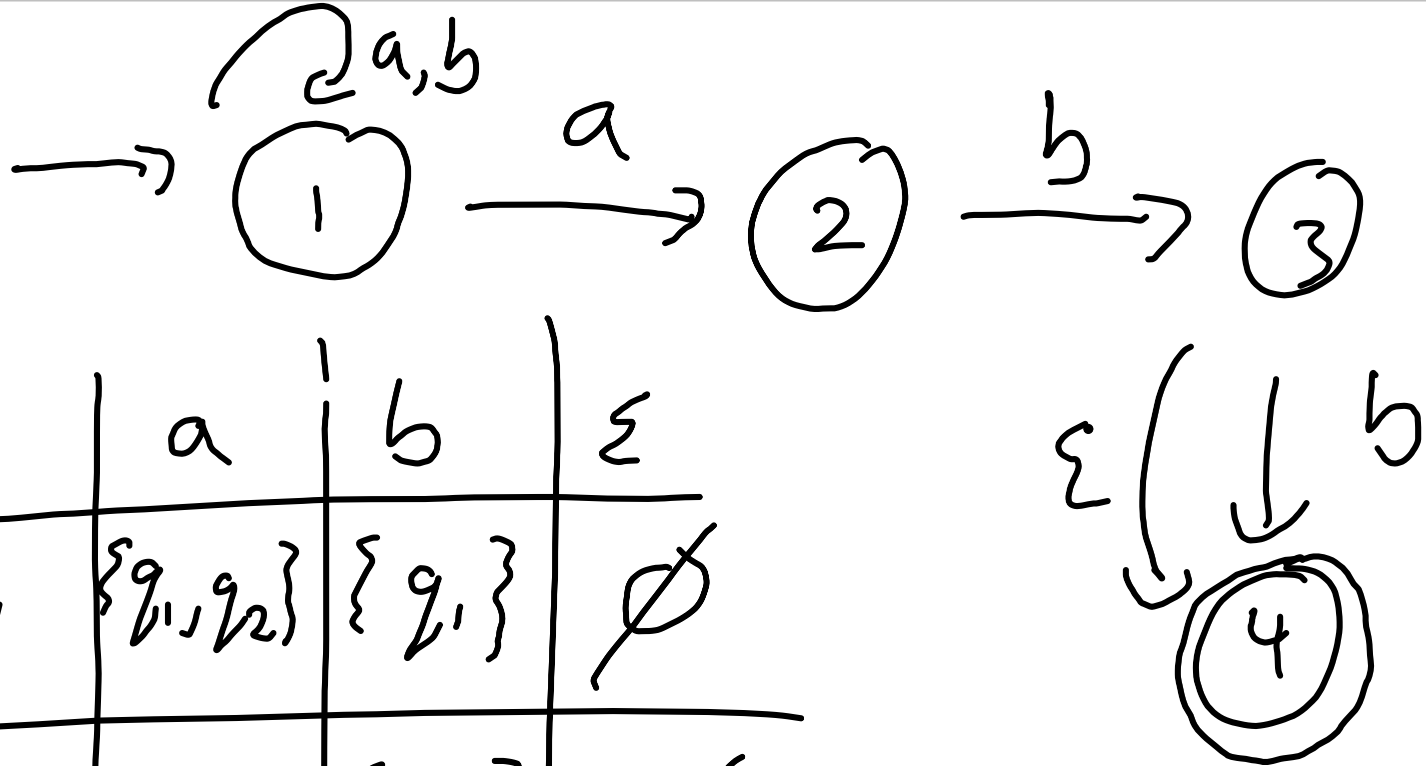
$Q = \{q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

q_1 is the start state

$F = \{q_4\}$

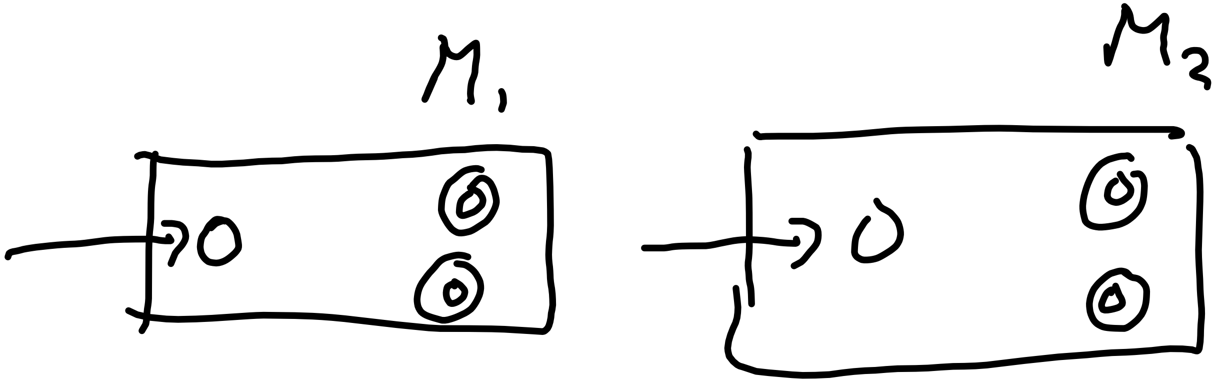
δ is defined by ....



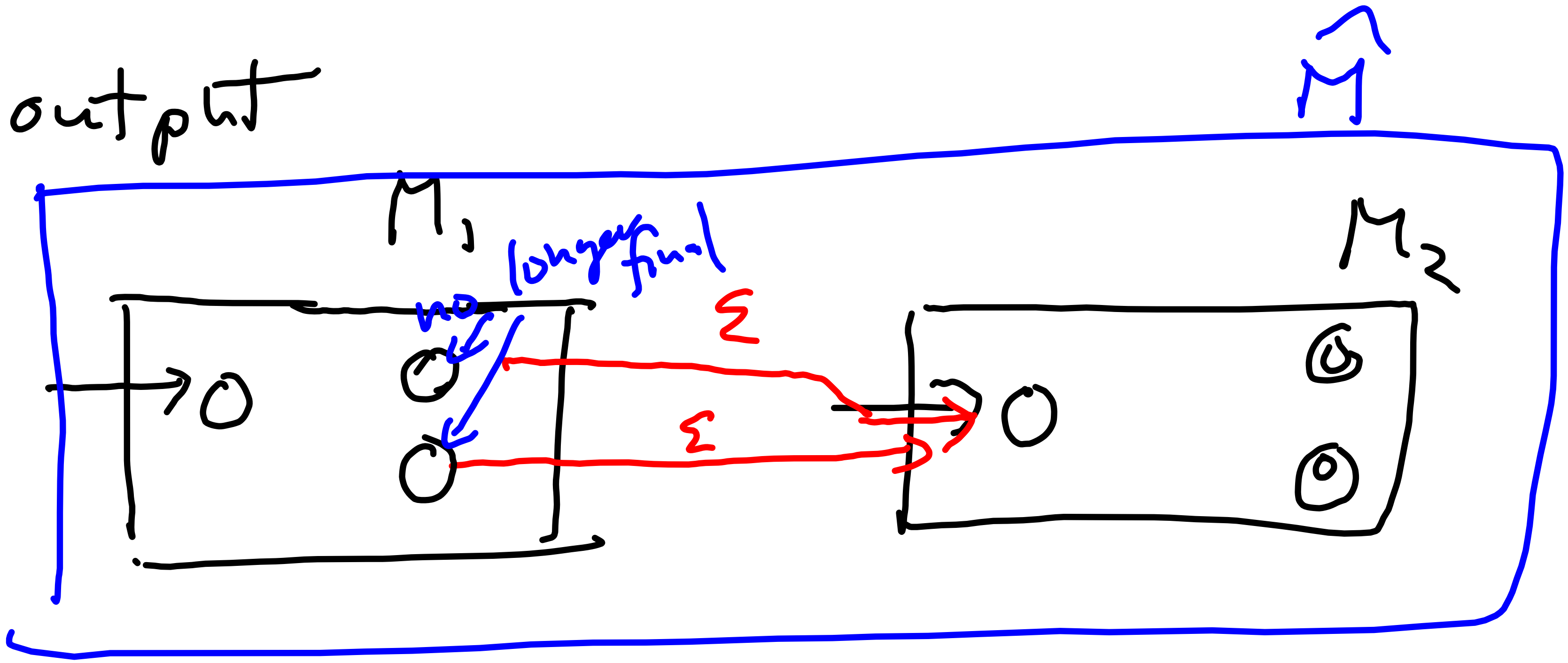
δ	a	b	ε
q_1	$\{q_1, q_2\}$	$\{q_1\}$	$\emptyset$
q_2	$\emptyset$	$\{q_3\}$	$\emptyset$
q_3	$\emptyset$	$\{q_4\}$	$\{q_4\}$
q_4	$\emptyset$	$\emptyset$	$\emptyset$

Closure under concatenation

inputs



output



Suppose we have 2 NFA's

$M_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$

$M_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$

recognizing regular languages

L_1 and L_2

Construct $\hat{M}$ recognizing

$L_1 L_2$

$$\hat{M} = \langle \hat{Q}, \Sigma, \hat{\delta}, \hat{q}, \hat{F} \rangle$$

$$\hat{Q} = Q_1 \cup Q_2$$

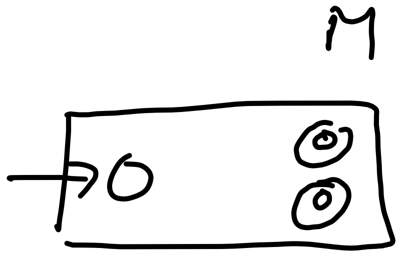
$$\hat{q} = q_1$$

$$F = F_2$$

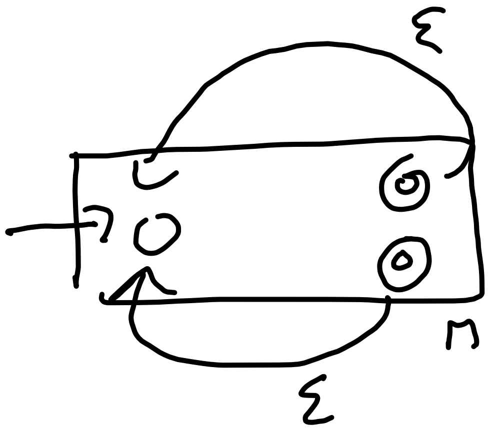
$$\hat{\delta}(q, a) = \begin{cases} \delta_2(q, a) & \text{if } q \in Q_2 \\ \delta_1(q, a) \cup \{q_2\} & \text{if } q \in F_1 \text{ and } a = \varepsilon \\ \delta_1(q, a) & \text{otherwise, i.e. } q \in Q_1 \text{ \& } (q \notin F_1 \text{ or } a \neq \varepsilon) \end{cases}$$

star

input



output ?



Might not accept input ε

extra state