Today Proving nonregularity via

1. Fooling sets

   formalises "memorylessness"

2. Closure properties.

   Leverage the fact that some other language is nonregular.

Challenging material ahead

Budget extra time
- Multiple passes
- Practice on many problems

Fooling Sets vs Pumping Lemma.

Fooling Sets Roadmap

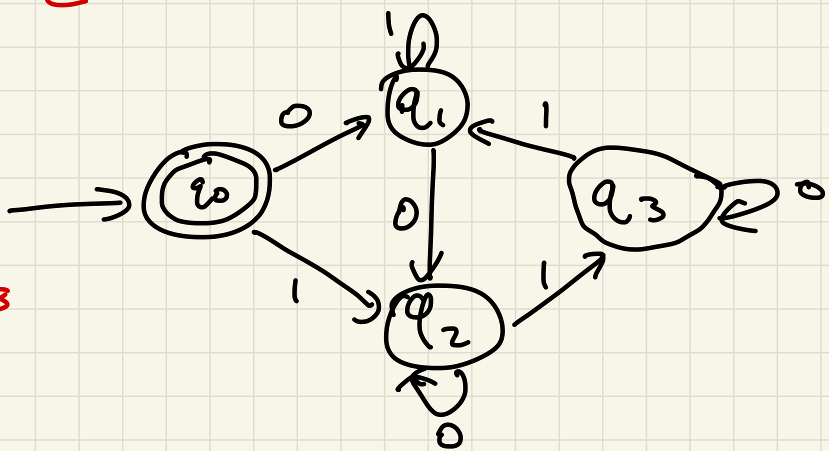
① Lower bounds on # states to recognize L

"no DFA w/ fewer than 5 states can recognize L"

② To prove non-regularity.

For every $k > 0$,

no DFA w/ $< k$ states can recognize L.



Def (Finite Automata)

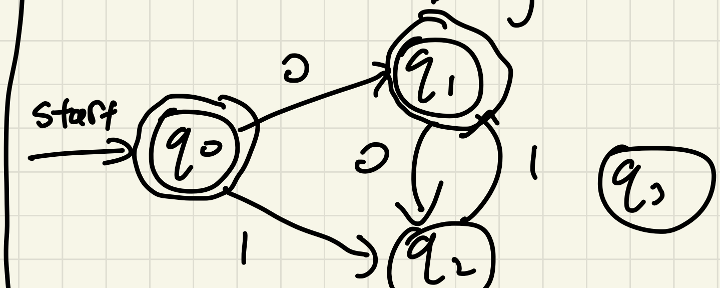
1. Alphabet Σ
2. Set of states Q
3. Initial state q_0
4. State transition function $\delta : Q \times \Sigma \to Q$

   δ(curr_state, next_char) = next_state.
5. Accept states F (subset of Q).

State Transition Table

		0	1
	q_0	q_1	q_2
(accept)	q_1	q_2	
	q_2		q_1
	q_3		

State Diagram

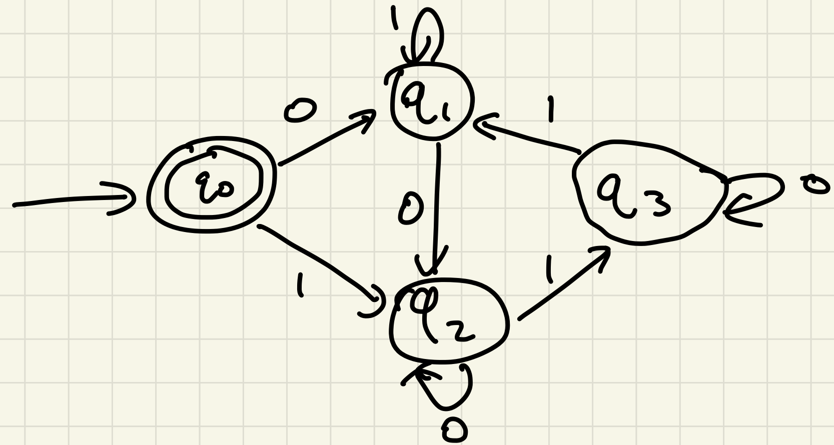


<u>**Def**</u> (Extended transition fn δ^*)

$$\delta^*(q, w) = \text{state after processing } w \text{ starting from state } q$$

q: curr state

w: remaining input string



Representation: $q \xrightarrow{w} r$

① Resulting state $q(w)$

$$q(w) = \delta^*(q_0, w)$$

② Prefix-suffix decomposition.

$0011 = 00\ 11$ $\quad 111 = 1\ 11$

$01011 = 010\ 11$

$$\underbrace{q_0 \xrightarrow{010} \overbrace{q(010)}^{r} \xrightarrow{11} q(01011)}$$

$$\delta^*(r, 11)$$

$w = xy$

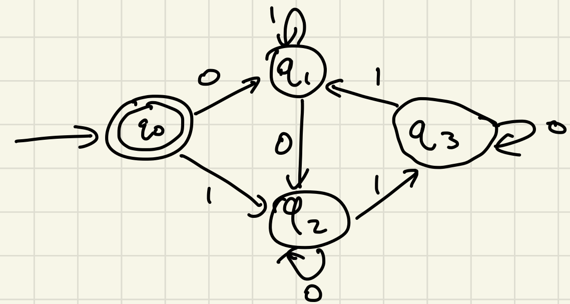
$$q_0 \xrightarrow{x} q(x) \xrightarrow{y} q(xy)$$

KEY TAKEAWAY

x y

same resulting state: $\begin{cases} 010\ 110\ 110 \\ 00\ 110\ 110 \\ 1\ 110\ 110 \end{cases}$

$\underbrace{\quad}_{\text{past}}$ present $\underbrace{\quad\quad\quad}_{\text{future}}$



Future transitions depend <u>only</u> on:

1. Present/current state
2. Future input

Lemma (Memorylessness)

For every string x, y,

if $q(x) = q(y)$ then

for every string z, $\underbrace{q(xz) = q(yz)}$.

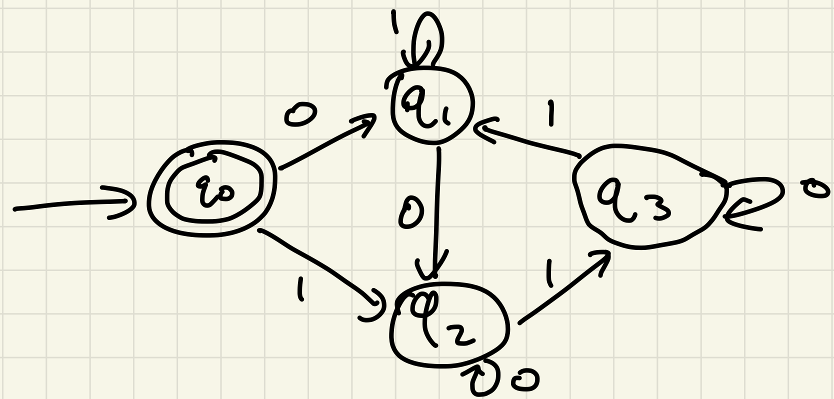
$\Downarrow$ implies

either M accepts xz and yz

or M rejects xz and yz

xz $\quad \overbrace{010}^{x}\,\overbrace{101}^{z}$

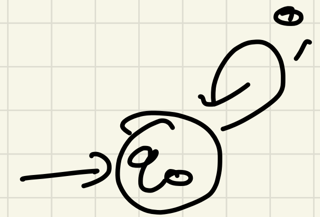
yz $\quad \underbrace{00}_{y}\,101$



Application of Memorylessness Lemma

Lemma (Characterization of languages of 1-state DFAs).

If M is a 1-state DFA, then M either accepts every string or rejects every string.



$$L(M) = \emptyset \quad \text{or} \quad L(M) = \Sigma^*$$

Pf Every string has the same resulting state.

Distinguishable pairs and distinguishing suffixes

Let L be a language.

Let x, y be strings (not necessarily in L)

Def x, y are distinguishable if

there exists z s.t. $xz \in L$ and $yz \notin L$

or $xz \notin L$ and $yz \in L$.

z: distinguishing suffix

Example L = set of even-length strings.

$= \{\varepsilon, 00, 01, 10, 11, 0000, \ldots\}$.

ε and 0 are distinguishable w/ suffix 0

$\varepsilon \cdot 0 = 0 \quad 0 \cdot 0 = 00$

$L =$ set of strings ending in 10

$x \quad y$

$1 \quad 0$

z

Suffix $\quad 0$

$10 \quad 00$

$xz \quad yz$