Announcements:

- Piazza
- Problem Set #0 due Wednesday at 11:59 PM
- Problem Set #1 out tomorrow
- No recitations tomorrow, start Thursday
- Midterm Exam #1 next Wednesday

DFAs
Regular languages

Operations on (Regular) Languages

Def'n: Let C be a set of languages. We say that C is <u>closed</u> under an operation $\oplus$ if $\oplus$ is applied to any language in C, the result is also in C.

Regular Operations:

- union: if L_1, L_2 are languages, then $L_1 \cup L_2 = \{x : x \in L_1 \text{ or } x \in L_2\}$

- concatenation:

  if w and x are strings where $w = w_1 w_2 \cdots w_n$ and $x = x_1 x_2 \cdots x_m$ then $wx = w_1 w_2 \cdots w_n x_1 x_2 \cdots x_m$ $(w \circ x)$

  if L_1, L_2 are languages, then $L_1 L_2 = \{wx : w \in L_1, x \in L_2\}$ $(L_1 \circ L_2)$

  $L_1 = \{0, 1\}$

  $L_2 = \{x, y, z\}$

  $L_1 L_2 = \{0x, 0y, 0z, 1x, 1y, 1z\}$

  In general, $L_1 L_2 \neq L_2 L_1$.

- (Kleene) star

  if L is a language, then $L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots$

More formally,

$$L^* = \bigcup_{k \geq 0} L^k$$

$$L^0 = \{\varepsilon\}$$

and $L^k = L L^{k-1}$ for $k \geq 1$

Ex: $L = \{0, 10\}$

$L^* = \{\varepsilon, 0, 10, 00, 010, 100, \ldots\}$

Complement

If Σ is the alphabet of language L, then

(L^c) $\overline{L} = \Sigma^* \setminus L$.

Theorem: The set of regular languages are closed under complement.

if L regular $\Rightarrow$ $\overline{L}$ is regular.

Proof: Since L is regular, there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ recognizing L.

$L(M) = \{ w \in \Sigma^* :$ M has an accepting computation on $w \}$.

$= \{ \_\_\_ :$ the only computation of M on w is accepting. $\}$.

How to represent all states in Q but not in F?

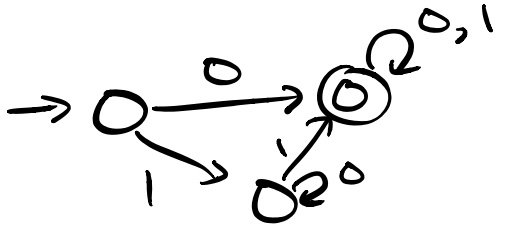
$Q \setminus F$ or $Q - F$

Make a DFA $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$

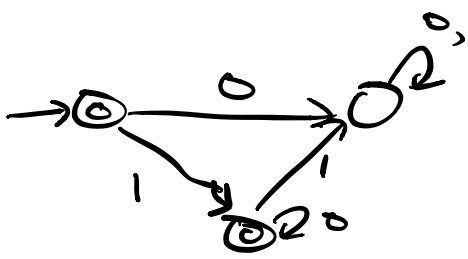
Claim: $L(\overline{M}) = \overline{L}$.

Proof: there is exactly 1 computation of M on w.

Ex:



Language L

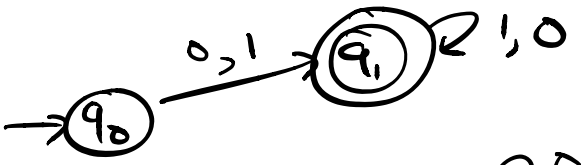


language $\overline{L}$

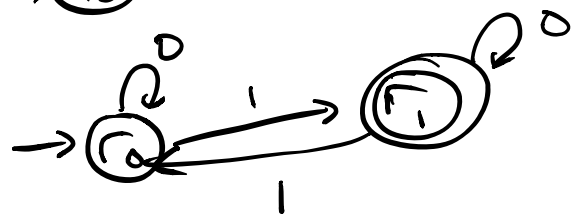
Union of RLs.

Question: if L_1, L_2 are reg. langs, is $L_1 \cup L_2$ regular?

M_1:



M_2:



	0	1	1	0
q_0	q_1	q_1	q_1	q_1 ← final
r_0	r_0	r_1	r_0	r_0

Theorem: Reg langs are closed under union.

Proof: Let L_1, L_2 be any regular languages.

$\Rightarrow$ there is a DFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ for L_1

——— $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ for L_2.

Construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$ as follows:

- $Q = Q_1 \times Q_2$
- $q_0 = (q_{01}, q_{02})$
- $\delta((q_1, q_2), a)$

  $= (\delta_1(q_1, a), \delta_2(q_2, a))$

"Product construction"

- $F = \{(q_1, q_2)$: either $q_1 \in F_1$ ~~and~~ $q_2 \in F_2$ ~~or both~~ $\}$

(for intersection)

Symmetric Difference:

$\{w : w \in L_1$ or $w \in L_2$ but not both$\}$

$F = \{(q_1, q_2) : q_1 \in F_1$ or $q_2 \in F_2$ but not both$\}$.

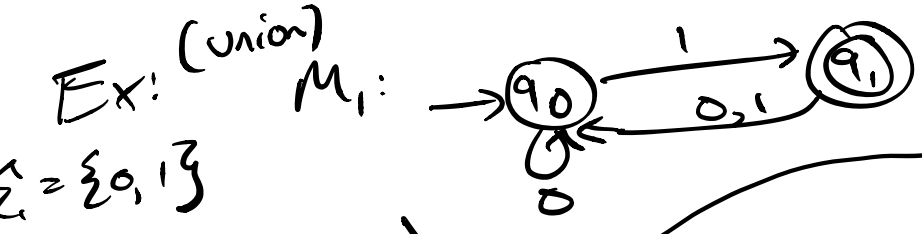
Set minus:

$L_2 - L_1 = L_2 \cap \overline{L_1}$

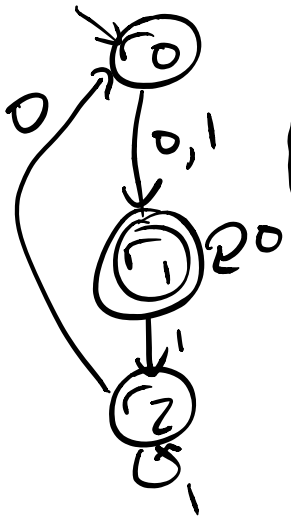
Ex: (union)

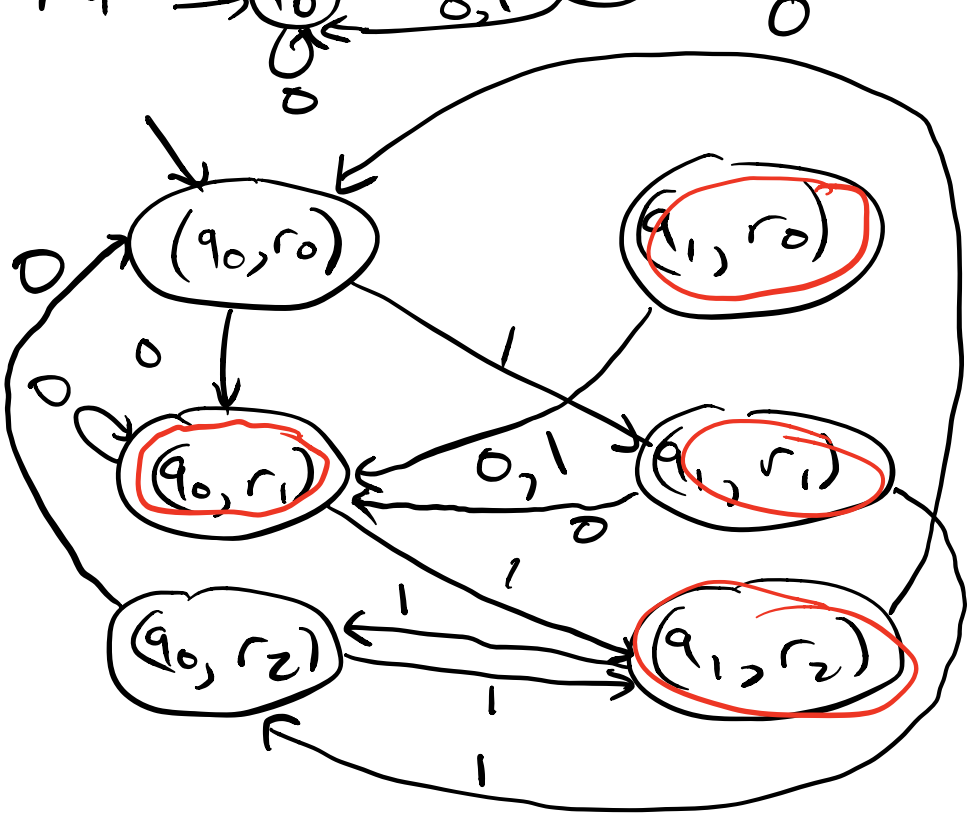
$\Sigma = \{0, 1\}$

M_1:



M_2:





Concatenation

$$L_1 L_2 = \{ w_1 w_2 : w_1 \in L_1, w_2 \in L_2 \}.$$

Q: if L_1, L_2 are regular, is $L_1 L_2$ reg?

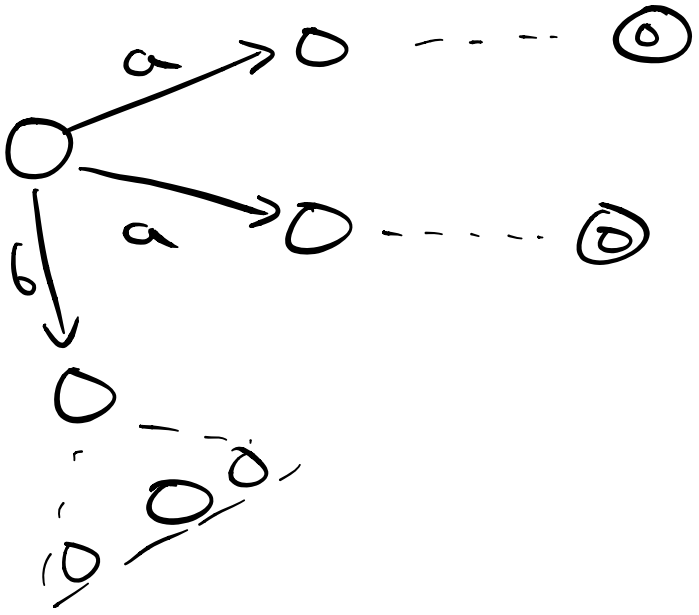
$w_1 w_2 \cdots w_{i-1}$ $\in L_1$; $w_i w_{i+1} \cdots w_n$ $\in L_2$

Nondeterministic Finite Automata

Ideas:

- ε-transitions: no input character is read when traversing the transition
- multiple transitions (incl. 0) on the same symbol

Informal def'n of choosing:

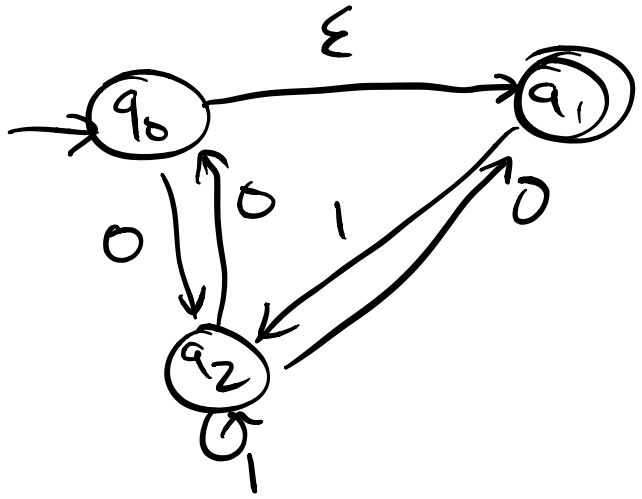


Formal def'n: An NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

- Q = finite set of states
- Σ = " alphabet

- $q_0 \in Q$ is the start state
- $F \subseteq Q$
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

Ex:



	1	0
q_0	q_1	q_2	q_0	q_1

✓

A computation of an NFA

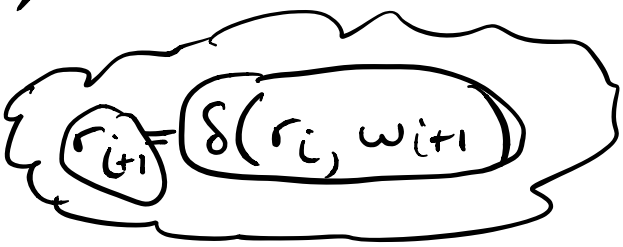
$M = (Q, \Sigma, \delta, q_0, F)$ on input $w \in \Sigma^*$

where $w = \underbrace{y_1 \cdots y_m}$ with

each $y_i \in \Sigma \cup \{\varepsilon\}$

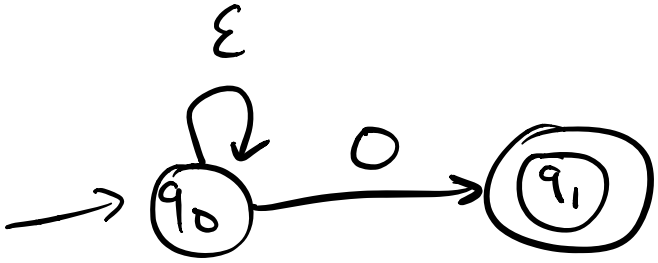
is a sequence of $n+1$ states $(r_0, r_1, \ldots, r_n)$ where:

- $r_0 = q_0$
- $r_{i+1} \in \delta(r_i, y_{i+1})$ for $0 \leq i < n$.



The <u>language</u> of an NFA M is the set of all strings for which M has <u>an</u> accepting computation ($r_n \in F$).

Ex:



How many accepting computations are there on 0?

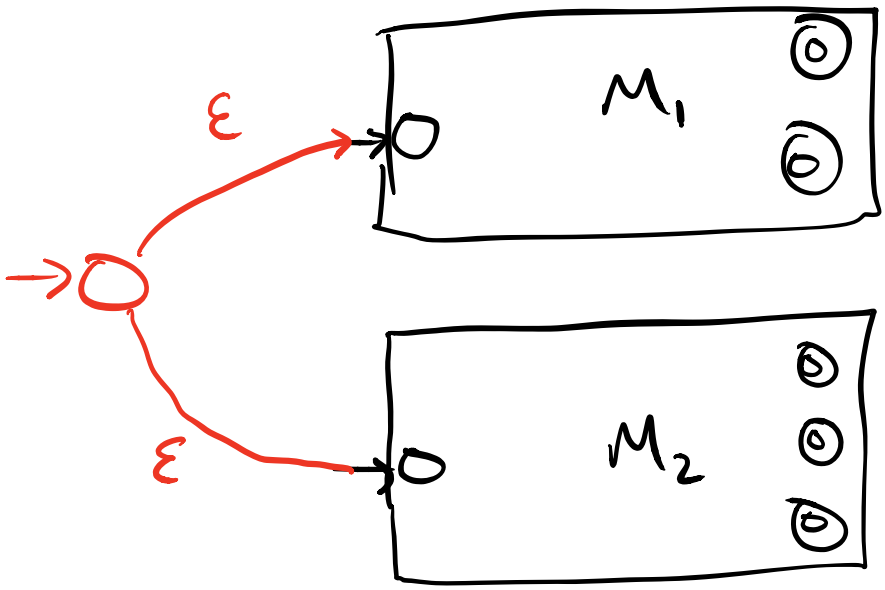
infinite

00?

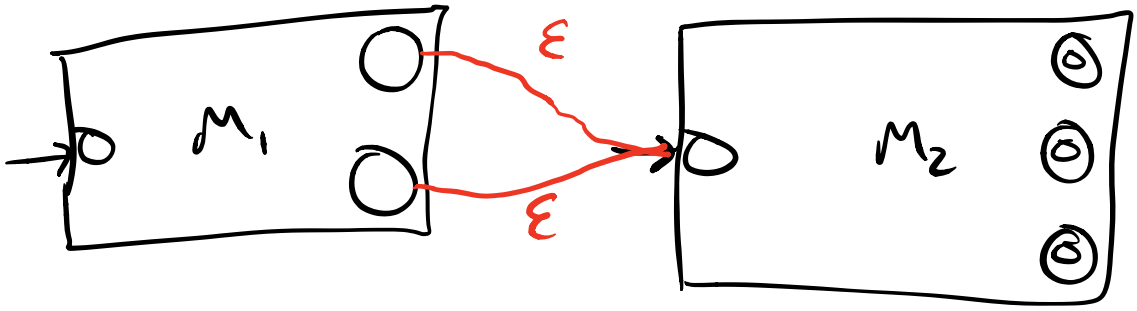
Zero

- Union

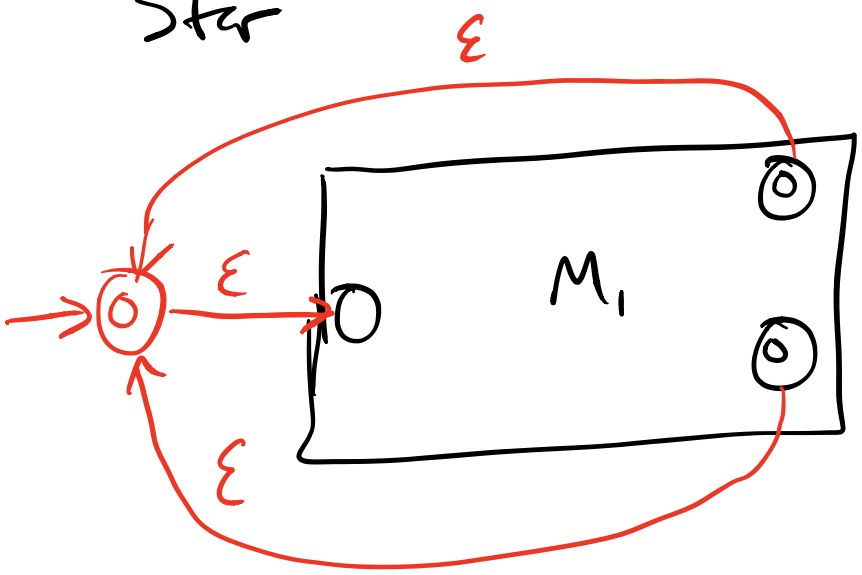
  Given NFAs M_1, M_2 produce an NFA for $L(M_1) \cup L(M_2)$?



Concatenation



Star



Problem Solving Session

Q: ≥2 Hampaths $\xrightarrow{?}$ is a cycle?

$L = \{w \in \{0,1\}^* :$ the binary representation of w is divisible by $5\}$.

0
101
1010

A DFA for L.

Make 5 states q_0, q_1, q_2, q_3, q_4

where q_i represents the string so far having remainder i when divided by 5.

101 – 5

1011 – 11

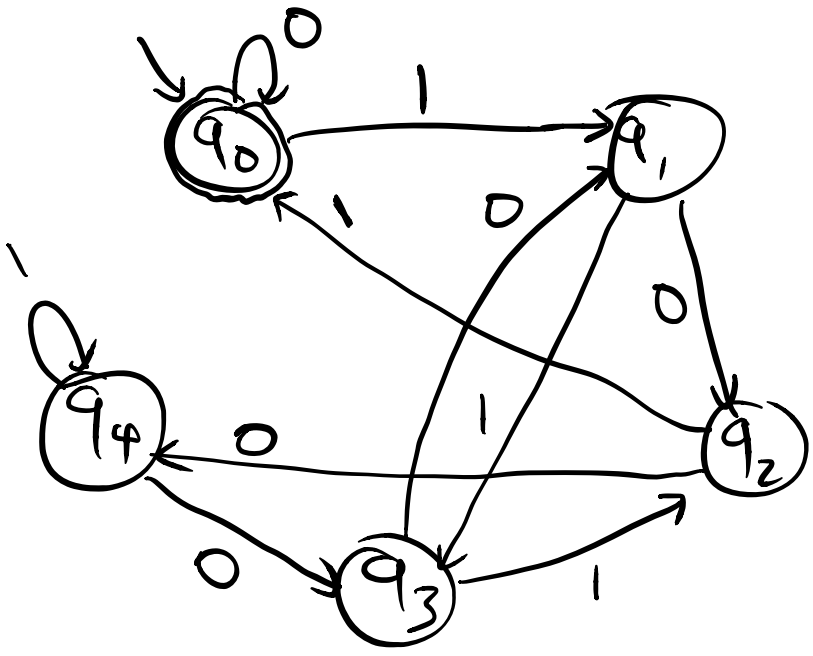
1000 8 (3)

10001 17 (2)

101 (5)

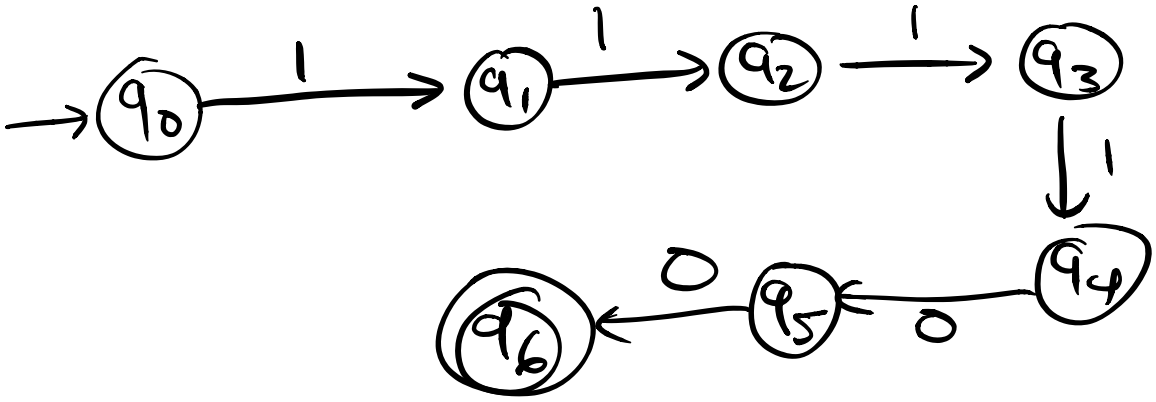
1010 (10)

1011 (11)



1	0	1	0	✓
q_0	q_1	q_2	q_0	q_0

$L = \{ w \in \{0,1,2\}^* : w$ contains the substring $111100\}$.

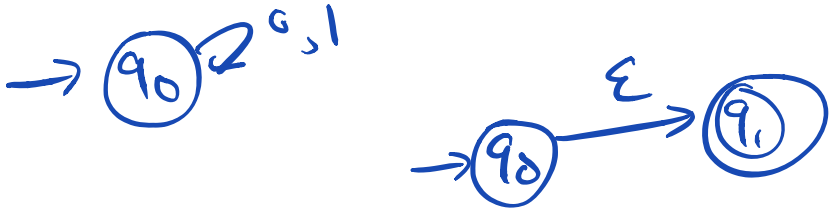


1111100_2

$L' = \{ w \in \{0,1,2\}^* : w$ ends in $111100\}$.

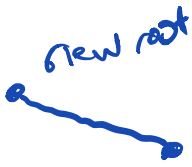
$L'' = \{111100\}$

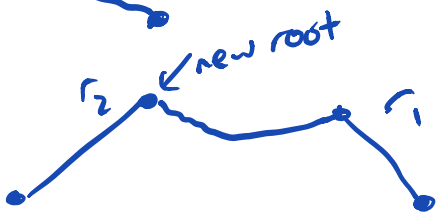
φ: $q_0 \notin F \rightarrow s_0 \quad \varepsilon \notin L$.



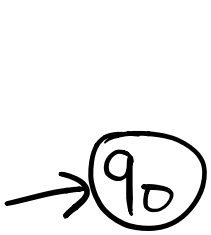
Q5: k-choice tree

$k=0$ •

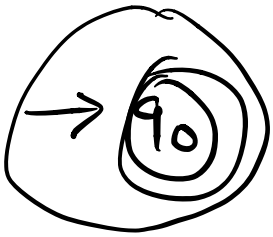
$k=1$ 

$k=2$ 

What if Σ for a DFA is $\emptyset$?



$L = \emptyset$



$L = \{\varepsilon\}$

$\emptyset^* = \{\varepsilon\} \cup \emptyset \cup \emptyset\emptyset \cup \cdots$

$\Sigma^* = \emptyset^* = \{\varepsilon\}$

If L is reg, how many DFAs recognize L?

≥ 1