NFAs

Sipser 1.2 (pages 47–54)

Recall…

- We showed that the class of regular languages is closed under:
  - Complement
  - Union
  - Intersection

Concatenation operation

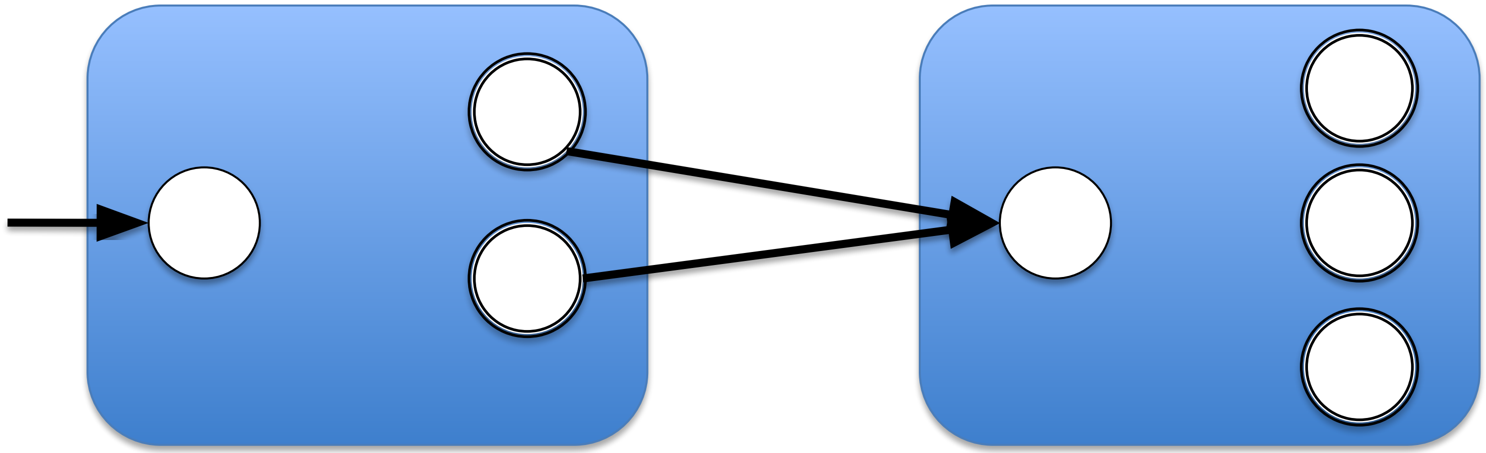
- Let A and B be languages.
- The **concatenation of A and B is** $A \circ B = \{xy \mid x \in A$ **and** $y \in B\}$**.**

Example

- Let $A = \{w \mid w$ is a string of 0s and 1s containing an odd number of 1s$\}$
- Let $B = \{w \mid w$ is a string of 0s and 1s containing an even number of 1s$\}$
- What are $A \circ B$; $B \circ A$; $A \circ A$; $B \circ B$?
  - Are any of these languages regular?

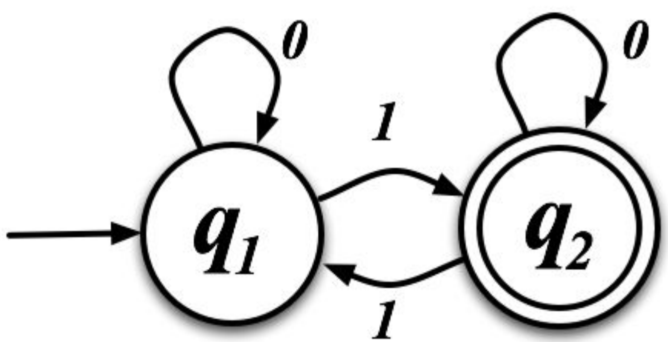
Concatenation

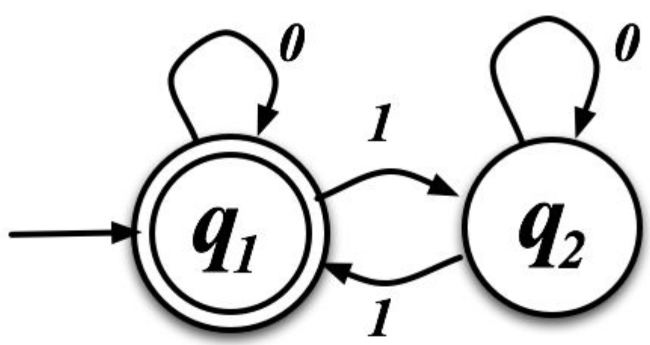
- Conjecture: The class of regular languages is *closed* under the *concatenation* operation.
- Proof idea:



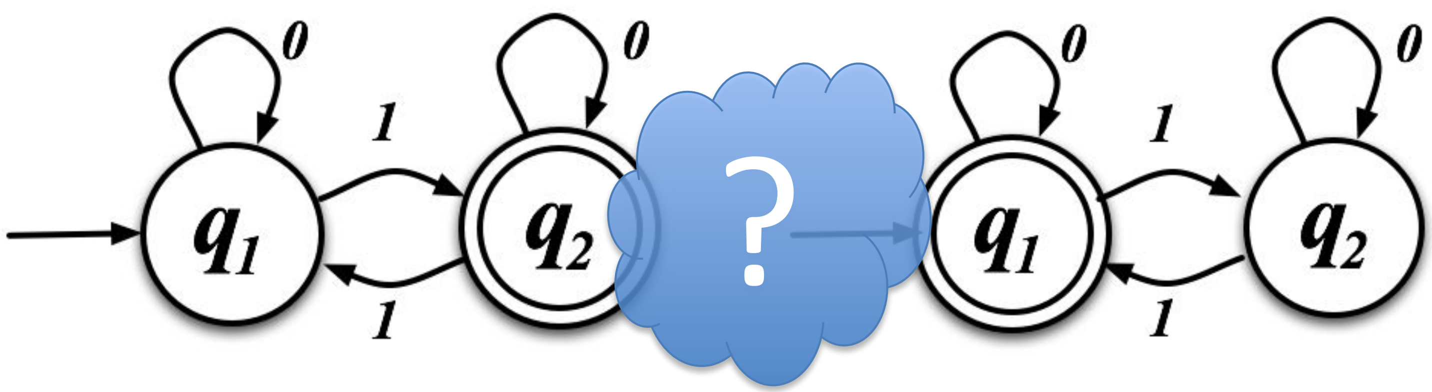
Hmm…

- $A = \{w \mid w$ is a string of 0s and 1s containing an odd number of 1s$\}$
- $B = \{w \mid w$ is a string of 0s and 1s containing an even number of 1s$\}$
- Create a machine by gluing… how?



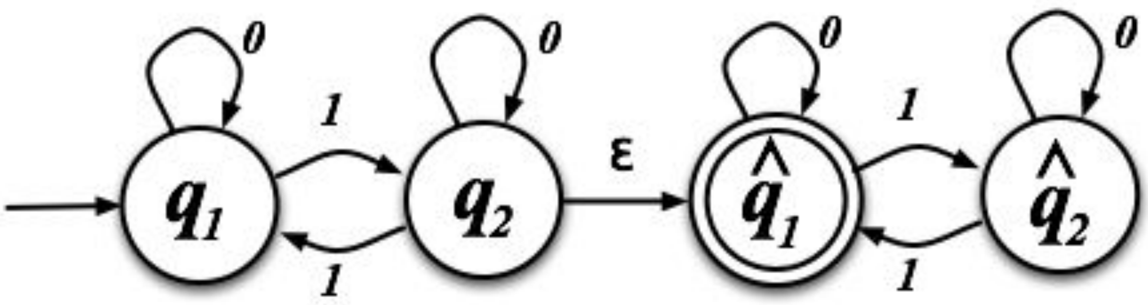


We need another approach



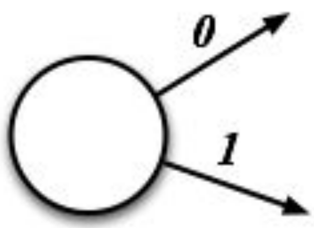
The empty symbol ε

- Seems like it might work
- Let's try running it…

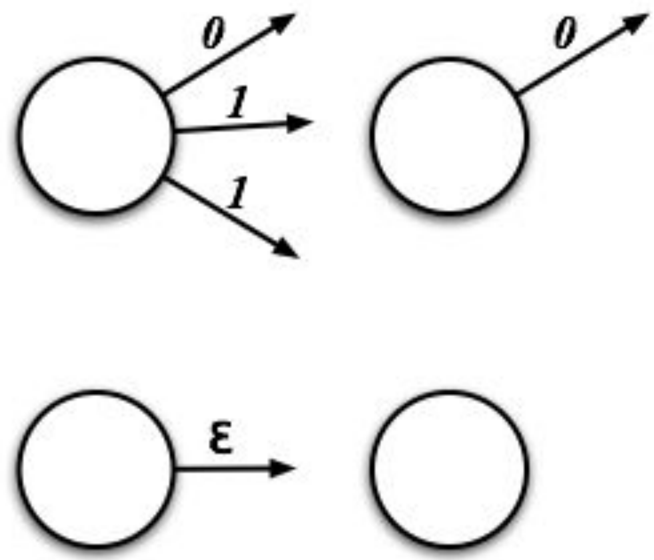


Relaxing the rules

Deterministic (DFA)



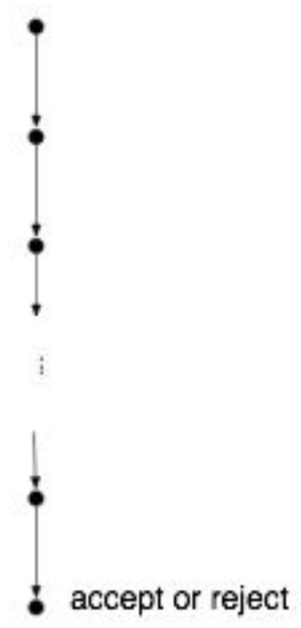
Nondeterministic (NFA)

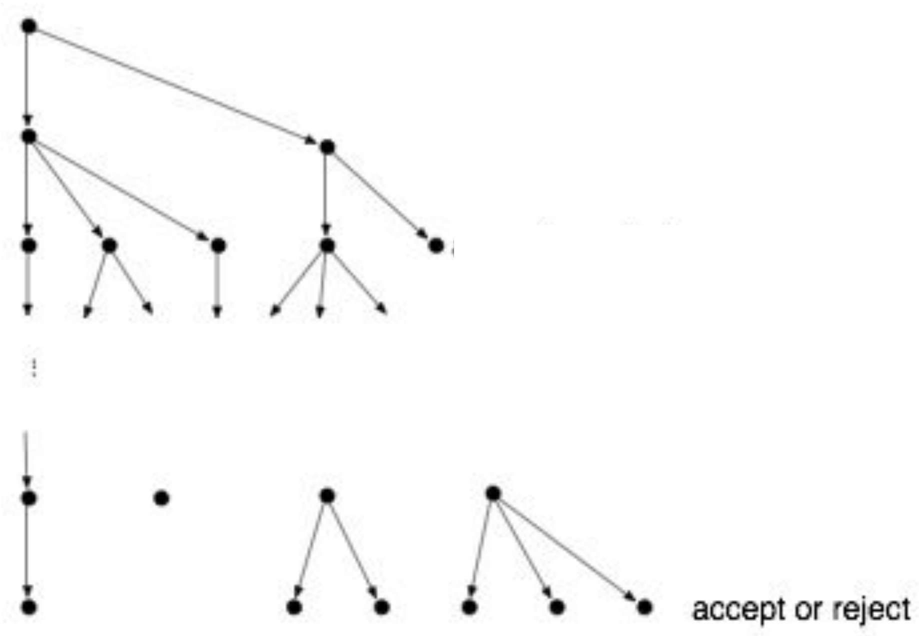


Running DFA vs NFA

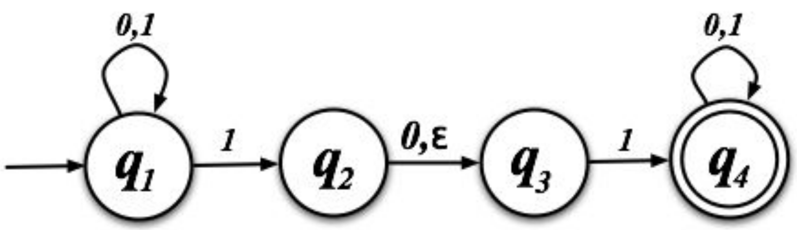
DFA computation (path)

NFA computation (tree)

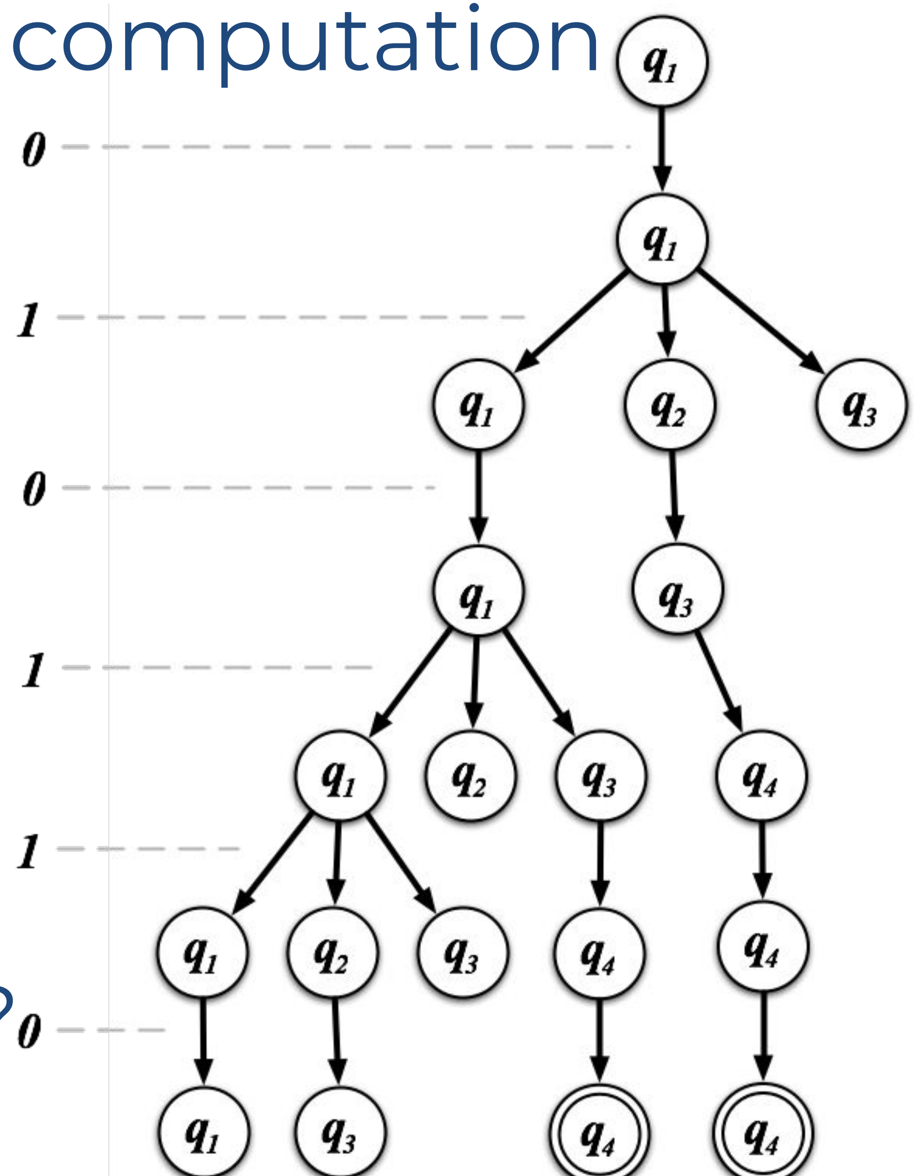




An example computation

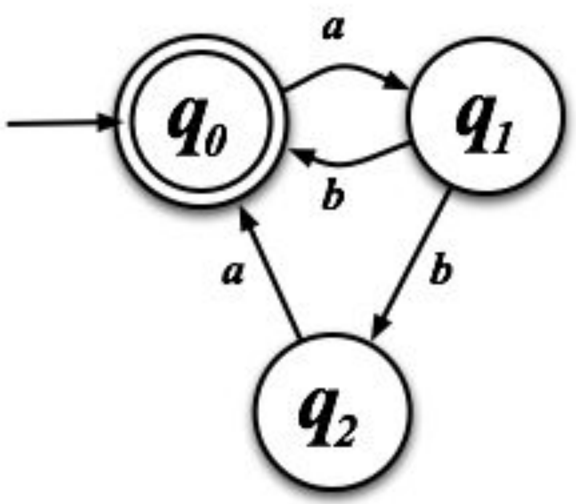


- What about 1011?



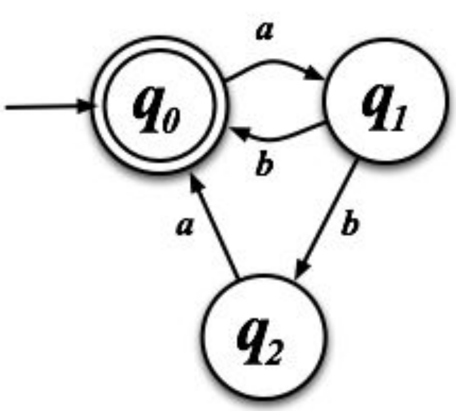
Another example

- What language is being recognized?
- *Hint:* can you start listing strings accepted?



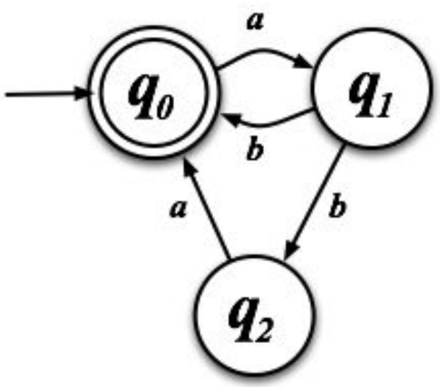
Formally…

- A ***nondeterministic finite automaton (NFA)*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where
  - Q is a finite set called the **states**
  - Σ is a finite set called the **alphabet**
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ is the **transition function**
  - $q_0 \in Q$ is the **start state**
  - $F \subseteq Q$ is a set of **accept states**
- In-class exercise:



NFA

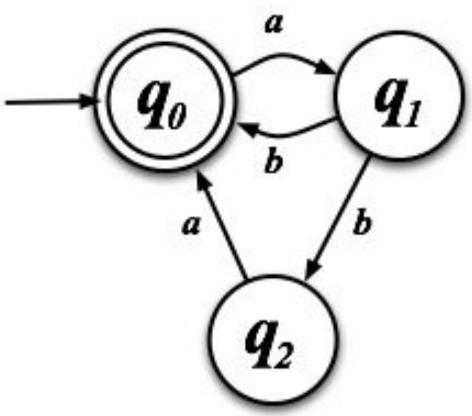
- A ***nondeterministic finite automaton (NFA)*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where
  - Q is a finite set called the **states**
  - Σ is a finite set called the **alphabet**
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ is the **transition function**
  - $q_0 \in Q$ is the **start state**
  - $F \subseteq Q$ is a set of **accept states**
- In-class exercise:



NFA computation

- Let $N=(Q, \Sigma, \delta, q_0, F)$ be a NFA and let w be a string over the alphabet Σ
- Then N **accepts** w if
  - w can be written as $w_1w_2w_3 \ldots w_m$ with each $w_i \in \Sigma\varepsilon$ and
  - There exists a sequence of states $s_0, s_1, s_2, \ldots, s_m$ exists in Q with the following conditions:

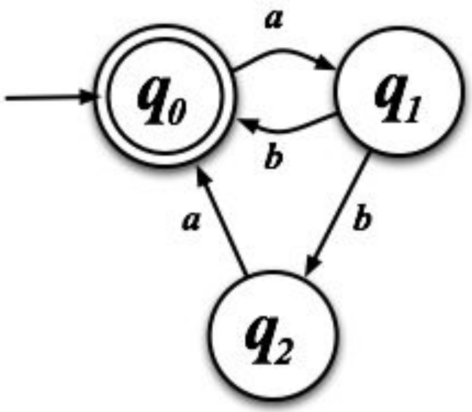
1. $s_0=q_0$
2. $s_{i+1} \in \delta(s_i, w_{i+1})$ for $i = 0, \ldots, m-1$
3. $s_m \in F$



NFA computation

- Let $N=(Q, \Sigma, \delta, q_0, F)$ be a NFA and let w be a string over the alphabet Σ
- Then N **accepts** w if
  - w can be written as $w_1w_2w_3\ldots w_m$ with each $w_i \in \Sigma\varepsilon$ and
  - There exists a sequence of states $s_0, s_1, s_2, \ldots, s_m$ exists in Q with the following conditions:

1. $s_0=q_0$
2. $s_{i+1} \in \delta(s_i, w_{i+1})$ for $i = 0,\ldots,m-1$
3. $s_m \in F$



Nondeterminism makes life easier

- Let's build an NFA that recognizes

  $B = \{w \mid w$ is a string over $\{a,b\}$ that starts and ends with the same symbol}

  $C = \{w \mid w$ is a string over $\{0,1\}$ that contains at least three 1's}

  $D = \{w \mid w$ is a string over $\{0,1\}$ that contains at least three consecutive 1's}

If at first you don't succeed…

… adjust your goal!

We wanted to prove:

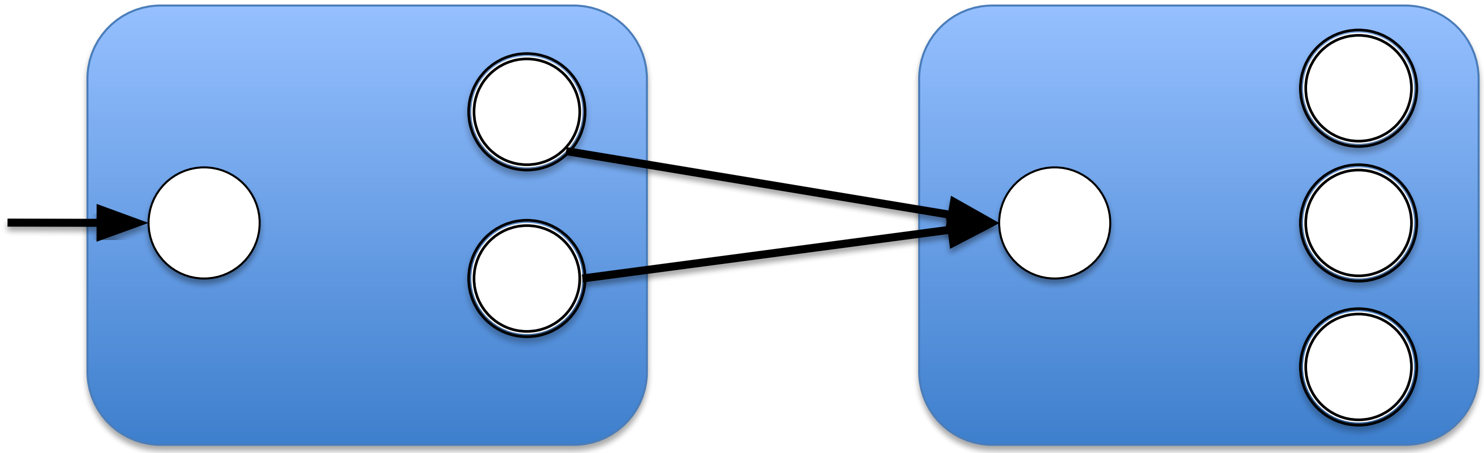
The class of regular languages is *closed* under the *concatenation* operation.

If at first you don’t succeed…

… adjust your goal!

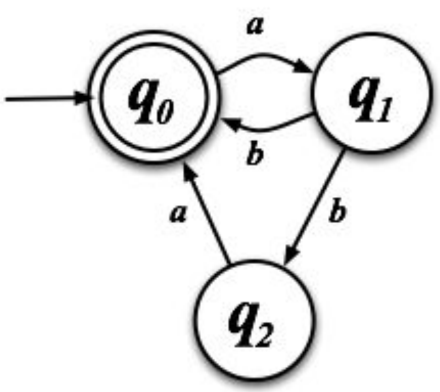
Instead, let’s prove the theorem:

The class of languages recognized by NFAs is *closed* under the *concatenation* operation.



NFA

- A ***nondeterministic finite automaton (NFA)*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where
  - Q is a finite set called the **states**
  - Σ is a finite set called the **alphabet**
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ is the **transition function**
  - $q_0 \in Q$ is the **start state**
  - $F \subseteq Q$ is a set of **accept states**
- In-class exercise:



Proof Let $A + B$ be languages recognized by NFAs $N_A + N_B$; $L(N_A) = A + L(N_B) = B$ where $N_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $N_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$.

We construct the NFA $N = (Q, \Sigma, \delta, \overset{q_A}{\cancel{q_0}}, \overset{F_B}{\cancel{F}})$, where $Q = Q_A \cup Q_B$

$$\delta(\underset{\in Q_A \cup Q_B}{q}, \underset{\in \Sigma_\varepsilon}{\sigma}) = \begin{cases} \delta_B(q,\sigma) & \text{if } q \in Q_B \\ \delta_A(q,\sigma) & \text{if } \underbrace{q \in Q_A \setminus F_A}_{\text{not accept in first machine}} \\ \{q_B\} \cup \delta_A(q,\sigma) & \text{if } q \in F_A \text{ and } \sigma = \varepsilon \\ \delta_A(q,\sigma) & \underbrace{\text{otherwise}}_{(q \in F_A \text{ and } \sigma \neq \varepsilon)} \end{cases}$$

$\cancel{q_0 = q_A}$

$\cancel{F = F_B}$

Since $L(N) = A \circ B$, the class of lang. recog by NFAs is closed under $\circ$.