Closure under regular operations I

- Recall we defined three operations:

$$\cup, \circ, *$$

- For

$$A_1 \cup A_2,$$

  we proved that it's regular by constructing a new DFA

Closure under regular operations II

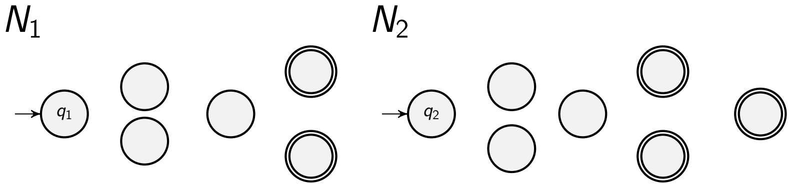
- But we had difficulties to prove that

$$A_1 \circ A_2$$

  is regular
- We will see that by using NFA, the proof is easier

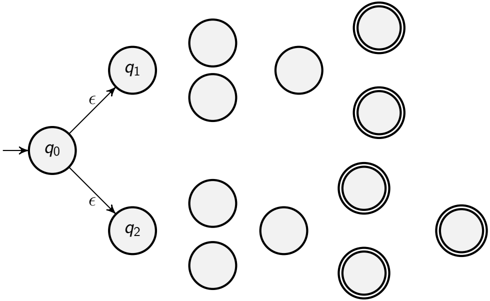
Union I

- Given two regular languages A_1, A_2 under the **same** Σ
  Is $A_1 \cup A_2$ regular?
- To prove that a language is regular, by definition, it should be accepted by one DFA (or an NFA)
  We will construct an NFA for $A_1 \cup A_2$
- Assume A_1 and A_2 are recognized by two NFAs N_1 and N_2, respectively



Union II

- We construct the following machine



- Formal definition

Union III

Two NFAs:

$$N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$$
$$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$$

Note for NFA, $\epsilon \notin \Sigma$

Union IV

- New NFA

$$
\begin{aligned}
Q &= Q_1 \cup Q_2 \cup \{q_0\} \\
q &= q_0 \\
F &= F_1 \cup F_2
\end{aligned}
$$

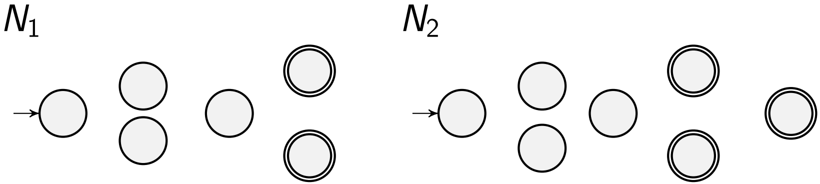
$$
\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \\ \delta_2(q, a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \epsilon \\ \emptyset & q = q_0 \text{ and } a \neq \epsilon \end{cases}
$$

Union V

- The last case of δ is easily neglected

Closed Under Concatenation I

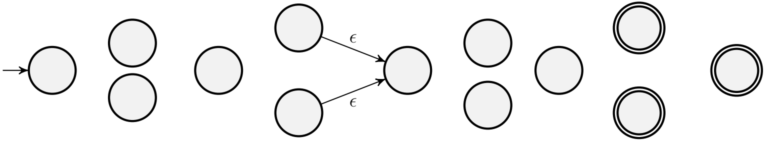
Given two NFAs



- Idea: from any accept state of N_1, add an ϵ link to q_2 (start state of N_2)
- Earlier in using DFA, the difficulty was that we didn't know where to cut the string to two parts

Closed Under Concatenation II

- Now we non-deterministically switch from the first to the second machine
- The new machine:



- Accept states of N_1 are no longer accept states in the new machine

Closed Under Concatenation III

- Formal definition. Given two automata

$$(Q_1, \Sigma, \delta_1, q_1, F_1)$$
$$(Q_2, \Sigma, \delta_2, q_2, F_2)$$

- New machine

$$Q = Q_1 \cup Q_2$$
$$q_0 = q_1$$
$$F = F_2$$

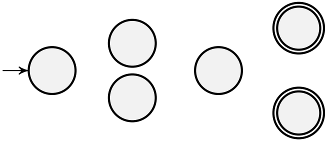
Closed Under Concatenation IV

δ function:

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \backslash F_1 \\ \delta_2(q, a) & q \in Q_2 \\ \delta_1(q, \epsilon) \cup \{q_2\} & q \in F_1, a = \epsilon \\ \delta_1(q, a) & q \in F_1, a \neq \epsilon \end{cases}$$

Closed under star I

- Given the following machine

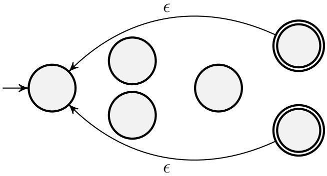


- Recall the star operation is defined as follows

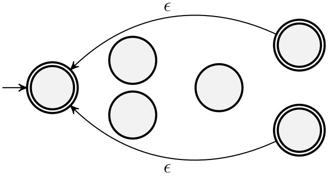
$$A^* = \{x_1 \cdots x_k \mid k \geq 0, x_i \in A\}$$

- The situation is related to $A_1 \circ A_2$, but we now work on the same machine A
- How about the following diagram

Closed under star II

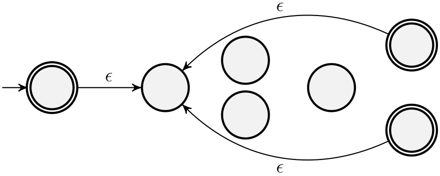


- The problem is that ϵ may not be accepted
- How about making the start state an accepting one



Closed under star III

- This may make the machine to accept strings not in A
- Some strings reaching the start state in the end were rejected. But now may be accepted
- A correct setting



- Formal definition

Closed under star IV

- Given the machine

$$(Q_1, \Sigma, \delta_1, q_1, F_1)$$

- New machine:

$$\begin{aligned} Q &= Q_1 \cup \{q_0\} \\ q_0 &: \text{ new start state} \\ F &= F_1 \cup \{q_0\} \end{aligned}$$

Closed under star V

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \setminus F_1 \\ \delta_1(q, a) \cup \{q_1\} & q \in F_1, a = \epsilon \\ \delta_1(q, a) & q \in F_1, a \neq \epsilon \\ \{q_1\} & q = q_0, a = \epsilon \\ \emptyset & q = q_0, a \neq \epsilon \end{cases}$$