Regular Languages’ Closure Properties, Take 1
Closure Properties for Regular Languages

**Union:** \( A \cup B = \{ w \mid w \in A \text{ or } w \in B \} \)

**Intersection:** \( A \cap B = \{ w \mid w \in A \text{ and } w \in B \} \)

**Complement:** \( \overline{A} = \{ w \in \Sigma^* \mid w \notin A \} \)

**Reverse:** \( A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A, w_i \in \Sigma \} \)

**Concatenation:** \( A \cdot B = \{ vw \mid v \in A \text{ and } w \in B \} \)

**Star:** \( A^* = \{ s_1 \ldots s_k \mid k \geq 0 \text{ and each } s_i \in A \} \)

**Theorem:** if \( A \) and \( B \) are regular then so are: \( A \cup B, A \cap B, \overline{A}, A^R, A \cdot B, \) and \( A^* \)
Union Theorem for Regular Languages

Given two languages $L_1$ and $L_2$, recall that the union of $L_1$ and $L_2$ is

$$L_1 \cup L_2 = \{ w \mid w \in L_1 \text{ or } w \in L_2 \}$$

**Theorem:** The union of two regular languages is also a regular language
Theorem: The union of two regular languages is also a regular language.

Proof: Let

\[ M_1 = (Q_1, \Sigma, \delta_1, q^1_0, F_1) \]
be a finite automaton for \( L_1 \)

and

\[ M_2 = (Q_2, \Sigma, \delta_2, q^2_0, F_2) \]
be a finite automaton for \( L_2 \)

We want to construct a finite automaton

\[ M = (Q, \Sigma, \delta, q_0, F) \]
that recognizes \( L = L_1 \cup L_2 \)
Proof Idea: Run both $M_1$ and $M_2$ “in parallel”!

$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ recognizes $L_1$ and

$M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$ recognizes $L_2$

$Q = \text{pairs of states, one from } M_1 \text{ and one from } M_2$

$= \{ (q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$

$= Q_1 \times Q_2$

$q_0 = (q_0^1, q_0^2)$

$F = \{ (q_1, q_2) \mid q_1 \in F_1 \text{ OR } q_2 \in F_2 \}$

$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$
**Theorem:** The union of two regular languages is also a regular language.
What about the INTERSECTION of two languages?
Proof Idea: Run both $M_1$ and $M_2$ “in parallel”!

$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ recognizes $L_1$ and $M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$ recognizes $L_2$

$Q = \text{pairs of states, one from } M_1 \text{ and one from } M_2$

$= \{ (q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$

$= Q_1 \times Q_2$

$q_0 = (q_0^1, q_0^2)$

$F = \{ (q_1, q_2) \mid q_1 \in F_1 \text{ AND } q_2 \in F_2 \}$

$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$
Union Theorem for Regular Languages:
The union of two regular languages is also a regular language
“Regular Languages Are Closed Under Union”

Intersection Theorem for Regular Languages:
The intersection of two regular languages is also a regular language
“Regular Languages Are Closed Under Intersection”
Complement Theorem for Regular Languages

The complement of a regular language is also a regular language:

If $L$ is regular than so is $\overline{L} = \{ w \in \Sigma^* | w \notin L \}$

Proof?
Closure Properties for Regular Languages

- **Union:** $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$
- **Intersection:** $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$
- **Complement:** $\overline{A} = \{ w \in \Sigma^* \mid w \notin A \}$
- **Reverse:** $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A, \ w_i \in \Sigma \}$
- **Concatenation:** $A \cdot B = \{ vw \mid v \in A \text{ and } w \in B \}$
- **Star:** $A^* = \{ s_1 \ldots s_k \mid k \geq 0 \text{ and each } s_i \in A \}$

**Theorem:** If $A$ and $B$ are regular then so are: $A \cup B$, $A \cap B$, $\overline{A}$, $A^R$, $A \cdot B$, and $A^*$
The Reverse of a Language

Reverse of L:
\[ L^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in L, w_i \in \Sigma \} \]

**Theorem:** If L is regular, then is \( L^R \) also regular

If \( L \) is recognized by the usual kind of DFA, Then \( L^R \) is recognized by a DFA that reads its strings from *right to left*

How can every “Right-to-Left” DFA be replaced by a normal “Left-to-Right” DFA?
Reversing DFAs

Assume $L$ is a regular language. Let $M$ be a DFA that recognizes $L$. We want to build a machine $M^R$ that accepts $L^R$.

If $M$ accepts $w$, then $w$ describes a directed path in $M$ from the start state to an accept state.

**First Attempt:** Try to define $M^R$ as $M$ with the arrows reversed, turn start state into a final state, turn final states into starts.
Problem: $M^R$ is not always a DFA!

It could have many start states

Some states may have *more than one* outgoing edge, or none at all!
Non-deterministic Finite Automata (NFA)

What happens with 100?

We will say this new machine accepts a string if there is some path that reaches some accept state from some start state.