It is often the case that we want to know how we can manipulate a class of objects and still end up with something in the same class (see Group Theory/Abstract Algebra). We call these manipulations (operations) **closed** on a class if the above property holds.

**Ex:** The integers are closed over addition.

\[ x, y \in \mathbb{Z}, \quad x + y \in \mathbb{Z}. \]

The regular languages are closed under a number of interesting properties. We’ve seen a few in exercises so far for specific cases. Here, we’ll formally prove three particularly important closure properties.

**Thm:** The regular languages are closed under concatenation.

**Pf:** Let \( A, B \) be regular languages. Then there are NFAs \( N_A \) and \( N_B \) which accept \( A, B \) respectively. We construct a new NFA \( N \) to recognize \( A \cdot B \) as follows.

[Diagram of an NFA construction for concatenation]
Let \( N_A = (Q_A, \Sigma_A, S_A, \delta_A, F_A) \) and \( N_B = (Q_B, \Sigma_B, S_B, \delta_0, F_B) \).

Define \( N = (Q_A \cup Q_B, \Sigma_A \cup \Sigma_B, S, \delta_0, A \cup B) \), where for \( c \in (\Sigma_A \cup \Sigma_B)_e \) and \( q \in Q_A \cup Q_B \),

\[
\delta(q, c) = \begin{cases} 
\delta_A(q, c) & q \in Q_A \setminus F_A \quad \text{and} \quad c \in (\Sigma_A)_e \\
\delta_A(q, c) & q \in F_A \quad \text{and} \quad c \in \Sigma_A \\
(q_0, B) \delta_0 A(q, c) & q \in F_A \quad \text{and} \quad c = \epsilon \\
\delta_B(q, c) & q \in Q_B \quad \text{and} \quad c \in (\Sigma_B)_e
\end{cases}
\]

It remains to show \( L(N) = A \cup B \).

Let \( w \in L(N) \). Then \( \exists \) an accepting path \( p \) from \( q_0 \) to some \( q_F \in F_B \) on input \( w \). By construction, \( q_0, B \) must be in \( p \) and \( p \) gets to \( q_0, B \) via an \( \epsilon \) transition from a state \( q_F' \in F_A \).

Thus \( p = rs \) where \( r \) is a path starting from \( q_0, A \) and ending in \( q_F' \) while \( s \) is a path starting from \( q_0, B \) and ending at \( q_F \).

As such, \( r \) accepts some string in \( A \) and \( s \) accepts some string in \( B \), hence \( w \in A \cup B \).

Now let \( w \in A \cup B \). Similarly to before, \( w = uv \) such that \( u \in A \) and \( v \in B \), so \( \exists \) accepting paths \( r \) and \( s \) for \( u \) and \( v \) in \( N_A \) and \( N_B \) respectively. In \( N \), \( r \) and \( s \) is bridged via an \( \epsilon \) transition by construction, so \( rs \) is a path in \( N \). Moreover, since \( s \) end in a state in \( F_B \), \( rs \) is an accepting path, thus \( rs \in L(N) \).
The regular languages are closed under union.

**Proof** Let \( A, B \) be regular languages. Then there are NFAs \( N_A = (Q_A, \Sigma_A, S_A, \delta_A, F_A) \) and \( N_B = (Q_B, \Sigma_B, S_B, \delta_B, F_B) \) such that \( L(N_A) = A \) and \( L(N_B) = B \).

We construct NFA \( N \) to accept \( A \cup B \) as follows.

![Diagram of NFA N]

We define \( N = (Q_A \cup Q_B, \Sigma_A \cup \Sigma_B, \delta, \epsilon_0, F_A \cup F_B) \), where for \( q \in Q_A \cup Q_B \) and \( c \in (\Sigma_A \cup \Sigma_B)_\epsilon \):

\[
\delta(q, c) = \begin{cases} 
\delta_A(q, c) & q = \epsilon_0 \text{ and } c = \epsilon \\
\delta_B(q, c) & q \in Q_B \text{ and } c \in (\Sigma_B)_\epsilon \\
\delta_A(q, c) & q \in Q_A \text{ and } c \in (\Sigma_A)_\epsilon
\end{cases}
\]

It is routine to verify that \( L(N) = A \cup B \).

The regular languages are closed under the (Kleene) star.

**Proof** Let \( A \) be a regular language. Then there is an NFA \( N_A = (Q_A, \Sigma_A, S_A, \delta_A, F_A) \) such that \( L(N_A) = A \).

We construct NFA \( N \) to accept \( A^* \) as follows.
We define \( N = (Q_A, \Sigma, \delta, q_0, F_A, \Sigma, q_0, F_A) \) where for \( q \in Q_A \cup \Sigma \), and \( c \in (\Sigma_A)^e \)

\[
\delta(q, c) = \begin{cases} 
\delta(q, c) & q \in Q_A \cap F_A \\
\delta(q, c) & q \in F_A \text{ and } c \in \Sigma_A \\
\delta(q, c) \cup \delta(q, c) & q \in F_A \text{ and } c = \varepsilon \\
\delta(q, c) & q = q_0 \text{ and } c = \varepsilon.
\end{cases}
\]

Note that because we must accept \( \varepsilon \) and don't know if \( N_A \) contains a cycle starting at \( q_A \), we must add accepting start state \( q_0 \).

From here it is routine to verify that \( L(N) = A^* \).

---

**Def:** For a language \( A \) over an alphabet \( \Sigma \),

\[
A^* = \{ w_1 w_2 w_3 \ldots w_k \mid \forall i, w_i \in \Sigma^* \text{ and } 0 \leq k \}
\]

\( \text{when } k = 0, \text{ we get the empty string} \)

For \( w \in \Sigma^* \), \( w^* = \varepsilon w_3^* \).