Recap: Formal Definition of DFA

Formally, a **Deterministic Finite Automaton** or **DFA** is defined by a 5-tuple

\[(Q, \Sigma, \delta, q_0, F)\]

where

- **Q**: Finite set of States
- \(\Sigma\): Alphabet (finite set of symbols)
- \(\delta \in Q \times \Sigma \rightarrow Q\): Transition Function
- \(q_0 \in Q\): Initial or Start State
- \(F \subseteq Q\): Accepting (or Final) States

Recap: Extended Transition Function

The **Extended Transition Function** is defined on a state and a **word** (string of symbols) instead of on a single symbol.

For a DFA \(A = (Q, \Sigma, \delta, q_0, F)\), the extended transition function is defined by:

\[
\hat{\delta} \in Q \times \Sigma^* \rightarrow Q
\]

\[
\hat{\delta}(q, \epsilon) = q
\]

\[
\hat{\delta}(q, xw) = \hat{\delta}(\hat{\delta}(q, x), w)
\]

where \(q \in Q\), \(x \in \Sigma\), \(w \in \Sigma^*\).

Recap: Language of a DFA

The **language** \(L(A)\) defined by a DFA \(A\) is the set of words **accepted** by the DFA. For a DFA

\[A = (Q, \Sigma, \delta, q_0, F)\]

the language is defined by

\[L(A) = \{ w \in \Sigma^* | \hat{\delta}(q_0, w) \in F \}\]