Additional Definitions

**Defn:** The *length* of a string $x \in \Sigma^*$ is defined recursively as

- $0$ if $x = \epsilon$; or
- the length of $y$ plus $1$ if $x = ay$, where $a \in \Sigma$, $y \in \Sigma^*$.

**Defn:** A *language* over an alphabet $\Sigma$ is a subset of $\Sigma^*$.

Deterministic Finite Automata

**Defn:** A *deterministic finite automaton* (DFA) is a $5$-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- $Q$ is a finite set of *states*;
- $\Sigma$ is the *input alphabet*;
- $\delta : Q \times \Sigma \rightarrow Q$ is the *transition function*;
- $q_0 \in Q$ is the *start state*; and
- $F \subseteq Q$ is the set of *final states* (or *accepting states*).
Behavior of DFAs

**Defn:** Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. We define $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ recursively as follows:

- $\hat{\delta}(q, \epsilon) = q$ for all $q \in Q$; and
- $\hat{\delta}(q, ay) = \hat{\delta}(\hat{\delta}(q, a), y)$ for all $q \in Q$, $a \in \Sigma$, $y \in \Sigma^*$.

Regular Languages

**Defn:** Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. The **language accepted by** $A$ is defined to be

$$L(A) = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \in F\}.$$ 

**Defn:** Let $L$ be a language over $\Sigma$. We say that $L$ is **regular** if there is a DFA $A = (Q, \Sigma, \delta, q_0, F)$ such that $L(A) = L$. 

Claim: For every $q \in Q$, $a \in \Sigma$, and $y \in \Sigma^*$, 
\[ \hat{\delta}(q, ya) = \delta(\hat{\delta}(q, y), a). \]

Proof: By induction on $y \in \Sigma^*$.

Base: $y = \epsilon$. Then 
\[
\hat{\delta}(q, ya) = \hat{\delta}(q, a) \\
= \hat{\delta}(\delta(q, a), \epsilon) \\
= \delta(q, a) \\
= \delta(\hat{\delta}(q, y), a).
\]

IH: Assume that for some $y \in \Sigma^*$, 
\[ \hat{\delta}(q, ya) = \delta(\hat{\delta}(q, y), a) \] for all $q \in Q$ and $a \in \Sigma$.

IS: Let $a, b \in \Sigma$, $q \in Q$. Then 
\[
\hat{\delta}(q, bya) = \hat{\delta}(\delta(q, b), ya) \\
= \delta(\hat{\delta}(\delta(q, b), y), a) \quad \text{(by the IH)} \\
= \delta(\hat{\delta}(q, by), a).
\]
Non-deterministic Finite Automata

**Defn:** A *non-deterministic finite automaton* (NFA) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where \(Q\), \(\Sigma\), \(q_0\), and \(F\) are as given in the definition of a DFA, and

\[
\delta : Q \times \Sigma \rightarrow 2^Q.
\]

**Defn:** Let \((Q, \Sigma, \delta, q_0, F)\) be an NFA. We define \(\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q\) recursively as follows:

- \(\hat{\delta}(q, \epsilon) = \{q\}\) for all \(q \in Q\); and
- \(\hat{\delta}(q, ay) = \bigcup_{p \in \delta(q, a)} \hat{\delta}(p, y)\) for all \(q \in Q\), \(a \in \Sigma\), and \(y \in \Sigma^*\).

**Defn:** Let \(A = (Q, \Sigma, \delta, q_0, F)\) be an NFA. The *language accepted by* \(A\) is defined to be

\[
L(A) = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \cap F \neq \emptyset\}.
\]
Claim: Let \((Q, \Sigma, \delta, q_0, F)\) be an NFA. Then 
\[ \hat{\delta}(q, ya) = \bigcup_{p \in \hat{\delta}(q, y)} \delta(p, a) \] 
for all \(q \in Q\), \(y \in \Sigma^*\), and \(a \in \Sigma\).