1 Defining NFST

Definition. A Nondeterministic Finite State Transducer (NFST) is a 5-tuple $M = (Q, \Sigma, \Gamma, \delta, s)$ consisting of

- A finite set of states $Q$
- A finite set of input symbols $\Sigma$
- A finite set of output symbols $\Gamma$
- A transition function $\delta : Q \times \Sigma \to \mathcal{P}(Q \times \Gamma^*)$
- A start state $s \in Q$

As all automata, at any given point during a computation, the internal state of an NFST is described by an element $q \in Q$ of the set of states. Initially the state is $s$, and it may change during the computation. During a computation, an NFST reads a string $w \in \Sigma^*$ over the input alphabet $\Sigma$, and outputs a string $u \in \Gamma^*$ over a possibly different alphabet $\Gamma$. The input symbols are read one at a time. When the machine is in state $q \in Q$ and reads the symbol $a \in \Sigma$, it selects (nondeterministically) an element $(p, w) \in \delta(q, a)$, updates its internal states to $p \in Q$ and prints $w \in \Gamma^*$. The output of the computation is obtained by concatenating the strings printed at every step. Since the machine is nondeterministic, there are several possible computations corresponding to the same input $w$, each producing a potentially different output string. So, the behavior of an NFST is described by a function $f_M : \Sigma^* \to \mathcal{P}(\Gamma^*)$ mapping the input string $w \in \Sigma^*$ to a set $f_M(w)$ of possible outputs. (This set can be empty if all computation branches abort.)

In order to formally define the output of an NFST, we first extend the transition function $\delta$ to a function $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q \times \Gamma^*)$ that can take strings as input, rather than single symbols.

Definition. Let $M = (Q, \Sigma, \Gamma, \delta, s)$ be an NFST. The extended transition function $\delta^*(q, w)$ is defined by induction on the length of $w$ as follows:

- Base case ($|w| = 0$): for every $q \in Q$, let $\delta^*(q, \epsilon) = \{(q, \epsilon)\}$
- Inductive case ($|w| > 0$): for every $a \in \Sigma$ and $w' \in \Sigma^*$, let
  $$\delta^*(q, aw') = \{(q'', u''u') : \exists q' \in Q, (q', u') \in \delta(q, a) \land (q'', u'') \in \delta^*(q', w')\}$$

The set of possible outputs of $M$ on input $w$ is defined as

$$f_M(w) = \{u \in \Gamma^* : \exists q \in Q, (q, u) \in \delta^*(s, w)\}$$

i.e., the set of all strings that can be obtained starting from the initial state $s$ and reading $w$. 

2 Closure of NFST under composition

Each NFST describes a function \( f: \Sigma^* \to \mathcal{P}(\Gamma^*) \), representing a (nondeterministic) system that on input a string \( w \in \Sigma^* \) may produce one of many possible output strings \( u \in f(w) \subseteq \Gamma^* \). Such functions can be composed together in a natural way: given \( f: \Sigma^* \to \mathcal{P}(\Gamma^*) \) and \( g: \Gamma^* \to \mathcal{P}(\Delta^*) \), the function composition of \( f \) and \( g \) is the function \( g \circ f: \Sigma^* \to \mathcal{P}(\Delta^*) \) defined as \( g \circ f(w) = g(f(w)) \), i.e.,

\[
g \circ f(w) = \{ v \in \Delta^* : \exists u \in f(w). v \in g(u) \}.
\]

In the homework you were asked to prove that (deterministic) FSTs are closed under composition, i.e., if \( f \) and \( g \) are computed by FSTs, then also the composite function \( g \circ f \) is computed by an FST. Here we prove a similar result for NFST.

**Theorem.** For any NFST \( M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, s_1) \) and \( M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, s_2) \) with compatible alphabets \( \Gamma_1 = \Gamma_2 \), there is an NFST \( M = M_2 \circ M_1 \) such that \( f_M = f_{M_2} \circ f_{M_1} \).

**Proof.** The NFST \( M \) is defined as follows. Let \( M = (Q, \Sigma_1, \Gamma_2, \delta, s) \) where \( Q = Q_1 \times Q_2 \), \( s = (s_1, s_2) \in Q \) and \( \delta: Q \times \Sigma_1 \to \mathcal{P}(Q \times \Gamma^*_2) \) is the function defined as

\[
\delta((q_1, q_2), a) = \{(q'_1, q'_2), v) : \exists u \in \Gamma_1. (q'_1, u) \in \delta_1(q_1, a) \land (q'_2, v) \in \delta_2(q_2, u) \}.
\]

It can be easily verified that \( f_M = f_{M_2} \circ f_{M_1} \).

A few words of explanation are due. The intuition behind the above construction is the following. The NFST \( M_2 \circ M_1 \) works by running \( M_1 \) on the input string \( w \in \Sigma_1^* \) to obtain some intermediate result \( u \in \Gamma_1^* \). The output alphabet \( \Gamma_1 \) is required to be the same as the input alphabet \( \Sigma_2 \) so that the output of \( M_1 \) can be fed as input to \( M_2 \). As \( M_1 \) outputs \( w \), the composed automaton \( M_2 \circ M_1 \) runs the second NFST on \( w \) to obtain the final output string \( v \). Since finite automata (and NFST in particular) do not have enough memory to store the intermediate result of the computation \( w \), the two component automata \( M_1, M_2 \) are run at the same time, and the output of \( M_1 \) is fed to \( M_2 \) as it is being produced. In order to run the two automata at the same time, we use the cartesian product \( Q_1 \times Q_2 \) as the set of states of the composite automaton. Each state \((q_1, q_2) \in Q \) records the current state of \( M_1 \) and the current state of \( M_2 \). When a symbol \( a \in \Sigma_1 \) is read from the input, we first invoke the transition function of \( M_1 \) to obtain all possible actions \((q'_1, u) \in \delta_1(q_1, a) \) that \( M_1 \) can perform. Recall that \((q'_1, u) \) means “move to state \( q'_1 \) and output \( u \)”. Accordingly, the composite automaton runs \( M_2 \) on input \( u \), starting from the current state \( q_2 \), to obtain an updated state and output string \((q'_2, v) \in \delta_2(q_2, u) \). Notice that since the intermediate output \( u \in \Gamma_1^* \) is not just a symbol, but a string, we need to use the extended transition function \( \delta_2^* \).