1 Non-deterministic Finite Automata (NFAs)

A Non-deterministic Finite Automaton (NFA) is a 5-tuple \((\Sigma, Q, q_0, A, \delta)\) where:

- \(\Sigma\) – the input alphabet
- \(Q\) – finite set of states
- \(q_0 \in Q\) – a starting state in the set of states
- \(A \subseteq Q\) – set of accepting states
- \(\delta : Q \times \Sigma \rightarrow P(Q)\) – the transition relation (some use \(2^Q\) to represent the power set of \(Q\) rather than \(P(Q)\))

The key difference between NFAs and DFAs is the transition relation. For DFAs—the transition relation is \(\delta : Q \times \Sigma \rightarrow Q\) — each \(\delta(q, \sigma)\) is a single state for any \(q \in Q\) and \(\sigma \in \Sigma\).

For NFAs, each \(\delta(q, \sigma)\) is a set of states, i.e. an element in the power set \(P(Q)\). Recall what power sets are:

\[
P(\{x, y, z\}) = \{\{\}, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}.
\]

This means that given some input, an NFA can be in multiple states at once.

1.1 \(\epsilon\)-NFA

An \(\epsilon\)-NFA allows for the use of \(\epsilon\)-transitions. \(\epsilon\)-transitions represent the transition from one state to another without consuming any input. This is useful when we want to try to connect multiple NFAs into one.

\(\epsilon\)-NFAs have a very similar definition as NFAs, but we add \(\epsilon\)-transitions for every state to the transition relation: \(\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)\)

2 NFA and \(\epsilon\)-NFA Problems

1. We can show that DFAs are also NFAs (how?), but not the converse (why?).

2. Draw an NFA diagram for the following languages:
   
   (a) Let \(A\) be the language of string over \(\Sigma = \{0, 1\}\) ending in “0001”.
   (b) Let \(B\) be the language of string over \(\Sigma = \{0, 1\}\) ending in either “01” or “10”.
   (c) Let \(C\) be the language of string over \(\Sigma = \{0, 1\}\) starting in “1000”, what did you notice about the NFA you’ve created?

3. Give an \(\epsilon\)-NFA for the language \(C(A \cup B)\).

4. Convert the NFA of language \(B\) to a DFA using subset construction.