Non-determinism

Idea: A non-deterministic finite automaton (NFA) always makes that "right" choice to get to an accept state. If it's possible to accept a string $x$, it accepts $x$.

Question: What are the languages of the following NFA?
Formalizing ε-NFA

**Defn:** An ε-NFA $\mathcal{M}$ consists of:

- A set $Q$ of states
- An alphabet $\Sigma$
- A transition function $\delta : Q \times 2^\Sigma \rightarrow 2^Q$
- A start state $q_0 \in Q$
- A set of accept states $A \subseteq Q$

---

![Diagram of ε-NFA with transitions and states labeled]

- $\delta(q, 0) = \{q, q'\} = \{q''\}$
- $\delta(q, 1) = \emptyset$
- $\varepsilon(q'') = \{q'\}$
- $\varepsilon(q) = \emptyset$
Language of an e-NFA

Define: If $M$ is a DFA, the language of $M$ (written $L(M)$) is the set of strings $M$ accepts.

Define: $M$ accepts $x$ if $q_0 \in A$.

Define: $F : Q \times \Sigma^* \to Q$ given the state $M$ ends in after processing $x$; it is given by

$F(q_x) = q$ and $F(q, x) := \delta(F(q, x), x)$.

Defn: $N$ is an NFA, $L(N)$ is the set of strings $N$ accepts.

Defn: $N$ accepts $x$ if there is an accept state $q$ in $\hat{\delta}((q, x), \epsilon)$.

Defn: For an NFA, the extended transition $\hat{\delta}$.

$\hat{\delta} : Q \times \Sigma^* \to 2^Q$ tells us where $N$ could reach on input $x$.

$\hat{\delta}(q, \epsilon) := \hat{\epsilon}(\{ q \})$

$\hat{\delta}(q, xa) := \hat{\delta}(\hat{\delta}(q, x), a)$

Defn: if $S$ is a set of states,

$\hat{\epsilon}(S)$ is the set of states reachable from $S$ with any path of $\epsilon$-trans

(possibly transitioning).

"$\epsilon$-closure of $S$"

$\hat{\delta}(q, \epsilon) := \hat{\epsilon}(\{ q \})$

$\hat{\delta}(q, xa) := \hat{\epsilon}(\bigcup_{q' \in \hat{\delta}(q, x)} \hat{\delta}(q', a))$
Removing non-determinism

**Question:** Are non-deterministic automata more powerful than deterministic automata?

**Claim:** For all NFA $N$ there exists a DFA $M$ with $L(M) = L(N)$.

**Proof:** Choose an arbitrary $N$. Let $M$ be given as follows:

![Diagram of a DFA and an NFA]