cs302 Theory of Computation UVa Spring 2008

Notes: Nondeterministic Finite Automata

Tuesday, 29 January

Upcoming Schedule

This week: Finish reading Chapter 1

Wednesday, 30 January (9:30-10:30am): Theory Coffee Hours (Wilsdorf Coffee Shop)

Wednesday, 30 January (6-7pm): Problem-Solving Session (Olsson 226D)

Thursday, 7 February: Problem Set 2 is due at the beginning of class. *Problem Set 2 is longer and harder than Problem Set 1. Don't wait to get started on it.*

Definition: A *nondeterministic finite automaton* (NFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0, and F are defined as they are for a DFA, and δ is defined as:

δ**:** $Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$ —- a function from a state and alphabet symbol to **a set of states** that is a member of $\mathcal{P}(Q)$, the power set of Q.

Example 1: Draw a NFA that recognizes the language of all strings in $\{0, 1\}^*$ that start and end with the same symbol.

Computation Model for an NFA

The NFA $A = (Q, \Sigma, \delta, q_0, F)$ accepts the language $\mathcal{L}(A) = \{w | \delta^*(q_0, w) \cap F \neq \emptyset\}$ where δ^*: $Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$ is the *extended transition function* defined by:

$\delta^*(q, \epsilon) = \{q\}$

For $w = ax$ where $a \in \Sigma$ and $x \in \Sigma^*$, $\delta^*(q, ax) = \bigcup_{q_i \in \delta(q,a)} \delta^*(q_i, x)$

Equivalence of NFAs and DFAs. Show that the set of languages that can be recognized by some NFA is equal to the set of languages that can be recognized by some DFA.

1. Every DFA has an equivalent NFA. (Proof by construction - trivial.)
2. Every NFA has an equivalent DFA. (Proof by construction below, and in book, Theorem 1.39)

Given $N = (Q, \Sigma, \delta, q_0, F)$, and NFA recognizing some language A. We prove that every NFA has an equivalent DFA by showing how to construct a DFA N' from N that recognizes the same language A. $N' = (Q', \Sigma', \delta', q_0', F')$ defined as:

1. $Q' = \mathcal{P}(Q)$ — we have a state in Q' to represent each possible subset of states in Q. The label for each state in Q' is a set.
2. $\Sigma' = \Sigma$ — the alphabet is the same
3. $\delta' : Q' \times \Sigma' \to Q'$ is defined to capture all possible states resulting from δ transitioning from the input state: $\delta'(R, a) = \bigcup_{r \in R} E(\delta(r, a))$
4. $q_0' = E(q_0)$
5. $F' = \{$states in Q' that correspond to subsets of states that include a state in $F\}$

where $E : Q' \to Q'$: is the epsilon-transition function defined by:

$$E(q) = q \cup \bigcup_{r \in \delta(q, \epsilon)} E(r)$$

Convert the NFA from Example 1 into a DFA.

Suppose language A can be recognized by an NFA with n states. What can we say about the number of states a DFA that recognizes A must have?

Prove that the regular languages are closed under reversal. That is, if L is a regular language, then $L^R = \{w | w^R \in L\}$ is a regular language.