Problem Set 1: Problems 1–4. Due Tuesday, January 17, 2006

Problem 1.

A. A **Moore machine** is like a DFA except that each state is annotated by a character. Unlike a DFA, which computes a predicate on strings, a Moore machine computes a string-valued function on strings. Formally define Moore machines and the function computed by a Moore machine.

B. A **Mealey machine** is like a DFA except that each transition out of a state is annotated by a character. As with a Moore machine, a Mealey machine computes a function on strings. Formally define Mealey machines and the function computed by a Mealey machine.

C. Show that a function f is computable by a Moore machine iff f is computable by a Mealey machine.

Problem 2. We have argued that, up to the naming of states, for any regular language L, there is a unique minimum-state DFA that accepts L. Resolve whether or not the same is true for NFAs: given a regular language L, is there a unique minimum-state NFA for it?

Problem 3. Use the Myhill-Nerode Theorem to give a different proof from the one given in class that L is regular iff L is decidable by a 2-Way Finite Automaton.

Problem 4. *Difficult.* Prove that if $L \subseteq 1^*$ then L^* is regular.