Problem Set 2 — Due Wednesday, January 21, 1998

Problem 1. An MSSDFA (Multiple Start State DFA) is just like an ordinary DFA except that there is a set of possible start states. Informally, a string x is accepted by an MSSDFA M if there is some start state q in M for which M accepts x if M begins in q.

(a) Formally define: what is an MSSDFA M; what it means for an MSSDFA to accept a string x; the language, $L(M)$, of an MSSDFA M; and the class of MSSDFA-acceptable languages.

(b) Prove that L is MSSDFA-acceptable if and only if it is DFA acceptable.

Problem 2. Construct a family of languages L_n, $n \geq 1$, such that there is a DFA M_n having n final states which accepts L_n but there is *no* DFA having $n-1$ final states which accepts L_n. Prove that L_n has this property.

Problem 3. A *Moore Machine* is like a DFA except that we associate an output string with each state. The machine outputs that string when it leaves a state. Thus a Moore machine M computes a function $f_M : \Sigma^* \to \Gamma^*$.

(a) Formally define a Moore machine M and the function f_M which it computes.

(b) Give a Moore machine which adds two binary numbers. The input to the Moore machine is a string $x = a_1 b_1 \cdots a_n b_n \#$, for some $n \geq 1$. (The symbol "#" is just a formal symbol in the alphabet; each a_i and b_i are binary digits.) The output is a string $c_1 \cdots c_{n+1}$ which encodes the sum of $a = a_n \cdots a_1$ and $b = b_n \cdots b_1$. By convention, we feed in the numbers least-significant-bit to most-significant-bit, which is the same order which the machine produces output. Represent your machine in an easy-to-read manner; describe any convention you employ.

Problem 4. Page 85, Exercise 1.12.

Problem 5. Page 89, Problem 1.32.