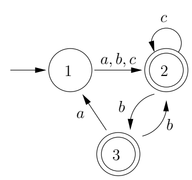
Problem Set 4 — Due Tuesday, January 31, 2006

Several problems below are taken directly from Sipser's Introduction to the Theory of Computation. We have repeated them because not everyone has the same edition of the text.

Problem 1. Use the procedure described both in lecture and in the textbook to convert the following regular expression to nondeterministic finite automata.

$(0 \cup 1)^*000(0 \cup 1)^*$

Problem 2. Use the procedure described both in class and in the textbook to convert the following finite automata to a regular expression.



Problem 3. Consider a new kind of finite automaton called an all-paths-NFA. An all-paths-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that recognizes $x \in \Sigma^*$ if *every* possible computation of M on x ends in a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* computation ends in an accept state. Prove that all-paths-NFA recognize the class of regular languages.

Problem 4. Use the pumping lemma to show that the following languages are not regular.

Part A. $L_a = \{0^m 1^n 0^{m+n} : m, n \geq 0\}$.

Part B. $L_b = \{1^n : n \text{ is prime.}\}$.

Problem 5. Determine if the following languages are or are not regular.

Part A. Let $L = \{w \mid$ contains an equal number of occurences of the substrings 01 and 10$\}$
Thus, $101 \in L$ because 101 contains a single 01 and a single 10, but $1010 \notin L$ because 1010 contains two 10s and one 01. Is L a regular language?

Part B. Let $L = \{x < y : x, y \in \{\texttt{0}, \texttt{1}\}^*$ and the number represented by x, in binary, is less than the number represented by y, in binary.$\}$

Here, "<" is a formal symbol; the alphabet is $\{0, 1, <\}$. The string "$0 < 1$" $\in L$, while the string "$1 < 0$" $\notin L$. Is L regular?