Problem Set 4 — Due February 1, 2005

Problem 1. (pretty hard.) Exhibit a family of languages \{L_n : n \geq 1\} over \Sigma = \{0, 1\} such that \(L_n\) is accepted by an NFA of size \(n + 2\), but \(L_n\) is not accepted by any DFA of size \(2^n - 1\). Prove that your family of languages has this property.

It is fine to solve this problem for different additive constants 2 and 1 (meaning \(n + c\) and \(2^n - d\) is fine, for any constants \(c, d\)).

Problem 2. Consider applying the product construction to NFAs \(M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)\) and \(M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)\) in order to show that the NFA-acceptable languages are closed under intersection.

Part A. Formally specify the product machine \(M = (Q, \Sigma, \delta, q_0, F)\).

Part B. Does the construction work—that is, is \(L(M) = L(M_1) \cap L(M_2)\)? Informally argue your conclusion.

Problem 3. Page 86, Exercise 1.16, part (b).

Problem 4 Imagine converting the regular expression \(\alpha = (00 \cup 001)^*\) into a DFA using the procedures given in class. How many states will the resulting DFA have? Compare this with the size of the smallest DFA that recognizes \(L(\alpha)\).

Problem 5 Give an algorithm (specify it as simply and clearly as you can) that answers the following question: given a regular expression \(\alpha\) over the alphabet \(\{a, b\}\), is \(aba \in L(\alpha)\)? Make your algorithm as efficient as you can, and comment on its running time.