CSE 322
Winter Quarter 2003
Assignment 4
Due Friday, January 31

All solutions should be neatly written or type set. All major steps in proofs and algorithms must be justified.

1. (10 points) In this problem you will practice creating NFAs from regular expressions and removing ϵ-transitions Consider the regular expression $\alpha = (01 \cup 1)^*0$.

   (a) Use the standard construction (see theorem 3.7 on page 102 of the text) to construct an ϵ-NFA that accepts the language $L(\alpha)$.

   (b) Take the result of part (a) and construct an NFA with no ϵ-transitions. To do this, first compute ECLOSE(q) $= \{p : p$ is reachable from q by ϵ transitions$\}$. Use this computation to construct the transition function of the new NFA.

2. (10 points) Regular expressions are formed using union, concatenation, and Kleene star. An alternative form of regular expressions is called *star-free regular exprssions* over Σ which are formed using union, concatenation, and complement. The basis star-free regular expressions are the symbols in Σ and ϕ. More formally, $L(a) = \{a\}$ and $L(\phi) =$ the empty set. If α and β are star-free regular expressions over Σ then so are $\alpha \cup \beta$, $\alpha\beta$, and $\neg\alpha$, where $L(\alpha \cup \beta) =$ the union of $L(\alpha)$ and $L(\beta)$, $L(\alpha\beta) =$ the concatenation of $L(\alpha)$ with $L(\beta)$, and $L(\neg\alpha) = \Sigma^* - L(\alpha)$. For example, the star-free regular expression $\neg\phi$ over the alphabet $\{0, 1\}$ represents the set of all binary strings, that is $L(\neg\phi) = \{0, 1\}^*$. Another example is $(\neg\phi)00(\neg\phi)$ over $\{0, 1\}$ which represents the set of all the binary strings that contain 00.

   (a) Design a star-free regular expression over $\{0, 1\}$ for the language $L = \{(01)^i; i \geq 0\} = \{\epsilon, 01, 0101, 010101, \ldots\}$.

   (b) Show that if L_1 and L_2 are definabled by star-free regular expressions then so is their intersection.

   (c) Show that every finite language is definable by a star-free regular expresson over $\{0, 1\}$.

3. (10 points) For this problem you should design algorithms in the style of problem 3 of the third assignment to decide properties of deterministic finite automata. In particular given a DFA $M = (Q, \Sigma, \delta, q_0, F)$ there is a natural directed graph G_M that models the transitions. The set of vertices of G_M is Q and (q, p) is an edge in G_M if $\delta(q, \sigma) = p$ for some $\sigma \in \Sigma$. The graph G_M is the transition diagram of M with the labels on the edges removed. You should use G_M in your algorithms.

   (a) Design an algorithm to decide whether a DFA accepts any strings at all. That is, the algorithm given a finite automaton M determines if $L(M) = \phi$.

   (b) Design an algorithm to decide whether a DFA accepts infinitely many strings.