Homework 2

Due midnight, Thursday, 2/7/2013

Please submit your homework solutions online at `http://www.dci.pomona.edu/tools-bin/cs081upload.php`. If you have more than one file to be turned in, please put it in a folder and zip it up before turning it in.

Problems from the texts are given in the form c.n where c is the chapter and n is the problem number. Thus problem 2.7 is problem 7 from Chapter 2.

1. (10 points) **Closure**

   Problem 8.7cd from Rich. You may use any of the representations of regular languages discussed in class. Do provide an argument as to why your construction gives all the strings of the desired language and no extras.

2. (10 points) **More Closure**

   In this problem, I want you to show the reverse of problem 8.7d.

   Let Σ, Δ be alphabets and $h : \Sigma \to \Delta^*$. We can extend h to Σ^* as follows:

   $h(\epsilon) = \epsilon$
   $h(wa) = h(w)h(a)$ for any $w \in \Sigma^*, a \in \Sigma$

   Any function $h : \Sigma^* \to \Delta^*$ defined in this way from a function $h : \Sigma \to \Delta$ is called a homomorphism.

   Let $h : \Sigma^* \to \Delta^*$ be a homomorphism. Show that if $L \subset \Delta^*$ is regular (i.e., accepted by a finite state machine) then so is $\{w \in \Sigma^* | h(w) \in L\}$.

   Hint: Start from a deterministic finite state machine M accepting L and construct one with the same states which, when it reads an input symbol a delivers $h(a)$ *in its imagination* to M for processing.

3. (10 points) **Regular Languages**

   Let $\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$.

   The alphabet Σ_3 has eight letters, consisting of all the size-three columns of 0's and 1's. Consider each row to be a binary number, and let

   $$B = \{w \mid \text{the bottom row is the sum of the two upper rows}\}.$$

   For example, $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in B$, but $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \notin B$, Show that B is regular.1 Hint: Working with $B^{\mathcal{R}}$ is easier. You may use the result of problem 1.

4. (10 points) **Regular expressions**

   Let $\Sigma = \{a, b\}$, and let A be the set of words w over Σ for which w contains an even number of a's or an odd number of b's. Give a regular expression for the language A.

5. (20 points) **Minimization algorithm**

   In this problem you will be creating an algorithm for minimizing finite state machines. Let $M = \{K, \Sigma, \delta, s, A\}$ and for each $q \in K$, let $M_q = \{K, \Sigma, \delta, q, A\}$ (that is, a machine that differs from M only by changing the start state). For $k \geq 0$, say states p and q of M are *k-distinguishable* if there is some input string of length at most k which is accepted by one of M_p and M_q, but not the other. Otherwise they are *k-indistinguishable*. Two states are equivalent if they are k-indistinguishable for every k.

   (a) Show that if a deterministic finite state machine has two equivalent states, one can be eliminated; but if all pairs of distinct states are inequivalent, then the machine is as small as possible.

   (b) Show that states p and q are k-indistinguishable if and only if

      i. both are accepting or both are non-accepting, and

      ii. if $k > 0$, then for each $a \in \Sigma$, $\delta(p, a)$ and $\delta(q, a)$ are $(k-1)$-indistinguishable.

   (c) Define a series of equivalence relations $\equiv_0$, $\equiv_1$, $\ldots$ on the states of M as follows:

      - $p \equiv_0 q$ if and only if p and q are accepting or both are non-accepting.
      - $p \equiv_{k+1} q$ if and only if $p \equiv_k q$ and, for each $a \in \Sigma$, $\delta(p, a) \equiv_k \delta(q, a)$.

      Show that $p \equiv_k q$ if and only if p and q are k-indistinguishable.

   (d) Show that there is an $n \leq |K|$ such that $\equiv_n$ and $\equiv_{n+1}$ are identical. Then by the previous part and the definition of equivalent states, all equivalent states can be discovered by carrying through the inductive definition of $\equiv_{k+1}$ at most $|K|$ times. Use part a of this problem to describe an algorithm to minimize a given deterministic finite state machine.

6. (5 points) **Minimizing FSMs**

   Problem 5.12 from Rich

7. (20 points) **Pumping Lemma, etc.**

   Problem 8.1n, s