Computer Science 236 Fall — Nov. 11, 2010
St. George Campus — University of Toronto

Assignment 3
Due Date: 2nd December, 2010

1. (10 marks) Assume that you are given a file of arbitrary length that contains student records for CSC236. Each line of the file has the following format:

   LastName,FirstName,Average

   There are no spaces and commas are the separators on each line. Assume that the only characters in the file are upper case letters, numerals (0-9) and commas and that the file is sorted by decreasing average. You need to find some information out about the students in the class. The easiest way to do this is to write a regular expression that matches the information you need (much like grep in Unix). If possible, construct a regular expression to extract the information. If no such regular expression exists, prove why not. [Note: you may use the notation $\backslash a$ to represent any alphabetic symbol and $\backslash n$ to represent any numeric symbol]

   (a) (2 marks) All those people whose last name begins with A, B, C or D.
   (b) (2 marks) All those people who are passing the course.
   (c) (2 marks) All those people who are failing the course.
   (d) (4 marks) Determine whether there are more people passing the course than failing the course. [Hint: notice that the entries are sorted by from largest to smallest average]

2. (16 marks) For each of the following, give a regular expression that denotes the set of strings as well as a DFA that accepts it.

   (a) $S_1 = \{x \in \{0,1\}^* \mid$ neither 00 nor 11 appear as substrings of $x\}$
   (b) $S_2 = \{x \in \{0,1\}^* \mid$ both 00 and 11 appear as substrings of $x\}$
   (c) $S_3 = \{0^n 1^m \mid n, m \geq 0$ and $n + m$ is odd $\}$
   (d) $S_4 = \{x \in \{a,b,c\}^* \mid x$ starts and ends with the same letter$\}$.

3. (12 marks) Consider a set S of strings over the alphabet $\Sigma = \{0,1\}$.

   (a) Construct a DFA that accepts $s \in S$ iff the decimal value of s is divisible by 5. For example, your DFA should accept 0, 101, 1010 and 1111 but it should not accept 10, 1001, 1011 etc.
   (b) Use induction to prove that your DFA is correct.

4. (6 marks) Show that if A is a regular language then A^R is as well, i.e., that the class of regular languages is closed under reversal.

5. (6 marks) A *palindrome* is a word that can be read forwards and backwards. For example, “bob”, “anna”, “kayak”, “radar”. Let Σ be the alphabet of lower case English letters. Consider the set P of *palindromes* from Σ^*. Let $\mathcal{L}$ be the language consisting of the set of strings in P. Determine whether $\mathcal{L}$ is regular and prove your claim either by giving a description of how to build a DFA to accept $\mathcal{L}$ or by constructing a contradiction.

6. (15 marks)

   (a) (5 marks) Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. Prove that $f \in O(g) \Leftrightarrow g \in \Omega(f)$.
   (That is, prove $f \in O(g) \Rightarrow g \in \Omega(f)$ and prove $g \in \Omega(f) \Rightarrow f \in O(g)$.)

   (b) (5 marks) Prove or disprove the following Conjecture.
   Conjecture: For every $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, if $f \in O(g)$, then $|f - g| \in O(g)$.
   (We define $|f - g|$ to be that function that maps n to $|f(n) - g(n)|$.)

   (c) (5 marks) Define $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$ by $f(n) = 100n^2 + 5n + 10$ and $g(n) =| n^3 - 100n^2 |$.
   Prove that $f \in O(g)$ by exhibiting particular constants c and n_0, and proving that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.

7. (10 marks) Consider the following recurrence for a function T that takes on nonnegative values and is defined on integers ≥ 1:

$$T(n) \leq \begin{cases} 5 & \text{if } 1 \leq n \leq 4 \\ T(\lfloor \frac{7}{10}n \rfloor) + T(\lfloor \frac{1}{5}n \rfloor) + 3n & \text{if } n > 4 \end{cases}$$

   Prove that $T(n)$ is $O(n)$. Note that it doesn't help to use the general theorem about divide-and-conquer recurrences. You should present a particular constant c and prove that $T(n) \leq c \cdot n$ for all $n \geq 1$.

   Motivation: This recurrence actually comes up in the following situation. Say we wish to find the median of n distinct elements, using as few comparisons as possible. More generally, say we wish to find the k-th smallest of n elements. We could sort the n numbers but this takes about $n \log n$ comparisons. Instead we do the following which, by the theorem proven in this question, uses $O(n)$ comparisons:

   Divide the elements up into groups of 5 (don't worry now about what to do if n isn't divisible by 5); find the median of each group of 5, using a *linear* in n number of comparisons in total. This gives us about $n/5$ median points. Recursively find the median of these median points using about $T(n/5)$ comparisons; call this element b. Using a *linear* in n number of comparisons, compare every element to b; let S_1 be the set of elements $\leq b$, and let S_2 be the set of elements $> b$. Because of the way b was chosen, each of S_1 and S_2 must have at most $\frac{7}{10}n$ elements. If $k \leq |S_1|$, then recursively find the k-th smallest element of S_1, otherwise recursively find the $(k - |S_1|)$-th smallest element of S_2; this takes at most $T(\frac{7}{10}n)$ comparisons.