Assignment 1 **CAS705 Fall 2005** **Due on September 26**

The work you submit must be your own. *You may discuss problems with each other; however, you should prepare written solutions alone. In particular, you should not leave with any written notes from such discussions. The style and clarity of your answers will be an important factor in the grade.*

Each question is worth 25%.

1. Present a linear-time time, two-string TM for deciding the language of palindromes (using definition 2.1 of TMs in the textbook). (This will be the last time in this course that you have to give a formal description of a TM.)

2. In this exercise you are asked to "fill in some blanks" in the proof of the lower bound for palindromes presented in class.

   Lemma: Let M be a one-string TM that decides $L_{\text{pal}} = \{x \in \{0,1\}^* | x = x^R\}$. Then, M requires $\Omega(n^2)$ many steps.

   Proof: We follow the outline in exercise 2.8.5 in Papadimitriou.

   We first define the *i-th crossing sequence of M on x* to be

   $$\{(q_1, \sigma_1), (q_2, \sigma_2), \ldots, (q_m, \sigma_m)\},$$

   and it means that the first time M crosses from square i to square $i+1$, M leaves σ_1 on the i-th square, and is in state q_1, and then the first time it crosses from square $i+1$ to square i, it arrives at square i in state q_2 and encounters σ_2 on the i-th square, etc.

   Note that the odd pairs denote crossings left-to-right, and even pairs denote crossings right-to-left.

   (a) Why $\sigma_{2i} = \sigma_{2i-1}$?

   Now consider inputs of the form $x0^n x^R$, where $|x| = n$, and x^R means "x in reverse." Let $T_M(x)$ be the number of steps that M takes to decide (and accept) $x0^n x^R$.

   There must be some i, $n \leq i \leq 2n$, i.e., some square in 0^n, for which the i-th crossing sequence has length $m \leq \frac{T_M(x)}{n}$.

   (b) Why must there be such a square?

   Claim: M, i, n, S describe x uniquely.

   To see this, we show how we can "extract" x from M, i, n, S. For all $x \in \{0,1\}^n$, we simulate M on $x0^{i-n}$. The first time M crosses from square i to square $i+1$ we check that it does so with the pair (q_1, σ_1), and we immediately reset the state to q_2 and put the head back on square i and continue. The second time M crosses from square i to square $i+1$ we again check that it does so with (q_3, σ_3), and again we immediately reset the state to q_4 and put the head back on square i and continue. If all the even crossing pairs are correct, we know that we have x.

   (c) Suppose that we find an x for which the crossing sequence matches; how do we know this is in fact the x that we want?

One minor technical point is that we must assume that M returns to $\triangleright$ before accepting—this adds only a linear number of steps to the computation.

(d) Why do we need to make this "minor technical point"?

So M, i, n, S describe x uniquely, and they can be encoded with

$$c_M + \log(n) + \log(n) + c \cdot m \leq c_m + 2\log(n) + c \cdot \frac{T_M(x)}{n}$$

many bits. However, for every n, no matter how we encode strings, there must be a string x_0 whose encoding requires at least n many bits.

(e) Why must there be such an x_0?

Therefore, we have that

$$n \leq c_m + 2\log(n) + c \cdot \frac{T_M(x_0)}{n}.$$

(f) Using the definition of Ω, show that the line above implies the lower bound $\Omega(n^2)$.

3. Let **P** be the class of languages decidable in polynomial time. Show that **P** is *robust* with respect to the number of strings (tapes) in the underlying TMs.

4. Let **LinearTime** be the class of languages decidable in linear time (i.e., $O(n)$). From the previous questions in this assignment we know that **LinearTime** is not robust (why?). However, **NLinearTime** (nondeterministic linear time) is more robust. Show that if M_k^{nlt} is a nondeterministic linear time bounded TM with k strings, then M_k^{nlt} can be simulated by M_2^{nlt}. (Assume that the input is always written on string 1.)