Homework 1
Due on 21st Feb, 2008

INTRODUCTION TO COMPUTATIONAL COMPLEXITY THEORY
(CSCE 424/824, SPRING 2008)
URL: `http://www.cse.unl.edu/~vinod/824s08/index.html`

On problems which ask you to design Turing machines you need to first explain your design in English first, and then give the transition-table with explanation as to what each state stands for. To make the table compact, you may omit those transitions which the TM will never make. Answers should be presented clearly and legibly. You are strongly advised to use some document preparation system for typesetting the answers.

1. (15 points) Design a (possibly multi-tape) TM which takes input the binary representations of n and m and outputs binary representation of $n+m$. Assume that the inputs n and m are separated by a special symbol #.

2. (15 points) Show that the language $\{0^n 1^n \mid n \geq 1\}$ is in the complexity class TIME(n) $\cap$ LOGSPACE.

3. **Padding**. Padding (or translation) is a very useful technique for establishing relations among complexity classes. This homework exercise will demonstrate the power of the padding technique.

   (a) (10 Points) Let $f : \mathbf{N} \to \mathbf{N}$ be a proper function with $f(n) \geq n$ and $L \in$ TIME($f(n)$). Consider the following padded language $L_f = \{1^{f(|x|)}0x | x \in L\}$. Show that $L_f \in$ TIME(n).

   (b) (15 Points) The time-hierarchy theorem that we proved in the class is actually very weak. For instance we cannot conclude TIME(n)$\subsetneq$TIME(n^2) from it. But we can prove this using padding.

      i. Show that if TIME(n^α) = TIME($n^{\alpha+\epsilon}$) for a rational number α and a small rational number ϵ, then in fact TIME(n^α) = TIME($n^{\alpha+k\epsilon}$) for all positive integers k.

      ii. Use the above result to show that for any rational α, β where $1 \leq \alpha \leq \beta$ TIME(n^α)$\subsetneq$TIME(n^β).

4. **Canonical Complete Languages**. For many complexity classes, there is a canonical way to define a complete language. These languages may not be natural like other real-life problems, but are enough to show the existence of complete languages.

   (a) (10 Points) For a Turing machine M, let $\langle M \rangle$ denotes the encoding of M. Consider the languages $L_{\text{NP}} = \{\langle M \rangle; x; 0^n | M$ is a nondeterministic TM and M accepts x in n steps $\}$. Show that L_{NP} is NP-complete.

   (b) (10 Points) Give machine based languages, similar to the above, which are complete for the classes EXP and PSPACE respectively.

5. **Closure under reductions**. A complexity class $\mathcal{C}$ is said to be closed under reductions if whenever a language $L \in \mathcal{C}$ and $L' \leq L$, then $L' \in \mathcal{C}$.

   (a) (5 Points) Let $\mathcal{C}' \subseteq \mathcal{C}$ are two complexity classes which have complete languages, and $\mathcal{C}'$ is closed under reductions. Show that $\mathcal{C}' = \mathcal{C}$ if and only if $\mathcal{C}'$ contains a $\mathcal{C}$-complete language

   (b) (10 Points) Show that the complexity class NP is closed under reductions but the complexity class E is not (E denotes the time complexity class $\bigcup_{k \geq 1}$ TIME(2^{kn}). Note that this is different from EXP). Hence conclude that NP $\neq$ E.

   (c) (10 Points) Let POLYLOG denote the space complexity class $\bigcup_{k \geq 1}$ SPACE($(\log n)^k$). Show that there are no POLYLOG-complete languages.