CS381 Fall 2001
Homework 7
Prof Shai Ben-David
Due: Friday, November 30, 2001 — 9:05 am

**FINAL VERSION

1. For each of the following languages, find out if it is recursive or not. If it is, describe a total Turing machine that computes it. If not, explain why the existence of such a machine entails a contradiction.

   (a) $\{0^{n_0}1^{n_1}0^{n_2}\ldots 1^{n_{2k}}$: for all $0 \leq i \leq 2k,\ n_i \in \mathbb{N}$ and n_0 is a solution to the equation $n_1 \cdot x^{n_2} + n_3x^{n_4} + n_{2k-1}x^{n_{2k}} = 0\}$

   (b) $\{\#M : \epsilon \in L(M)\}$

   (c) $\{\#M : |L(M)| < 100\}$ (that is, M accepts less than 100 strings)

   (d) $\{(\#M_1, \#M_2) : L(M_1) = L(M_2)\}$ (where M_1, M_2 are Turing machines)

2. (a) Prove that if one changes the definition of Turing machines to allow an infinite set of states Q, then for every $L \subseteq \{0\}^\star$ there exists a total machine that computes it.

   (b) Prove that there exists a language $L \subseteq \{0\}^\star$ such that for every Turing machine (under the usual definition), $L(T) \neq L$.

3. Prove that the family of all recursive languages is closed under the * operation. Namely, if L is recursive then so is $L^\star = \{w_1 \ldots w_n : n \in \mathbb{N}$ and for all $i,\ w_i \in L\}$.

4. Given a Turing machine T, let $\overline{T}$ denote the machine obtained by switching the 'r' and 't' states of T. That is, the transition function of $\overline{T}$, $\overline{\delta}$, is obtained by replacing each occurence of t in the function δ of T by an r and vice versa. Prove or refute each of the following claims:

   (a) For every $T, L(\overline{T}) = \overline{L(T)}$ (where $\overline{L}$ is the complement of a language L).

   (b) For every pair of machines T_1, T_2, if $L(T_1) = L(T_2)$ then $L(\overline{T}_1) = L(\overline{T}_2)$.

   (c) For every machine T, if $L(T)$ is recursive, then so is $L(\overline{T})$.