Homework 5

Definition

A Turing machine *transducer* is a Turing machine that outputs strings, rather than accepting or rejecting. That is, if M is a Turing machine, then M computes the function which takes an input x to the value that is left on the tape when M halts. If M does not halt on input x, then the function computed by M is undefined on x. A function f that is computed by a Turing machine transducer is called *partial computable* ("partial," since M may not halt on some inputs, and then f is undefined on those inputs).

Homework Questions

1. In class/the book, you were given the following definition of "recursively enumerable:"

> A language L is recursively enumerable if there is a TM M such that M accepts exactly those strings in L. (On strings not in L, M may either reject or not halt.)

Show that each of the following conditions is equivalent to being L recursively enumerable (in the above sense):

(a) There is a partial computable function f such that L is the range of f, that is, $L = \{f(x) : x \in \Sigma^*\}$. Such an f is called an *enumerator* of L, since when f is run on all strings in succession, it enumerates/lists the elements of L.

(b) There is a partial computable function f such that L is the domain of f, that is, $L = \{x : f(x) \text{ is defined }\}$. Equivalently, there is a TM M such that $L = \{x : M \text{ halts on input } x\}$.

2.

(a) Show the recursive analog of 1(a): L is recursive if and only if there is an enumerator f for L such that f is *monotone*, that is, $x < y$ implies $f(x) < f(y)$ (thinking of x, y as natural numbers). **In the definition of monotonicity for partial functions, treat undefined values of $f(x)$ as $f(x) = +\infty$.**

(b) Show that every infinite recursively enumerable language L contains an infinite subset $L' \subseteq L$ such that L' is recursive. *Hint:* this is not unrelated to 1(a) and 2(a).

3. Consider the following two languages:

$$Inf = \{M | L(M) \text{ is infinite}\},$$

and

$$Tot = \{M | M \text{ is a total TM, that is } M \text{ halts on all inputs}\}.$$

(a) Give a reduction from Inf to Tot.

(b) Give a reduction from Tot to Inf.

(c) Using the generalized/second Rice's Theorem (a.k.a. Rice-Shapiro Theorem), show that neither Inf nor its complement $\overline{Inf}$ are recursively enumerable.

(d) Using 3(c) and the reductions from the earlier part of this problem, show that neither Tot is recursively enumerable.

(e) Using the Rice-Shapiro Theorem, show that neither Tot nor its complement $\overline{Tot}$ are recursively enumerable. Do this using the Rice-Shapiro Theorem directly (rather than as in 3(d)).