Assignment 4: Parameter Theorem, R.E. Sets, Reducibility, Rice-Shapiro Theorem

This assignment is due Friday, March 12th, at the beginning of class (9:00am).

1. Suppose that $f : \mathcal{N} \to \mathcal{N}$ is a strictly increasing function: in other words, $f(n+1) > f(n)$ for all $n \in \mathcal{N}$. Prove that the set

$$B = \{f(n) : n \in \mathcal{N}\}$$

is recursive.

Consider the following program:

(A) IF $f(Z) = X$ GOTO B
IF $f(Z) > X$ GOTO E
$Z \leftarrow Z + 1$
GOTO A
(B) $Y \leftarrow Y + 1$
(E)

The program runs through every value of Z. If $f(Z) < X$, it continues running. If $f(Z) = X$, it halts with value 1. If $f(Z) > X$, then for no higher value of Z will $X = f(Z)$ (since f is strictly increasing), so it halts with value 0. Thus, the program computes P_B, so B is recursive.

2. Show that every infinite recursively enumerable set B has an infinite subset $B' \subseteq B$ which is recursive (hint: use the previous question).

Since B is recursively enumerable, there exists a primitive recursive function f so that $B = \{f(n) : n \in \mathcal{N}\}$. Consider the function g defined by the primitive recursion equations:

$$g(0) = f(0)$$

$$g(t+1) = f[\min_z(f(z) > f(t))]$$

Since g is μ-recursive, it is partially computable. However, note that the minimization always exists: if it didn't, there would be some value $f(n)$ so that no element of B is higher than $f(n)$: but this is impossible, since B is infinite. Thus, the function g is actually computable, not just partially computable. Moreover, the definition requires that $g(t+1) > g(t)$, so g is increasing. Let $B' = \{g(n) : n \in \mathcal{N}\}$. Then by the previous question, B' is recursive, since g is a strictly increasing computable function. Finally, since $g(t)$ is always $f(z)$ for some z, $g(t)$ is always in B, so B' is an infinite subset of B, as required.

3. Suppose A and B are subsets of $\mathcal{N}$. Prove the following properties of many-one reducibility:

   (a) $A \leq_m B$ if and only if $\bar{A} \leq_m \bar{B}$;

   (b) if A and B are m-complete, then $A \equiv_m B$;

   (c) if A is m-complete, then A is not recursive.

   (a) If $A \leq_m B$, then there exists a computable f so that $x \in A \Leftrightarrow f(x) \in B$. But this means that $x \notin A \Leftrightarrow f(x) \notin B$. So, using the same f, $\bar{A} \leq_m \bar{B}$. Similarly, if $\bar{A} \leq_m \bar{B}$, we can use the same f to show $A \leq_m B$.

   (b) Since A is m-complete and B is RE., $B \leq_m A$. Similarly, since B is m-complete and A is RE, $A \leq_m B$. Thus $A \equiv_m B$.

   (c) Since K is RE and A is m-complete, $K \leq_m A$. But K is not recursive, so A is not recursive either.

4. Let INF $= \{x \in \mathcal{N} : W_x \text{ is infinite}\}$. Show that TOT $\equiv_m$ INF.

   To show TOT $\equiv_m$ INF, we need to show that TOT $\leq_m$ INF, and INF $\leq_M$ TOT. We begin with TOT $\leq_m$ INF. To show this, we want to find a program with some number q such that $\Phi_q(x, p)$ is defined for infinitely many x if and only if $\Phi_p(x)$ is defined for all x. We can then use the parameter theorem to get a computable function that shows TOT $\leq_m$ INF. We define the program Q by:

   (A) IF $Z > X_1$ GOTO E
   $Y \leftarrow \Phi(Z, X_2)$
   $Z \leftarrow Z + 1$
   GOTO A
   (E)

   Let q be the number of this program. This program terminates exactly when the program with number X_2 is defined for every $Z \leq X_1$. Now, suppose that the program with number p is total. Then for any x, $\Phi_Q(x, p)$ is defined. Conversely, suppose the program with number p is not total. Then it is undefined at some x_0. Then for any $x \geq x_0$, the program Q will not halt, so $\Phi_q(x, p)$ is not defined for infinitely many x (all $x \geq x_0$). Thus, we have that $\Phi_q(x, p)$ is defined for infinitely many x if and only if $\Phi_p(x)$ is total. Let q be the number of the program Q. We then have:

$$
\begin{aligned}
p \in \text{TOT} &\Leftrightarrow \Phi_p(x) \text{ defined for all } x \\
&\Leftrightarrow \Phi_q(x, p) \text{ defined for infinitely many } x \text{ (by the argument above)} \\
&\Leftrightarrow \Phi(x, p, q) \text{ defined for infinitely many } x \\
&\Leftrightarrow \Phi(x, S_1^1(p, q)) \text{ defined for infinitely many } x \text{ (using the parameter theorem)} \\
&\Leftrightarrow S_1^1(p, q) \in \text{INF}.
\end{aligned}
$$

So if we let $f(p) = S_1^1(p, q)$, then f is computable by the parameter theorem, and $p \in \text{TOT} \Leftrightarrow f(p) \in \text{INF}$. Thus, $\text{TOT} \leq_m \text{INF}$, as required.

To show $\text{INF} \leq_M \text{TOT}$, we need to find the opposite: a program with number q so that $\Phi_q(x, p)$ is defined for all x if and only if $\Phi_p(x)$ is defined for infinitely many x. To do this, we need a program that keeps track of how many values a program p is defined for. However, we cannot do this directly, in a computable way. Instead, we can only keep track of which numbers have halted after a certain number of steps. Thus, we begin by defining a primitive recursive function H:

$$H(n, p) = \sum_{i=0}^{n} \text{STP}(i, n, p)$$

For a program with number p, $H(n, p)$ determines how many numbers less than or equal to n the program p halts with after n or fewer steps. We then define a μ-recursive function g:

$$g(x, p) = \min_z H(z, p) > H(x, p)$$

For a program with number p, $g(x, p)$ finds the next time that the program p halts with some number z by z steps.

Since g is μ-recursive, it is partially computable, so it is represented by some program with number q. We claim that $\Phi_q(x, p)$ is defined for all x if and only if $\Phi_p(x)$ is defined for infinitely many x. Indeed, suppose the program with number p is defined for infinitely many x. We need to show that for any x, $g(x, p)$ is defined. Since p is defined for infinitely many x, there exists a $y > x$ so that p halts on input y. Say it halts after t steps. Then define $z = \max(y, t)$. Then $f(z, p) > f(x, p)$, since $z > x$ and p halts after z or fewer steps on input z. Thus, the minimization for g does exist, so $g(x, p)$ is defined.

Conversely, suppose that the program with number p is only defined for finitely many x, say for values $x_0 \cdots x_n$, which halt after $t_0 \cdots t_n$ steps. Let $x = \max(x_0 \cdots x_n, t_0 \cdots t_n)$. Then for every i, program p halts after x or fewer steps on each x_i. So $f(x, p) = n$. But then $g(x, p)$ is undefined, since there is no z with $f(z, p) > n$ (p is only defined for n values). Thus, $\Phi_q(x, p)$ is not defined for all x.

So, we have:

$$\begin{aligned} p \in \text{INF} &\Leftrightarrow \Phi_p(x) \text{ defined for infinitely many } x \\ &\Leftrightarrow \Phi_q(x, p) \text{ defined for all } x \text{ (by the argument above)} \\ &\Leftrightarrow \Phi(x, p, q) \text{ defined for all } x \end{aligned}$$

$\Leftrightarrow \quad \Phi(x, S_1^1(p,q))$ defined for all many x (using the parameter theorem)
$\Leftrightarrow \quad S_1^1(p,q) \in$ TOT.

So if we let $f(p) = S_1^1(p,q)$, then f is computable by the parameter theorem, and $p \in$ INF $\Leftrightarrow f(p) \in$ TOT. Thus, INF $\leq_m$ TOT, as required.

5. Show that each of the following sets are not recursively enumerable by using the Rice-Shapiro Theorem:

   (a) INF $= \{x \in \mathcal{N} : W_x$ is infinite$\}$;

   (b) FIN $= \{x \in \mathcal{N} : W_x$ is finite$\}$;

   (c) PREDICATE $= \{x \in \mathcal{N} : \Phi_x$ is a predicate$\}$.

   (a) Suppose that INF was recursively enumerable. Let A be the set of partially computable functions which are defined for infinitely many values. Then by the Rice-Shapiro theorem, if $f \in A$, there exists a finite function θ, $\theta \leq f$, and $\theta \in A$. But this is a contradiction, since a finite function cannot be defined for infinitely many values.

   (b) Suppose that FIN was recursively enumerable. Let A be the set of partially computable functions which are defined for finitely many values. In particular, the empty function n (undefined for all values) is in A. The function $f(x) = x$ has the property that $n \leq f$. Then by the Rice-Shapiro theorem, since $n \leq f$ and $n \in A$, $f \in A$. But this is a contradiction, since f is defined for all values, but A only consists of functions defined for finitely many values.

   (c) Suppose that PREDICATE was recusively enumerable. Let A be the set of partially computable functions which are predicates. By the Rice-Shapiro theorem, for any $f \in A$, there exists a finite function θ, $\theta \leq f$, and $\theta \in A$. But this is a contradiction, since a finite function cannot be a predicate (which are always total).