Problem Set 4
Some Comments on Homework 6

Institute for Logic, Language and Computation
Universiteit van Amsterdam

First some comments and definitions. Let $f : \mathbb{N} \to \mathbb{N}$ be any function (partial or total). Recall that $f(x) \downarrow$ means that f is defined on x, i.e., $f(x) \in \mathbb{N}$. Define the following sets (called the domain and range of f respectively), $\mathsf{dom}(f) = \{x \mid f(x) \downarrow\}$ and $\mathsf{ran}(f) = \{f(x) \mid f(x) \downarrow\}$. Now f is **p.c.** provided f is can be generated from the partial recursion operations and the μ operator. The function f is **recursive** if f is total and p.c..

If f is recursive, then (using the Church-Turing Thesis) there is an algorithm (Turing machine, URM machine, etc.), call it F, such that on input n, F halts and outputs $f(n)$. Of course, f being recursive does not, in general, mean that $\mathsf{ran}(f)$ is recursive. We have $y \in \mathsf{ran}(f)$ provided there is an n such that $f(n) \downarrow$ and $y = f(n)$. The algorithm F can be used to enumerate all such n, but cannot be used to determine for any given y whether or not $y \in \mathsf{ran}(f)$. This is, in part, why we say $\mathsf{ran}(f)$ is a **c.e.** set.

First of all, recall that a set $A \subseteq \mathbb{N}$ is **recursive** provided χ_A (the charactersitic function of A) is a recursive function.

Definition 1 *g dominates f if for some $n_0 \in \mathbb{N}$, if $n > n_0$ then $g(n) \geq f(n)$*

We will make use of the following lemma. Recall that a function $f : \mathbb{N} \to \mathbb{N}$ is increasing if $f(n) < f(n+1)$ for all $n \in \mathbb{N}$.

Lemma 2 (Cooper, Example 5.1.9, pg. 72) *A is recursive iff it is the range of an increasing recursive function.*

Lemma 3 *Suppose that $g : \mathbb{N} \to \mathbb{N}$ is a recursive 1-1 function and $\mathsf{ran}(g) = \{g(x) \mid x \in \mathbb{N}\}$ is recursive. Then the range of any recursive function f that dominates g is recursive.*

Proof Suppose that $g : \mathbb{N} \to \mathbb{N}$ is a recursive 1-1 function and $\mathsf{ran}(g) = \{f(x) \mid x \in \mathbb{N}\}$ is recursive. Suppose that f dominates g. We must show $\mathsf{ran}(f)$ is recursive. Since f dominates g, there is an $n_0 \in \mathbb{N}$ such that for each $n > n_0$, $f(n) \geq g(n)$. Since $\mathsf{ran}(g)$ is recursive, there is a recursive increasing function r such that $\mathsf{ran}(g) = \mathsf{ran}(r)$. Intuitively, this means that $\mathsf{ran}(g)$ can be enumerated in increasing order. Define the following sets. Let $N = \{0, 1, \ldots, n_0\}$ and for each

$y \in \mathbb{N}$, $G_y = \{y' \mid y' \leq y \text{ and } y' \in \mathsf{ran}(r)\}$. Note that for each $y \in \mathbb{N}$, G_y is finite (this follows from the fact that r is increasing). Now $g^{-1}[G_y] = \{x \mid g(x) \in G_y\}$ is also recursive. This follows since g is 1-1.[1] Both N and $g^{-1}[G_y]$ are finite computable sets and since f is recursive, $f[N \cup g^{-1}[G_y]] = \{f(x) \mid x \in N \cup g^{-1}[G_y]\}$ is also recursive (the algorithm is obvious). For each $y \in \mathbb{N}$ say $F_y = f[N \cup g^{-1}[G_y]]$. Now, for each $y \in \mathbb{N}$, there is an algorithm to construct F_y (the algorithm is evident from the above discussion). The result will follow from the following claim:

Claim *For each $y \in \mathbb{N}$, $y \in F_y$ iff $y \in \mathsf{ran}(f)$.*

Proof (of claim) The left to right direction is obvious. Suppose that $y \in \mathsf{ran}(f)$ and $y \notin F_y$. Then there is an $x \in \mathbb{N}$ such that $f(x) = y$. Since $y \notin F_y$, $x \notin N$, so $x > n_0$. Furthermore since, $y \notin F_y$, $x \notin g^{-1}[G_y] = \{x' \mid g(x') \leq y\}$. Note that $y \in \mathsf{ran}(g)$, otherwise we can derive a contradiction since $y \in \mathsf{ran}(f)$ and f dominates g . Therefore, $g(x) > y$. Hence, since $x > n_0$ and f dominates g, $f(x) \geq g(x)$. Putting everything together, we have $y = f(x) \geq g(x) > y$, which is a contradiction.

□ (of claim)

Thus the above claims show that for each $n \in \mathbb{N}$

$$\chi_{\mathsf{ran}(f)}(n) = \begin{cases} 1 & n \in F_n \\ 0 & n \notin F_n \end{cases}$$

Since F_n can be compute for each n and $n \in F_n$ is recursive, $\mathsf{ran}(f)$ is recursive.

□

[1]The following algorithm will compute this set. Let P_g be the algorithm that computes g. For each $y' \in G_y$, run P_g on input $0, 1, \ldots$ until P_g outputs y'. Since $y' \in \mathsf{ran}(g)$ we know that P_g will eventually halt on an x where $g(x) = y'$. Furthermore, since g is 1-1, such an x is the **unique** x such that $g(x) = y'$.