Math 29: Computability Theory **Spring 2024**

PSET 3 — 04/19/2024

Prof. Miller ***Student: Amittai Siavava***

Problem 1.

Is there a fastest growing total computable function, i.e. a function $f : \mathbb{N} \to \mathbb{N}$ such that

$$f(n+1) - f(n) \geq \varphi_e(n+1) - \varphi_e(n)$$

for all e such that φ_e is total and all n? Prove that your answer is correct.

No, there is no fastest growing total computable function.

Proof. Suppose, for the sake of contradiction, that there exists such a fastest growing total computable function f. This implies:

1. f is total computable.

2. For all e such that φ_e is total computable and for all $n \in \mathbb{N}$, $f(n+1) - f(n) \geq \varphi_e(n+1) - \varphi_e(n)$.

We'll show that for any such f, we can construct a new total computable function g that grows faster than f. Define g as follows:

$$\begin{aligned} g &: \mathbb{N} \to \mathbb{N} \\ n &\mapsto f(n)^2 \end{aligned}$$

1. g is total computable because f is total computable and squaring a natural number is total computable, so their composition is also total computable.

2. We claim that g grows faster than f for all $n \in \mathbb{N}$.

Given the assumption that f is the fastest growing function, we see that $f(n+1) - f(n) \geq 2$ since it has to outgrow the total computable function $\textbf{double}(n) = 2n$.

Since $f(n) \geq 0$ for all $n \in \mathbb{N}$, this implies that $f(n+1) + f(n) \geq 2$. Therefore:

$$\begin{aligned} g(n+1) - g(n) &= f(n+1)^2 - f(n)^2 \\ &= (f(n+1) + f(n))(f(n+1) - f(n)) \\ &\geq 2\,(f(n+1) - f(n)) \\ &> f(n+1) - f(n) \end{aligned}$$

Since g is total computable there exists some e such that $\varphi_e(n+1) - \varphi_e(n) > f(n+1) - f(n)$ for all $n \in \mathbb{N}$. This contradicts the assumption that f was the fastest growing function. $\square$

Problem 2.

Use a diagonalization argument to show that there is no uniform listing of all characteristic functions of the computable sets.

Suppose, for the sake of contradiction, that there is a uniform listing of all characteristic functions of the computable sets. Let φ_e be the e-th characteristic function in this listing. We will construct a new characteristic function φ that is not in this listing.

Define φ as follows:

$$\varphi(n) = \begin{cases} 1 & \text{if } \varphi_n(n) = 0 \\ 0 & \text{otherwise} \end{cases}$$

We claim that φ is a characteristic function of a computable set.

- φ is total computable because it is defined by a finite number of operations.
- φ is characteristic because it is defined in terms of the characteristic function φ_n.

We will show that φ is not in the listing. Suppose, for the sake of contradiction, that $\varphi = \varphi_e$ for some e. By definition of φ, we have:

$$\varphi(e) = 1 \leftrightarrow \varphi_e(e) = 0$$
$$\varphi(e) = 0 \leftrightarrow \varphi_e(e) = 1$$

This means that φ and φ_e disagree at e. Therefore, $\varphi \neq \varphi_e$, contradicting the assumption that $\varphi = \varphi_e$. Therefore, φ is not in the listing, meaning there is no uniform listing of all characteristic functions of the computable sets since we can always, find a characteristic function not included in the listing.

Problem 3.

Prove that a set X is computable *if and only if* it and its complement (that is, $\{n : n \notin X\}$) are c.e.

Definition 3.1. A set X is computable if and only if its characteristic function $\chi_X : \mathbb{N} \to \{0, 1\}$ or its principal function $\chi_X : \mathbb{N} \to \{0, 1\}$ is Turing computable.

Definition 3.2. A set X is said to be *computably enumerable* (or c.e.) if it is the domain of φ_e for some e. Then we write $X = W_e$.

We shall show that X is computable if and only if X and its complement are computably enumerable (c.e.) in two steps; first, by showing that if X is computable then X and its complement are c.e., then by showing that if X and its complement are c.e. then X is computable.

1. If X is computable then X and X^{C} are c.e.

Proof. Suppose X is computable. Then we can construct a Turing machine M_X that computes $\chi_X : \mathbb{N} \to \{0, 1\}$, the characteristic function of X.

Consider these two universal machines:

TM **1:** Compute $f_1 : n \mapsto \begin{cases} 1 & \text{if } M_X(n) = 1 \\ \uparrow & \text{otherwise.} \end{cases}$

On input n:
Run M_X on n.
if M_X *outputs* 1 **then**
    output 1
else if M_X *outputs* 0 **then**
    diverge

TM 1 halts and outputs 1 on input n when M_X halts and outputs 1 on input n. For all other values of n, it diverges. Thus, $X = W_e$ where e is the index of TM 1.

TM 2: Compute $f_2 : n \mapsto \begin{cases} 1 & \text{if } M_X(n) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$.

On input n:
Run M_X on n.
if M_X *outputs* 1 **then**
    diverge
else if M_X *outputs* 0 **then**
    output 1

TM 2 halts and outputs 1 on input n when M_X halts and outputs 0 on input n. For all other values of n, it diverges. Thus, $X^{\mathsf{C}} = W_{e'}$ where e' is the index of TM 2.

Therefore, X and X^{C} are c.e. □

2. If X and its complement are c.e. then X is computable.

Proof. Suppose X and its complement are c.e. Then $X = W_e$ and $X^{\mathsf{C}} = W_{e'}$ for some e, e'. We will construct a Turing machine M_X that computes $\chi_X : \mathbb{N} \to \{0,1\}$, the characteristic function of X.

TM 3: Compute $\chi_X : n \mapsto \begin{cases} 1 & \text{if } n \in X \\ 0 & \text{otherwise.} \end{cases}$

On input n:
Initialize φ_e and $\varphi_{e'}$ on input n.
loop
    Run φ_e on n for one more incremental step.
    Run $\varphi_{e'}$ on n for incremental step.
    if φ_e ***halts*** **then**
        output 1
    else if $\varphi_{e'}$ ***halts*** **then**
        output 0

□

Note that since X and X^{C} are c.e., for any given input n, either φ_e or $\varphi_{e'}$ will *eventually* halt. Therefore, TM 3 eventually halts and output 1 if $n \in X$, or 0 if $n \notin X$. Therefore, TM 3 computes χ_X, so X is computable.

Problem 4.

Prove that every infinite c.e. set contains an infinite computable subset.

Let X be an infinite c.e. set. This means that $X = W_e$ for some e. Precisely, running φ_e on input $n \in \mathbb{N}$ will *always* halt if $n \in X$, but it may diverge if $n \notin X$.

We will construct an infinite subset of X by listing elements of X in increasing order of their index. Let $X_0 = \varnothing$ and $X_{i+1} = X_i \cup \{n \mid \varphi_e(n) \downarrow\}$.

Problem 5.

Prove that every total computable function has infinitely many fixed points.

We will show that every total computable function has infinitely many fixed points using the Recursion Theorem with Parameters.

Theorem 1. Recursion Theorem with Parameters
Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be a total computable function. Then there exists a total computable function $r : \mathbb{N}^k \to \mathbb{N}$ such that

$$\varphi_{r(x_0,x_1,\ldots,x_l)} = \varphi_{f(r(x_0,x_1,\ldots,x_k),x_0,x_1,\ldots,x_k)}.$$

Definition 5.1. For a function $f : \mathbb{N} \to \mathbb{N}$, e is a *fixed point* of f if

$$\varphi_e = \varphi_{f(e)}.$$

Proof. Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be a total computable function. Pick $r : \mathbb{N}^k \to \mathbb{N}$ per the Recursion Theorem with Parameters, so that for all $x_0, x_1, \ldots, x_{k-1}$,

$$\varphi_{r(x_0,\ldots,x_{k-1})} = \varphi_{f(r(x_0,\ldots,x_{k-1}),x_0,x_1,\ldots,x_{k-1})},$$

and r is injective and total computable.

Since $x_0, \ldots, x_{k-1}$ occur as free parameters, this means that there are infintely many combinations of $x_0, \ldots, x_{k-1}$ that $\varphi_{r(x_0,x_1,\ldots,x_{k-1})} = \varphi_{f(r(x_0,x_1,\ldots,x_{k-1}),x_0,x_1,\ldots,x_{k-1})}$, hence f has infinitely many fixed points. $\square$