Math 29: Computability Theory **Spring 2024**

Mid-Term Exam 05/06/2024

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

Problem 1.

Give a register machine which, when given a natural number n in R_0 at ***start***, halts and outputs $x+1$ in R_1 if there is some natural x satisfying $x^2+n^2=2nx$, and returns 0 otherwise.

First, let's rewrite and simplify the equation.

$$\begin{aligned} x^2+n^2&=2nx \\ x^2-2nx+n^2&=0 \\ (x-n)^2&=0 \\ x&=n. \end{aligned}$$

Thus, it is equivalent to create a machine that, given an integer n in R_0 at ***start***, halts and outputs $n+1$ in R_1.

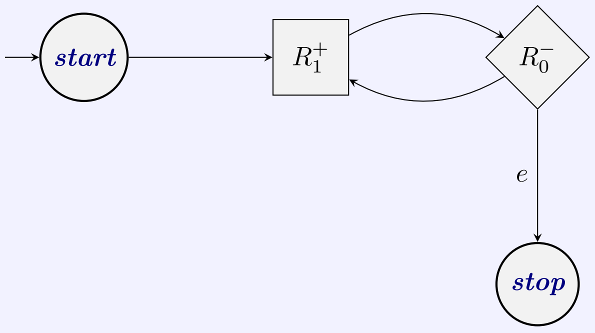

Figure 1: $f(n)\downarrow = x+1$ if $x^2+n^2=2nx$.

Note that the machine never returns 0 because for any input $n \in \mathbb{N}$, there is always a solution $x=n$ that satisfies the equation $n^2+x^2=2nx$.

The register machine starts by putting 1 in R_1. It then repeatedly subtracts 1 from R_0 and adds 1 from R_1 until R_0 is zero. Thus, if R_0 contained n at the start, then R_1 will contain $n+1$ at the end (which is equivalent to $x+1$ given $x=n$).

Problem 2.

Prove that no index set is immune.

A set S is *immune* if it is infinite but has no infinite c.e. subset.

Let I be an index set.

1. **1.** If I is finite, then I is trivially not immune. In fact, the padding lemma tells us that for any e, there are infinitely many indices f such that $\varphi_e = \varphi_f$, and I must contain any such f for any e that it contains. Therefore, if I is finite, then $I = \varnothing$.
2. **2.** Now, suppose I is infinite. Let φ_ℓ be a turing machine that simulates χ_I. Since I is not necessarily computable, φ_ℓ does not necessarily halt on some inputs — but since I is infinite, we know that φ_ℓ *eventually* halts (and returns 1) on infinitely many inputs.

   Then we can construct an infinite c.e. subset $S \subseteq I$ as follows.

   Define

   $$f : \omega^2 \to \{0, 1\}$$
   $$f(e, k) = \begin{cases} 1 & \text{if } \varphi_{\ell,k}(e) \downarrow = 1 \\ 0 & \text{otherwise.} \end{cases}$$

   For $n = 0, 1, 2, 3, \ldots$, Define

   $$I_n = \{e \mid f(e, n) = 1\}.$$

   First, we claim that each I_n is c.e. — in fact, it is computable. Given an input e, we only need to simulate $\varphi_\ell(e)$ for n steps to determine if $e \in I_n$ or not.

   Next, We claim that there exists an n such that I_n is infinite. For the sake of contradiction, suppose I_n is finite for all n. This implies that for any given n (no matter how arbitrarily large), there are only finitely many e such that $\varphi_{\ell,n}(e) \downarrow = 1$, which implies that φ_ℓ converges for only finitely many e, hence I itself is finite. This contradicts the condition that I is infinite.

   Thus, for any index set I, there exists some n such that I_n as constructed above is an infinite, c.e. subset of I — hence I is not immune.

Problem 3.

Give an example of a partial computable function $i(x, y)$ such that:

- $i(x, y) \downarrow$ implies $i(x, y) = 0$ or $i(x, y) = 1$.
- If A is computable, then there exists an e such that $i(e, n) = \chi_A(n)$ for all n.
- $I_x \coloneqq \{y \mid i(x, y) \downarrow > 0\}$ is computable for all x.

Why does $i(x, y)$ not contradict Homework 3, Question 2?

Construction of $i(x, y)$

Let $i(x, y)$ be the following partial computable function:

$$i(x, y) = \begin{cases} 1 & \text{if } \varphi_x(y) \downarrow = 1 \\ 0 & \text{if } \varphi_x(y) \downarrow = 0 \text{ or } \varphi_x(y) \uparrow . \end{cases}$$

1. Whenever $i(x, y) \downarrow$, then $i(x, y) = 0$ or $i(x, y) = 1$.

2. When A is computable, then there exists some index e such that $\varphi_e = \chi_A$. Then $i(e, n) = \varphi_e(n) = \chi_A(n)$ for all n.

3. $I_x = \{y \mid i(x, y) \downarrow > 0\} = \{y \mid \varphi_x(y) = 1\}$. For any fixed x, we can compute I_x by iterating through all $y \in \mathbb{N}$ and checking if $i(x, y) = 1$.

Comparison with Homework 3, Question 2

Homework 3, Question 2 used a diagonalization argument to show that there is no uniform listing of all characteristic functions of the computable sets.

The function $i(x, y)$ does not contradict this fact because it does not exclusively list all characteristic functions of computable sets. Instead, it simulates every other function, althogh it only converges if the simulated function converges to $i \in \{0, 1\}$ and diverges otherwise.

Problem 4.

Let M be the set

$$\{x \mid \forall\, y < x\, (\varphi_x \neq \varphi_y)\}\,.$$

That is, M is the set of minimal indices of computable functions: the smallest indices which define a given (partial) computable function.

(a) Is M immune? Simple?

M is immune but not simple.

Immunity

A set is immune if it is infinite but has no infinite c.e. subset.

While M is infinite, it does not contain any infinite c.e. subset. For the sake of contradiction, suppose it did and let the subset be S. Then there exists a bijection $f : \mathbb{N} \to S$ (since S is c.e.). By the recursion theorem, there exists a fixed point e such that $e \neq f(e)$ and $\varphi_e = \varphi_{f(e)}$. But since f is a bijection, $f(e) \in S \subset M$, so either e is not minimal or $f(e)$ is not minimal, and both cases contradict the constitution of M.

Simplicity

A set is simple if it is c.e. and its complement is immune. Since no immune set is c.e., M cannot be simple.

(b) Is M productive? Creative?

Productivity

Since M is immune, M is not productive.

Given a productive set P, the productive function ψ can be used to generate a c.e. subset of of P, so no productive set can be immune.

Creativity

Since M is immune, M is not c.e., hence not creative.

Problem 5.

Prove that $W_e \leq_1 H$ for all e, then prove that $H \leq_1 K$.

$$M = \{\langle e, k, n\rangle \mid \varphi_e(k) \downarrow = n\}$$

$$H = \{\langle e, k\rangle \mid \varphi_e(k) \downarrow\}$$

$$K = \{e \mid \varphi_e(e) \downarrow\}$$

$W_e \leq_1 H$ for all e

Define the function $f : \mathbb{N} \to \mathbb{N}^2$ by

$$f(k) = \langle e, k\rangle$$

1. When $n \in W_e$, then $\varphi_e(n) \downarrow$, so $f(n) = \langle e, n\rangle \in H$.
2. When $n \notin W_e$, then $\varphi_e(n) \uparrow$, so $f(n) \uparrow$.
3. f is an injection from W_e to H since e is fixed after the definition of f, and every each $k' \in W_e$ is mapped to a unique element $\langle e, k'\rangle \in H$.

Therefore, $n \in W_e$ *if and only if* $f(n) \in H$, so $W_e \leq_1 H$ with witness f.

$H \leq_1 K$

For any $e \in \mathbb{N}$ with $W_e \subseteq \mathbb{N}$ being the domain of φ_e, we can define the function $g : \mathbb{N}^2 \to \mathbb{N}$ as follows:

First, for each $(i, j) \in \mathbb{N}^2$, let $e_{(i,j)}$ be the code of the machine that computes $h_{(i,j)}$, where $h_{(i,j)} : \mathbb{N} \to \mathbb{N}$ is defined as:

$$h_{(i,j)}(n) = \begin{cases} 1 & \text{if } n = e_{(i,j)} \text{ and } \varphi_i(j) \downarrow \\ \uparrow & \text{otherwise.} \end{cases}$$

The recursion theorem with parameters tells us that such a function $h_{(i,j)}$ exists and can be obtained computably.

Proof. To see this, let $f(y, x_0, x_1)$ be a total computable function, then the recursion theorem tells us that there exists an injective, total computable function $r(x_0, x_1)$ such that

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

Let e be an index for $f(r(x_0, x_1), x_0, x_1)$, and define r via $r(x_0, x_1) = d(e, x_0, x_1)$. Then $r(x_0, x_1)$ is a fixed point. □

Define the function $g : \mathbb{N}^2 \to \mathbb{N}$ by

$$g(e,k) = e_{(e,k)}$$

1. When $\langle e,k\rangle \in H$, $\varphi_e(k)\downarrow$. Note that $\varphi_{(e,k)}(e_{(e,k)})$ simulates $\varphi_e(k)$, and $\varphi_e(k)\downarrow$, and since it received $\varphi_{(e,k)}$ as input, it halts and outputs 1. Thus, $g(e,k) = e_{(e,k)} \in K$ when $\langle e,k\rangle \in H$.

2. When $\langle e,k\rangle \notin H$, then $\varphi_{(e,k)}(e_{(e,k)})$ simulates $\varphi_e(k)$, and $\varphi_e(k)\uparrow$, so $g(e,k)\uparrow$. Thus, when $\langle e,k\rangle \notin H$, $g(e,k) = \varphi_{(e,k)} \notin K$.

3. g is an injection from H to K since $\langle e,k\rangle \in H$ is mapped to a unique element $e_{(e,k)} \in K$.

Therefore, $\langle e,k\rangle \in H$ *if and only if* $g(e,k) \in K$, so $H \leq_1 K$ with witness g.

Problem 6.

Suppose C is the set of valid codes for machines based on our coding scheme defined in class. Give a total, computable bijection from ω to C, i.e. a computable function which associates every natural number to a unique machine.

The set of all valid codes, C, is is c.e. and can be enumerated by iterating through all possible codes $1, 2, 3, \ldots$ and checking if φ_i is a valid machine. List the code for the first valid machine as c_1, the code for the second valid machine as c_2, and so on. Then define $f : \mathbb{N} \to C$ with $f(n) = c_n$.

Since there are infinite machines, given any $n \in \mathbb{N}$, eventually a unique code c_n will be listed in C. Therefore, every $n \in \mathbb{N}$ is associated with a unique code for a valid machine, so f is total and bijective.

The function f is also computable since it can be simulated by a Turing Machine that, on input n, enumerates through all codes $1, 2, 3, \ldots$ until it finds the n-th valid code c_n and returns it.

Problem 7.

Show that there is an e such that

$$W_e = \left\{e+1, e^2+4, e^3+9, \ldots\right\}.$$

Let f be the function defined by

$$f : \mathbb{N}^2 \to \mathbb{N}$$
$$(e, n) \mapsto e^n + n^2.$$

f is total computable since polynomial functions, exponential functions, and addition are computable. Construct a self-referential turing machine M as follows:

TM 1: Turing machine M with domain W_e

```
1 On input $n$;
2 Let $e$ be the code for this machine.
3 for $i = 1, 2, 3, \ldots$ do
4     $y_i \leftarrow f(e, i)$
5     if $y_i = n$ then
6         output 1
```

The machine passes its own code as the first argument to f (the recursion theorem guarantees that such a machine exists and can be constructed computably). The machine passes successive integers $1, 2, 3, \ldots$ as the second argument. Accordingly, it computes values

- $f(e, 1) = e^1 + 1^2 = e + 1$,
- $f(e, 2) = e^2 + 2^2 = e^2 + 4$,
- $f(e, 3) = e^3 + 3^2 = e^3 + 9$,
- and so on.

Finally, M checks successive outputs of f against n, halting and outputting 1 if it finds a match. M will only halt if an input n is in the set $\left\{e+1, e^2+4, e^3+9, \ldots\right\}$, so

$$W_e = \left\{e+1, e^2+4, e^3+9, \ldots\right\}.$$

Specifically, M computes the function $h : \mathbb{N} \to \mathbb{N}$ defined by

$$h(x) = \varphi_e(x) = \begin{cases} 1 & \text{if } x \in \left\{e+1, e^2+4, e^3+9, \ldots\right\} \\ \uparrow & \text{otherwise.} \end{cases}$$