Computability Exam Solutions

September 14, 2009

Exercise 1

Rice-Shapiro Application

Given: $A \subseteq C$ (set of computable functions), $f \in A$, and $\forall$ finite $\theta \subseteq f$: $\theta \notin A$.

Prove: $A = \{x \in \mathbb{N} \mid \varphi_x \in A\}$ is not r.e.

Proof:

This is a direct application of Rice-Shapiro theorem.

Rice-Shapiro Theorem: Let $A \subseteq C$ be a set of computable functions, and $A = \{x \mid \varphi_x \in A\}$. Then A is not r.e. if: $\exists f \in A$ such that $\forall$ finite $\theta \subseteq f$: $\theta \notin A$.

Application: We are given exactly the conditions of Rice-Shapiro:

- $f \in A$ (given)
- For every finite function $\theta \subseteq f$, we have $\theta \notin A$ (given)

By Rice-Shapiro theorem, $A = \{x \in \mathbb{N} \mid \varphi_x \in A\}$ is not r.e.

Intuition: The theorem captures the idea that if a set A contains a function f but excludes all finite approximations to f, then enumerating A is impossible because we cannot "build up" to f through finite stages.

Exercise 2

Question: Does there exist a total non-computable $f : \mathbb{N} \to \mathbb{N}$ such that $f(x) \neq \varphi_x(x)$ for only one value $x \in \mathbb{N}$?

Answer: No, such a function cannot exist.

Proof:

Suppose f is total, non-computable, and $f(x) \neq \varphi_x(x)$ for exactly one value $x = c$.

Then for all $x \neq c$: $f(x) = \varphi_x(x)$.

Construction of computable function agreeing with f:

Define $g : \mathbb{N} \to \mathbb{N}$ by:

```
g(x) = {
  f(c)     if x = c
  ϕₓ(x)    if x ≠ c
}
```

Computability of g:

- For $x = c$: return the constant $f(c)$ (computable since it's just one fixed value)
- For $x \neq c$: compute $\varphi_x(x)$ using the universal function (computable)
- The case distinction $x = c$ vs $x \neq c$ is decidable

Therefore g is computable.

Verification that g = f:

- $g(c) = f(c)$ (by construction)
- For $x \neq c$: $g(x) = \varphi_x(x) = f(x)$ (since f agrees with diagonal except at c)

So $g(x) = f(x)$ for all $x \in \mathbb{N}$, meaning $g = f$.

This contradicts the assumption that f is non-computable.

Therefore, no such function can exist.

Exercise 3

Classification of $A = \{x \in \mathbb{N} : \exists y,z \in \mathbb{N}.\ z > 1 \wedge x = y^z\}$

A is the set of perfect powers (excluding first powers): {4, 8, 9, 16, 25, 27, 32, ...}

A is r.e.:

```
scₐ(x) = 1(μ⟨y,z⟩. z > 1 ∧ y^z = x)
```

For any $x \in A$, there exist y,z with $z > 1$ and $x = y^z$. The search will eventually find such y,z.

A is recursive: To decide $x \in A$:

```
For z = 2, 3, 4, ..., ⌊log₂(x)⌋:
  For y = 0, 1, 2, ..., ⌊x^(1/z)⌋:
    If y^z = x, return "x ∈ A"
Return "x ∉ A"
```

This algorithm terminates because:

- The outer loop is bounded by $\lfloor \log_2(x) \rfloor$
- The inner loop is bounded by $\lfloor x^{(1/z)} \rfloor$
- Exponentiation and root extraction are computable

$\bar{A}$ is recursive: Since A is recursive, $\bar{A}$ is also recursive.

Final classification: A and $\bar{A}$ are both recursive.

Exercise 4

Classification of $V = \{x \in \mathbb{N} : |W_x| > 1\}$

V is r.e.:

```
scᵥ(x) = 1(μ(y₁,y₂,t). y₁ ≠ y₂ ∧ H(x,y₁,t) ∧ H(x,y₂,t))
```

This searches for two distinct elements in W_x.

V is not recursive: V is saturated since it expresses the property $|dom(\varphi_x)| > 1$.

By Rice's theorem, since V is saturated and non-trivial:

- $V \neq \emptyset$: Functions with $|W_x| > 1$ exist (e.g., identity function)
- $V \neq \mathbb{N}$: Functions with $|W_x| \leq 1$ exist (e.g., everywhere undefined function, constant functions)

Therefore V is not recursive.

$\bar{V}$ is not r.e.: Since V is r.e. but not recursive, $\bar{V}$ is not r.e.

Final classification: V is r.e. but not recursive; $\bar{V}$ is not r.e.

Exercise 5

Second Recursion Theorem and Application

Second Recursion Theorem: For every total computable function $f : \mathbb{N} \to \mathbb{N}$, there exists $e_0 \in \mathbb{N}$ such that $\varphi_{e0} = \varphi f(e_0)$.

Proof that $\exists n$ such that $W_n = E_n = \{x \cdot n : x \in \mathbb{N}\}$

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

```
g(n,y) = {
  ⌊y/n⌋  if n > 0 and y is a multiple of n
  ↑       otherwise
}
```

More precisely:

```
g(n,y) = {
  y/n  if n > 0 and n divides y
  ↑    otherwise
}
```

Properties of g for fixed n > 0:

- Domain: all multiples of n, i.e., {0, n, 2n, 3n, ...}
- Codomain: {0, 1, 2, 3, ...} = $\mathbb{N}$

So $W_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$ and $E_{s(n)} = \mathbb{N}$.

We need to modify g to make $E_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$ as well.

Corrected construction:

```
g(n,y) = {
  y  if n > 0 and y is a multiple of n
  ↑  otherwise
}
```

Then:

- $W_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$ (domain)
- $E_{s(n)} = \{x \cdot n : x \in \mathbb{N}\}$ (codomain, since output = input for multiples)

By s-m-n theorem, ∃ total computable $s : \mathbb{N} \to \mathbb{N}$ such that $\varphi_{s(n)}(y) = g(n,y)$.

Define f(n) = s(n). By Second Recursion Theorem, ∃n such that $\varphi_n = \varphi f(n) = \varphi_{s(n)}$.

Therefore: $W_n = E_n = \{x \cdot n : x \in \mathbb{N}\}$.