Computability Exam Solutions

September 17, 2010

Exercise 1

Formal Statement and Proof of Closure under Unbounded Minimization

Statement: If $f : \mathbb{N}^{k+1} \to \mathbb{N}$ is computable, then $g : \mathbb{N}^k \to \mathbb{N}$ defined by $g(\vec{x}) = \mu y.f(\vec{x},y)$ is also computable.

Definition:

```
g(x⃗) = μy.f(x⃗,y) = {
  the least y such that f(x⃗,y) = 0  if such y exists
  ↑                                    otherwise
}
```

Proof:

Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be computable. Since f is computable, there exists a URM program P_f that computes f.

Construction of URM program for g:

```
Program P_g on input x⃗ = (x₁,...,xₖ):

Registers: R₁,...,Rₖ (input), Rₖ₊₁ (counter y), additional working registers

1. Store input x⃗ in R₁,...,Rₖ
2. Z(k+1)                        // Initialize y = 0
3. LOOP:
   a. Set up arguments (x⃗,y) for P_f
   b. Execute P_f to compute f(x⃗,y)
   c. J(result,0,FOUND)         // If f(x⃗,y) = 0, jump to FOUND
   d. S(k+1)                    // Increment y
   e. J(0,0,LOOP)               // Jump back to LOOP
4. FOUND:
   a. T(k+1,output)             // Copy y to output register
   b. HALT
```

Correctness:

- If $\exists y$ such that $f(\vec{x},y) = 0$: The program finds the least such y and terminates
- If $\forall y$: $f(\vec{x},y) \neq 0$: The program loops forever (correct behavior for $\uparrow$)

Computability: Since P_g uses only basic URM instructions and calls the computable function f, g is computable.

Therefore, the set of computable functions is closed under unbounded minimization.

Exercise 2

Analysis of $f(x) = \varphi_x(x) + 1$ if $\forall y \leq x: \varphi_y(y) \downarrow$, 0 otherwise

Answer: The function f is not computable.

Proof by contradiction:

Suppose f is computable. We'll derive a contradiction.

Define the set:

```
A = {x ∈ ℕ : ∀y ≤ x. ϕᵧ(y) ↓}
```

Then:

```
f(x) = {
  ϕₓ(x) + 1  if x ∈ A
  0          if x ∉ A
}
```

If f is computable, we can decide membership in A:

```
x ∈ A ⟺ f(x) ≠ 0 ∧ ϕₓ(x) ↓ ∧ f(x) = ϕₓ(x) + 1
```

Constructing a contradiction:

Since we can decide A, define $h : \mathbb{N} \to \mathbb{N}$ by:

```
h(x) = {
  ϕₓ(x) + 1  if x ∈ A
  ↑          if x ∉ A
}
```

If f is computable, then A is decidable, so h is computable.

By the s-m-n theorem, $\exists$ total computable $s : \mathbb{N} \to \mathbb{N}$ such that $\varphi_{s(x)}(y) = h(x)$ for all y (constant function).

In particular, $\varphi_{s(x)}(s(x)) = h(x)$.

Case analysis:

Case 1: $s(x) \in A$ Then $\forall y \leq s(x)$: $\varphi_y(y) \downarrow$, so in particular $\varphi_{s(x)}(s(x)) \downarrow$. Also, $h(x) = \varphi_x(x) + 1$, and $\varphi_{s(x)}(s(x)) = h(x) = \varphi_x(x) + 1$. For $s(x) \in A$, we need $\varphi_{s(x)}(s(x)) \downarrow$, which is true.

Case 2: $s(x) \notin A$
Then $\exists y \leq s(x)$: $\varphi_y(y) \uparrow$, and $h(x) \uparrow$, so $\varphi_{s(x)}(s(x)) \uparrow$. But this means $\varphi_{s(x)}(s(x)) \uparrow$, so $s(x) \notin A$ is consistent.

The contradiction arises from the self-referential nature and the fact that we're essentially trying to solve the halting problem uniformly.

Therefore, f is not computable.

Exercise 3

Classification of A = {x | φ_x quasi-total}

A function f is quasi-total if it is undefined on a finite set of points.

A is saturated: A = {x | φ_x ∈ $\mathcal{A}$} where $\mathcal{A}$ = {f | f is quasi-total}.

A is not r.e.: We use Rice-Shapiro theorem. Consider any total function g (e.g., identity). Then g is quasi-total (undefined on 0 points), so g ∈ $\mathcal{A}$.

For any finite function θ ⊆ g, θ is defined only on finitely many points, so θ is quasi-total, hence θ ∈ $\mathcal{A}$.

This doesn't immediately give us Rice-Shapiro. Let me reconsider.

Consider a function g that is undefined everywhere except on finitely many points, so g is quasi-total. For any finite θ ⊆ g, θ is also quasi-total.

Actually, let me try a different approach. Consider the everywhere undefined function ∅. This function is quasi-total (undefined on all points, which includes being undefined on finitely many). So ∅ ∈ $\mathcal{A}$.

Wait, let me clarify the definition. A function is quasi-total if its domain is co-finite (i.e., the complement of the domain is finite).

Consider the identity function id, which is total, hence quasi-total. So id ∈ $\mathcal{A}$.

Consider a finite function θ ⊆ id. Then θ has finite domain, so its complement is infinite. Therefore θ is not quasi-total, so θ ∉ $\mathcal{A}$.

Since id ∈ $\mathcal{A}$ and ∀ finite θ ⊆ id: θ ∉ $\mathcal{A}$, by Rice-Shapiro theorem, A is not r.e.

Ā is not r.e.: Consider a function g with infinite undefined set (not quasi-total). For finite θ ⊆ g, θ still has finite domain, so θ ∉ $\mathcal{A}$, hence θ ∈ $\bar{\mathcal{A}}$.

Using similar Rice-Shapiro arguments, Ā is not r.e.

Final classification: A and Ā are both not r.e. (and hence not recursive).

Exercise 4

Proof that $\bar{K} \leq_m B$ where B = {x ∈ ℕ | φ_x total}

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

```
g(x,y) = {
  0      if x ∉ K
  ↑      if x ∈ K
}
```

This can be implemented as:

```
g(x,y) = μz.H(x,x,z)
```

Since H is decidable, g is computable.

By the s-m-n theorem, there exists a total computable function $s : \mathbb{N} \to \mathbb{N}$ such that:

```
φₛ₍ₓ₎(y) = g(x,y)
```

Verification of the reduction:

- **If $x \notin K$:** Then $\varphi_x(x) \uparrow$, so $\forall z\ \neg H(x,x,z)$, hence $\forall y$: $\varphi_{s(x)}(y) = 0$. Therefore $\varphi_{s(x)}$ is total, so $s(x) \in B$.
- **If $x \in K$:** Then $\varphi_x(x) \downarrow$, so $\exists z$: $H(x,x,z)$, hence $\forall y$: $\varphi_{s(x)}(y) \uparrow$. Therefore $\varphi_{s(x)}$ is nowhere defined (not total), so $s(x) \notin B$.

Therefore, $x \in \bar{K} \iff s(x) \in B$, which means $\bar{K} \leq_m B$ via s.

Exercise 5

s-m-n Theorem and Application

s-m-n Theorem: For every m, n ≥ 1, there exists a total computable function s_{m,n} : $\mathbb{N}^{m+1} \to \mathbb{N}$ such that for all $e \in \mathbb{N}$, $\vec{x} \in \mathbb{N}^m$, $\vec{y} \in \mathbb{N}^n$:

```
φₑ⁽ᵐ⁺ⁿ⁾(x⃗, y⃗) = φₛₘ,ₙ₍ₑ,ₓ⃗₎⁽ⁿ⁾(y⃗)
```

Proof of existence of $s : \mathbb{N}^2 \to \mathbb{N}$ such that $W_{s(x,y)} = \{z : x \cdot z = y\}$

Define $g : \mathbb{N}^3 \to \mathbb{N}$ by:

```
g(x,y,z) = {
  z  if x · z = y
  ↑  otherwise
}
```

This function is computable since:

- Multiplication $x \cdot z$ is computable
- Equality testing $x \cdot z = y$ is decidable
- Conditional branching is computable

By the s-m-n theorem (with m = 2, n = 1), there exists a total computable function $s : \mathbb{N}^2 \to \mathbb{N}$ such that:

```
φs(x,y)(z) = g(x,y,z)
```

Verification:

```
Ws(x,y) = {z : φs(x,y)(z) ↓}
        = {z : g(x,y,z) ↓}
        = {z : x · z = y}
```

Therefore, s is the desired function such that $W_{s(x,y)} = \{z : x \cdot z = y\}$.