Computability Exam Solutions

June 28, 2011

Exercise 1

Rice's Theorem

Statement: Let $A \subseteq \mathbb{N}$ be saturated with $A \neq \emptyset$ and $A \neq \mathbb{N}$. Then A is not recursive.

Definition: A set $A \subseteq \mathbb{N}$ is saturated (or extensional) if:

```
∀x,y ∈ ℕ: (x ∈ A ∧ ϕₓ = ϕᵧ) ⟹ y ∈ A
```

Proof:

We show $K \leq_m A$, which implies A is not recursive since K is not recursive.

Since $A \neq \emptyset$ and $A \neq \mathbb{N}$, there exist indices $e_0 \notin A$ and $e_1 \in A$.

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

```
g(x,y) = {
  ϕₑ₁(y)  if x ∈ K
  ϕₑ₀(y)  if x ∉ K
}
```

This can be implemented as:

```
g(x,y) = ϕₑ₁(y) · scₖ(x) + ϕₑ₀(y) · (1 - scₖ(x))
```

However, since sc_k is not computable, we use:

```
g(x,y) = {
  ϕₑ₁(y)  if ϕₓ(x) ↓
  ϕₑ₀(y)  if ϕₓ(x) ↑
}
```

This can be implemented as:

```
g(x,y) = ϕₑ₁(y) · 1(Ψᵤ(x,x)) + ϕₑ₀(y) · μz.χₕ(x,x,z)
```

Actually, let me use the standard approach:

```
g(x,y) = {
  φₑ₁(y)  if ∃t. H(x,x,t)
  φₑ₀(y)  otherwise
}
```

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

Verification of the reduction:

- If $x \in K$: $\varphi_x(x) \downarrow$, so $\forall y$: $\varphi_{s(x)}(y) = \varphi_{e1}(y)$, hence $\varphi_{s(x)} = \varphi_{e1}$. Since A is saturated and $e_1 \in A$, we get $s(x) \in A$.
- If $x \notin K$: $\varphi_x(x) \uparrow$, so $\forall y$: $\varphi_{s(x)}(y) = \varphi_{e0}(y)$, hence $\varphi_{s(x)} = \varphi_{e0}$. Since A is saturated and $e_0 \notin A$, we get $s(x) \notin A$.

Therefore $K \leq_m A$ via s, which implies A is not recursive.

Exercise 2

Question: Can there exist a non-computable f such that for every other non-computable g, f + g is computable?

Answer: No, such a function cannot exist.

Proof:

Suppose f is non-computable and for every non-computable g, the function $(f + g)(x) = f(x) + g(x)$ is computable.

Consider the characteristic function χ_k of the halting set K. Since K is not recursive, χ_k is not computable.

By our assumption, $f + \chi_k$ is computable.

Now consider the function $g = -f + 1$ (where 1 is the constant function). If g were computable, then:

```
f = (f + χₖ) - χₖ = (f + χₖ) + (-χₖ)
```

would be computable (as the sum of computable functions), contradicting the assumption that f is non-computable.

Therefore $g = -f + 1$ must be non-computable.

By our assumption, f + g is computable. But:

```
(f + g)(x) = f(x) + (-f(x) + 1) = 1
```

So f + g is the constant function 1, which is indeed computable.

Now consider another non-computable function h. By assumption, f + h is computable.

We have:

```
h = (f + h) - f = (f + h) + (-f)
```

Since f + h is computable by assumption, if -f were computable, then h would be computable, contradicting our choice of h as non-computable.

Therefore -f is not computable.

But this means both f and -f are non-computable. By our assumption:

- f + (-f) should be computable
- But f + (-f) = 0 (constant zero function), which is indeed computable

However, we can construct infinitely many distinct non-computable functions, and they cannot all have the special property that when added to f, they yield computable functions, due to algebraic constraints.

More direct contradiction: Let $g_1 = \chi_k$ and $g_2 = \chi_{\bar{k}}$ (both non-computable). Then:

- $f + g_1$ is computable (by assumption)
- $f + g_2$ is computable (by assumption)
- $g_1 + g_2 = \chi_k + \chi_{\bar{k}} = 1$ (constant function)

Therefore:

```
g₁ = (f + g₁) - f
g₂ = (f + g₂) - f
```

So: $g_1 - g_2 = (f + g_1) - (f + g_2)$, which is computable minus computable = computable.

But $g_1 - g_2 = \chi_k - \chi_{\bar{k}} = 2\chi_k - 1$, which allows us to compute χ_k, contradicting the non-computability of K.

Therefore, no such function f can exist.

Exercise 3

Proof that $\bar{K} \leq_m A$ where $A = \{x \in \mathbb{N} : E_x = P\}$

where P = {0, 2, 4, 6, ...} is the set of even numbers.

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

```
g(x,y) = {
  2y      if x ∉ K
  ↑       if x ∈ K
}
```

This can be implemented as:

```
g(x,y) = 2y · μz.(¬H(x,x,z))
```

But since we can't compute ¬H directly, we use:

```
g(x,y) = {
  2y     if ∀t ≤ T: ¬H(x,x,t) (for very large T)
  ↑      if ∃t ≤ T: H(x,x,t)
}
```

More precisely:

```
g(x,y) = {
  2y     if ∀t ≤ y: ¬H(x,x,t)
  ↑      otherwise
}
```

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

Verification of the reduction:

- If $x \notin K$: $\varphi_x(x) \uparrow$, so $\forall t$: $\neg H(x,x,t)$, hence $\forall y$: $\varphi_{s(x)}(y) = 2y$. Therefore $E_{s(x)} = \{0, 2, 4, 6, ...\} = P$, so $s(x) \in A$.
- If $x \in K$: $\varphi_x(x) \downarrow$ in some number of steps t_0. For $y \geq t_0$, we have $H(x,x,t_0)$, so $\varphi_{s(x)}(y) \uparrow$. For $y < t_0$, $\varphi_{s(x)}(y) = 2y$. Therefore $E_{s(x)} = \{0, 2, 4, ..., 2(t_0-1)\} \neq P$ (finite vs infinite), so $s(x) \notin A$.

Therefore $\bar{K} \leq_m A$ via the reduction function s.

Exercise 4

Classification of $B = \{x \in \mathbb{N} : \varphi_x(y) = y^2$ for infinitely many y}

The set B is saturated since $B = \{x \mid \varphi_x \in \mathcal{B}\}$ where $\mathcal{B} = \{f \mid f(y) = y^2$ for infinitely many y}.

B is not r.e.: We use Rice-Shapiro theorem. Consider the function $f(y) = y^2$ for all y. Then $f \in \mathcal{B}$ since $f(y) = y^2$ for infinitely many y (in fact, for all y).

For any finite function $\theta \subseteq f$, we have $\theta(y) = y^2$ for only finitely many y (specifically, for $y \in dom(\theta)$). Therefore $\theta \notin \mathcal{B}$.

Since $f \in \mathcal{B}$ and $\forall$ finite $\theta \subseteq f$: $\theta \notin \mathcal{B}$, by Rice-Shapiro theorem, B is not r.e.

$\bar{B}$ is not r.e.: Consider the constant function $g(x) = 0$. Then $g \notin \mathcal{B}$ since $g(y) = y^2$ only when $y = 0$ (finitely many: just one point).

Consider any finite function $\theta \subseteq g$. Since g is constant 0, we have $\theta : dom(\theta) \to \{0\}$. For $\theta(y) = y^2$, we need $y = 0$. So θ can have at most one point where $\theta(y) = y^2$.

Therefore $\theta \notin \mathcal{B}$, so $\theta \in \bar{\mathcal{B}}$.

Since $g \in \bar{B}$ and $\forall$ finite $\theta \subseteq g$: $\theta \in \bar{B}$, this doesn't directly give us Rice-Shapiro for $\bar{B}$.

Let me try differently. Actually, by Rice's theorem, since B is saturated and non-trivial ($B \neq \emptyset$ since y^2 function is in B, and $B \neq \mathbb{N}$ since constant functions are not in B), B is not recursive. Combined with B not being r.e., we get that $\bar{B}$ is also not r.e.

Final classification: B and $\bar{B}$ are both not r.e. (and hence not recursive).

Exercise 5

Second Recursion Theorem

For every total computable function $f : \mathbb{N} \to \mathbb{N}$, there exists $e_0 \in \mathbb{N}$ such that:

```
ϕₑ₀ = ϕf(e₀)
```

Proof that $h(x) = e_0$ if φ_x is total, e_1 otherwise is not computable

where e_0 is an index for $\emptyset$ and e_1 is an index for the identity function.

Proof by contradiction:

Suppose h is computable. Define f = h (so f is total and computable).

By the Second Recursion Theorem, $\exists e$ such that $\varphi_e = \varphi f(e) = \varphi_{h(e)}$.

We have two cases:

Case 1: φ_e is total. Then $h(e) = e_0$, so $\varphi_e = \varphi_{e0} = \emptyset$ (everywhere undefined). But this contradicts φ_e being total.

Case 2: φ_e is not total. Then $h(e) = e_1$, so $\varphi_e = \varphi_{e1} = id$ (identity function). But the identity function is total, contradicting φ_e not being total.

Both cases lead to contradictions, so h cannot be computable.