Computability Exam Solutions

March 30, 2010

Exercise 1

Definition of saturated set and proof that K is not saturated

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

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

In other words, A is saturated if it expresses a property of functions rather than specific indices.

Proof that K is not saturated:

We'll construct indices e and e' such that $\varphi_e = \varphi_{e'}$, $e \in K$, but $e' \notin K$.

Step 1: Construct a specific function using Second Recursion Theorem.

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

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

This can be written as:

```
g(x,y) = μz.|y - x|
```

Since g is computable, by the s-m-n theorem, ∃ total computable $s : \mathbb{N} \to \mathbb{N}$ such that:

```
ϕs(x)(y) = g(x,y)
```

By the Second Recursion Theorem, ∃e such that $\varphi_e = \varphi s(e)$.

Therefore:

```
ϕₑ(y) = ϕs(e)(y) = g(e,y) = {
  0  if y = e
  ↑  otherwise
}
```

Step 2: Show $e \in K$. Since $\varphi_e(e) = 0 \downarrow$, we have $e \in K$.

Step 3: Find $e' \neq e$ such that $\varphi_e = \varphi_e'$. Since there are infinitely many indices for each computable function, $\exists e' \neq e$ such that $\varphi_e' = \varphi_e$.

Step 4: Show $e' \notin K$. We have $\varphi_e'(e') = \varphi_e(e') \uparrow$ (since $e' \neq e$), so $e' \notin K$.

Conclusion: We have $\varphi_e = \varphi_e'$, $e \in K$, but $e' \notin K$. Therefore, K is not saturated.

Exercise 2

Analysis of $f(x) = x+2$ if $\varphi_x(x) \downarrow$, $x \dot{-} 1$ otherwise

Answer: The function f is not computable.

Proof:

Suppose f is computable. Then we can decide the halting problem as follows:

Given input x, compute f(x):

- If $f(x) = x + 2$, then $\varphi_x(x) \downarrow$, so $x \in K$
- If $f(x) = x \dot{-} 1$, then $\varphi_x(x) \uparrow$, so $x \notin K$

This gives us a decision procedure for $K = \{x : \varphi_x(x) \downarrow\}$.

Verification:

- If $\varphi_x(x) \downarrow$: $f(x) = x + 2 \neq x \dot{-} 1$ (since $x + 2 > x \geq x \dot{-} 1$)
- If $\varphi_x(x) \uparrow$: $f(x) = x \dot{-} 1 \neq x + 2$

So the two cases are distinguishable, making K decidable if f were computable.

Since K is undecidable, f cannot be computable.

Exercise 3

Classification of $A = \{x \in \mathbb{N} : |W_x| > |E_x|\}$

A is not saturated: Consider two functions with the same input-output behavior but different domain/codomain sizes due to internal computation structure. The condition depends on cardinalities which can vary between equivalent functions.

A is r.e.:

```
x ∈ A ⟺ |Wₓ| > |Eₓ|
```

We can semi-decide this by:

```
scA(x) = 1(μt. [∃ injective f: Eₓ → Wₓ witnessed within t steps])
```

If $|W_x| > |E_x|$, then eventually we'll find enough evidence to establish this inequality.

Actually, let me be more precise. We can enumerate elements of W_x and E_x up to time t, and check if $|W_x \cap [0,t]| > |E_x \cap [0,t]|$. If this becomes true and remains true, then $|W_x| > |E_x|$.

A is not recursive: We can show this is undecidable by reducing from totality or other undecidable problems. The difficulty is that comparing infinite cardinalities requires examining the full extent of both sets.

Ā is not r.e.: Since A is r.e. but not recursive, Ā is not r.e.

Final classification: A is r.e. but not recursive; Ā is not r.e.

Exercise 4

Classification of $B = \{x \in \mathbb{N} : img(f) \cap E_x \neq \varnothing\}$

where $f : \mathbb{N} \to \mathbb{N}$ is a fixed total computable function.

B is r.e.:

```
scB(x) = 1(μ⟨y,z,t⟩. y ∈ img(f) ∧ S(x,z,y,t))
```

Since f is computable, we can enumerate img(f) and simultaneously search for elements that appear in both img(f) and E_x.

More precisely:

```
scB(x) = 1(μ⟨w,z,t⟩. S(x,z,f(w),t))
```

This searches for w,z,t such that $\varphi_x(z) = f(w)$, which means $f(w) \in E_x \cap img(f)$.

B is generally not recursive: The recursiveness depends on the specific function f, but for most f, determining whether E_x intersects with img(f) is undecidable.

B̄ is generally not r.e.: To show $x \in \bar{B}$, we need to prove $img(f) \cap E_x = \varnothing$, which requires showing that no element of img(f) ever appears in E_x. This is typically undecidable.

Final classification: B is r.e.; B and B̄ are typically not recursive.

Exercise 5

Theorem: $A \subseteq \mathbb{N}$ is recursive $\Longleftrightarrow A \leq_m \{0\}$

Proof:

(⇒) If A is recursive, then $A \leq_m \{0\}$

If A is recursive, then its characteristic function χ_a is computable.

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

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

Since A is recursive, Ā is also recursive, so χ_Ā is computable, hence f is computable.

For the reduction property:

```
x ∈ A ⟺ f(x) = 0 ⟺ f(x) ∈ {0}
```

Therefore A $\leq_m$ {0} via f.

(⇐) If A $\leq_m$ {0}, then A is recursive

Suppose A $\leq_m$ {0} via some total computable function $f : \mathbb{N} \to \mathbb{N}$.

Then:

```
x ∈ A ⟺ f(x) ∈ {0} ⟺ f(x) = 0
```

We can compute the characteristic function of A as:

```
χ_a(x) = {
  1  if f(x) = 0
  0  if f(x) ≠ 0
}
```

Since f is computable and equality/inequality with 0 is decidable, χ_a is computable.

Therefore A is recursive.

Conclusion: A is recursive ⟺ A $\leq_m$ {0}.