Computability Exam Solutions

March 23, 2011

Exercise 1

Statement of the s-m-n theorem

For every $m, n \geq 1$, 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) = ϕ_{s_{m,n}(e,x⃗)}^{(n)}(y⃗)
```

Informal proof using encoding/decoding

The key idea is that we can "pre-load" some arguments into a program.

Given a program P_e that computes $\varphi_e^{(m+n)}$, we want to construct a program $P_{s(e,\vec{x})}$ that computes the function $\lambda \vec{y}.\varphi_e^{(m+n)}(\vec{x}, \vec{y})$.

Construction:

1. **Encoding step:** Given e and fixed values $\vec{x} = (x_1,\ldots,x_m)$, we construct a new program $P_{s(e,\vec{x})}$ that:
   - First stores the values $x_1,\ldots,x_m$ in designated registers
   - Then takes input $\vec{y} = (y_1,\ldots,y_n)$ in the standard input registers
   - Calls the original program P_e with the combined input $(\vec{x}, \vec{y})$
2. **Effective construction:** The function $s_{m,n}(e,\vec{x})$ can be computed by:
   - Taking the program code for P_e
   - Prepending instructions that load $x_1,\ldots,x_m$ into registers
   - Adjusting register numbering and jump addresses appropriately
   - Encoding the resulting program to get index $s_{m,n}(e,\vec{x})$
3. **Computability:** Since we can effectively manipulate program codes (using encoding/decoding of URM programs), and the transformation is algorithmic, $s_{m,n}$ is computable.

The theorem holds because the constructed program $P_{s(e,\vec{x})}$ computes exactly $\varphi_e^{(m+n)}(\vec{x}, \vec{y})$ when given input $\vec{y}$.

Exercise 2

Question: Does there exist a non-computable increasing function?

A function $f : \mathbb{N} \to \mathbb{N}$ is increasing if it's total and $\forall x,y \in \mathbb{N}: x \leq y \implies f(x) \leq f(y)$.

Answer: Yes, such functions exist.

Construction:

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

```
f(x) = x + |{y ≤ x : y ∈ K}|
```

where K is the halting set.

Verification:

1. **f is total:** For each x, the set $\{y \le x : y \in K\}$ is finite, so its cardinality is well-defined.
2. **f is increasing:** If $x \le x'$, then $\{y \le x : y \in K\} \subseteq \{y \le x' : y \in K\}$, so:

   ```
   f(x) = x + |{y ≤ x : y ∈ K}| ≤ x' + |{y ≤ x' : y ∈ K}| = f(x')
   ```

3. **f is not computable:** If f were computable, we could decide K as follows:

   ```
   To decide if x ∈ K:
   - Compute f(x) and f(x-1) (if x > 0)
   - If f(x) > f(x-1) + 1, then x ∈ K
   - Otherwise x ∉ K
   ```

   This would contradict the undecidability of K.

Therefore, non-computable increasing functions exist.

Exercise 3

Classification of $A = \{x \in \mathbb{N} : W_x \cap E_x = \emptyset\}$

The set A contains indices of functions whose domain and codomain are disjoint.

A is r.e.:

```
scₐ(x) = 1(μ⟨y,z,t⟩. H(x,y,t) ∧ S(x,z,y,t))
```

This searches for evidence of a contradiction: $y \in W_x$ and $y \in E_x$. If such evidence is never found, $x \in A$.

Actually, this is backwards. Let me reconsider. We want:

```
x ∈ A ⟺ Wₓ ∩ Eₓ = ∅ ⟺ ¬∃y. (y ∈ Wₓ ∧ y ∈ Eₓ)
```

Since we need to show the absence of intersection elements, A is actually **not r.e.**

A is not r.e.: We show $K \le_m \bar{A}$. Define $g : \mathbb{N}^2 \to \mathbb{N}$ by:

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

By s-m-n theorem, ∃s such that $\varphi_{s(x)}(y) = g(x,y)$.

- If $x \in K$: $W_{s(x)} = E_{s(x)} = \mathbb{N}$, so $W_{s(x)} \cap E_{s(x)} = \mathbb{N} \neq \emptyset$, hence $s(x) \notin A$
- If $x \notin K$: $W_{s(x)} = E_{s(x)} = \emptyset$, so $W_{s(x)} \cap E_{s(x)} = \emptyset$, hence $s(x) \in A$

This gives $K \leq_m \bar{A}$, so $\bar{A}$ is not r.e., hence A is not recursive.

Ā is r.e.:

```
scĀ(x) = 1(μ⟨y,z,t⟩. H(x,y,t) ∧ S(x,z,y,t))
```

This searches for y such that $y \in W_x \cap E_x$.

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

Exercise 4

Classification of B = {x ∈ ℕ : ∃y > x. y ∈ E_x}

B is r.e.:

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

This searches for y > x and z,t such that $\varphi_x(z) = y$ in t steps.

B is not recursive: We show $K \leq_m B$. Define $g : \mathbb{N}^2 \to \mathbb{N}$ by:

```
g(x,y) = {
  x + 1  if x ∈ K
  ↑      if x ∉ K
}
```

By s-m-n theorem, ∃s such that $\varphi_{s(x)}(y) = g(x,y)$.

- If $x \in K$: $E_{s(x)} = \{x + 1\}$, and since $x + 1 > x$, we have $s(x) \in B$
- If $x \notin K$: $E_{s(x)} = \emptyset$, so no $y > x$ exists in $E_{s(x)}$, hence $s(x) \notin B$

This gives $K \leq_m B$, so B is not recursive.

B̄ is not r.e.: Since B is r.e. but not recursive, B̄ is not r.e.

Final classification: B is r.e. but not recursive; B̄ is not r.e.

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 $\exists n \in \mathbb{N}$ such that φ_n is total and $|E_n| = n$

We use the Second Recursion Theorem with a carefully constructed function.

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

```
h(x,y) = {
  ⌊y/x⌋  if x > 0 and y < x²
  ↑      otherwise
}
```

For fixed $x > 0$, this function has:

- Domain: $\{0, 1, 2, ..., x^2 - 1\}$
- Codomain: $\{0, 1, 2, ..., x - 1\}$
- $|\text{Domain}| = x^2$, $|\text{Codomain}| = x$

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

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

For this n:

- $\varphi_n(y) = h(n,y)$ which is total on $\{0, 1, ..., n^2 - 1\}$ and undefined elsewhere
- $E_n = \{0, 1, 2, ..., n - 1\}$
- $|E_n| = n$

If we want φ_n to be total, we need to modify the construction. Define instead:

```
h(x,y) = y mod x  (for x > 0)
```

Then φ_n is total, $W_n = \mathbb{N}$, $E_n = \{0, 1, ..., n-1\}$, and $|E_n| = n$.