Computability Exam Solutions

April 4, 2011

Exercise 1

Definition of Unbounded Minimization

Given a function $f : \mathbb{N}^{k+1} \to \mathbb{N}$, the unbounded minimization operation $\mu y.f(\vec{x},y)$ produces a function $g : \mathbb{N}^k \to \mathbb{N}$ defined by:

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

Proof that URM-computable functions are closed under unbounded minimization

Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be URM-computable, and define $g(\vec{x}) = \mu y.f(\vec{x},y)$.

Since f is URM-computable, there exists a URM program P_f that computes f.

URM program construction for g:

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

1. Store x⃗ in registers R₁,...,Rₖ
2. Initialize counter y = 0 in register Rₖ₊₁
3. Loop:
   a. Set up input (x⃗,y) for program P_f
   b. Execute P_f to compute f(x⃗,y)
   c. If f(x⃗,y) = 0:
      - Copy y to output register
      - Halt with result y
   d. Otherwise:
      - Increment y (using S(k+1))
      - Jump back to step 3a
```

Detailed URM instructions:

- Use Z(n), T(m,n), S(n), J(m,n,t) as basic operations
- The loop structure uses conditional jumps J(m,n,t)
- Increment operation S(k+1) for the counter
- Comparison with 0 using J(result_reg, zero_reg, found_label)

Since this algorithm systematically searches for the minimal y satisfying $f(\vec{x},y) = 0$, and terminates when such y is found (or runs forever if none exists), it correctly computes $g(\vec{x}) = \mu y.f(\vec{x},y)$.

The construction uses only basic URM operations, so g is URM-computable.

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

Exercise 2

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

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

Answer: Yes, such functions exist.

Construction:

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

```
f(x) = max(0, N - |{y ≤ x : y ∈ K}|)
```

where K is the halting set and N is a sufficiently large constant.

Alternative construction:

```
f(x) = {
  2^{x+1} - 2^x - |{y ≤ x : y ∈ K}|  if this is ≥ 0
  0                                    otherwise
}
```

Verification:

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

   ```
   |{y ≤ x : y ∈ K}| ≤ |{y ≤ x' : y ∈ K}|
   ```

   Therefore:

   ```
   f(x) = N - |{y ≤ x : y ∈ K}| ≥ N - |{y ≤ x' : y ∈ K}| = f(x')
   ```

3. **f is not computable:** If f were computable, we could decide membership in K:

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

   This would contradict the undecidability of K.

Therefore, non-computable decreasing functions exist.

Exercise 3

Classification of A = {x ∈ ℕ : E_x = W_{x+1}}

where for $X \subseteq \mathbb{N}$, we define $X + 1 = \{x + 1 : x \in X\}$.

Wait, let me re-read this. The notation is $E_x = W_{x+1}$, meaning the codomain of φ_x equals the domain of φ_{x+1}.

A is not saturated: The condition depends on specific indices x and x+1, not just the function φ_x. If $\varphi_x = \varphi_y$ but $x \neq y$, then we compare E_x with W_{x+1} versus E_y with W_{y+1}, which are different conditions.

A is r.e.:

```
x ∈ A ⟺ ∀z. (z ∈ Eₓ ⟺ z ∈ Wₓ₊₁)
```

This can be expressed as:

```
scₐ(x) = lim_{t→∞} [∀z≤t ((∃y,s≤t S(x,y,z,s)) ⟹ (∃s≤t H(x+1,z,s)))]
```

If the equivalence holds for all z up to some bound and continues to hold, eventually we can confirm x ∈ A.

A is not recursive: The problem is that checking $E_x = W_{x+1}$ requires verifying both inclusions, and checking that elements are NOT in these sets is generally undecidable.

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

Exercise 4

Classification of B = {x ∈ ℕ : ∀y > x. 2y ∈ W_x}

B is not r.e.: The condition requires that ALL y > x satisfy $2y \in W_x$. This is a universal quantification over an infinite set, which typically leads to non-r.e. sets.

We can show $\bar{K} \leq_m B$. Define $g : \mathbb{N}^2 \to \mathbb{N}$ by:

```
g(x,z) = {
  z/2    if z is even, z/2 > x, and x ∉ K
  ↑      otherwise
}
```

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

- If $x \notin K$: For all y > x, we have $2y \in W_{s(x)}$ (since g(x,2y) = y), so $s(x) \in B$
- If $x \in K$: $W_{s(x)} = \emptyset$, so ∀y > x: $2y \notin W_{s(x)}$, hence $s(x) \notin B$

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

$\bar{B}$ is r.e.:

```
x ∈ B̄ ⟺ ∃y > x. 2y ∉ Wₓ
```

This is equivalent to:

```
scB̄(x) = 1(μy. y > x ∧ ∀t≤T ¬H(x,2y,t))
```

for sufficiently large T. Actually, this doesn't work directly since we need to show 2y is never in W_x.

Let me reconsider. Actually:

```
x ∈ B̄ ⟺ ∃y > x. 2y ∉ Wₓ
```

We can search for a $y > x$ such that we can prove $2y \notin W_x$, but this is difficult since proving non-membership in W_x is undecidable.

Alternative approach: $\bar{B}$ is r.e. if we can find a $y > x$ such that we have enough evidence that 2y will never be in W_x.

Final classification: B is not r.e.; $\bar{B}$ might be r.e. depending on precise analysis, but likely also 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 $\Delta(x) = \min\{y : \varphi_y \neq \varphi_x\}$ is not computable

Proof by contradiction using Second Recursion Theorem:

Suppose Δ is computable. Define $f : \mathbb{N} \to \mathbb{N}$ by $f(x) = \Delta(x)$.

By the Second Recursion Theorem, $\exists e$ such that $\varphi_e = \varphi f(e) = \varphi\Delta(e)$.

By definition of Δ, we have $\Delta(e) = \min\{y : \varphi_y \neq \varphi_e\}$.

Since $\varphi_e = \varphi\Delta(e)$, we get $\varphi_e = \varphi\Delta(e)$ where $\Delta(e)$ is the smallest index of a function different from φ_e.

This creates a contradiction: $\Delta(e)$ is supposed to be the index of a function different from φ_e, but we also have $\varphi_e = \varphi\Delta(e)$.

More precisely:

- By definition: $\varphi_{\Delta(e)} \neq \varphi_e$ (since $\Delta(e) = \min\{y : \varphi_y \neq \varphi_e\}$)
- By Second Recursion Theorem: $\varphi_e = \varphi_{\Delta(e)}$

These two facts contradict each other.

Therefore, Δ cannot be computable.