# Computability Exam Solutions

## June 20, 2012

### Exercise 1

#### URM⁻ variant with predecessor P(n) instead of successor S(n)

The URM$^-$ machine replaces the successor instruction S(n) with predecessor P(n), where P(n) replaces the content $r_n$ of register n with $r_n \dot{-} 1$ (proper subtraction).

**Relationship between $C^-$ and C:**

**$C^- \subset C$ (strict inclusion)**

**Proof:**

**1. $C^- \subseteq C$:** Every URM$^-$-computable function is URM-computable.

The predecessor function pred(x) = $x \dot{-} 1$ is URM-computable using standard URM instructions:

```
pred(x) can be computed by:
- Copy x to working register
- Use loop with successor and comparison to compute x-1
```

Since pred is URM-computable, any URM$^-$ program can be simulated by a URM program by replacing each P(n) instruction with a subroutine that computes the predecessor.

**2. $C^- \neq C$:** The successor function succ(x) = x + 1 is not URM$^-$-computable.

In URM$^-$, we only have: Z(n), T(m,n), J(m,n,t), P(n).

With these instructions, we can only:

- Set registers to 0
- Copy between registers
- Jump based on equality
- Decrement registers

There is no way to increment a register value. Since all operations either preserve or decrease values, and we start with finite input, we cannot produce a value larger than the maximum input value.

Therefore, succ $\notin C^-$, but succ $\in$ C.

**Conclusion:** $C^- \subset C$ (strict inclusion).

### Exercise 2

**Question: Does there exist a total non-computable $f : \mathbb{N} \to \mathbb{N}$ with $f(x) = x^2$ for every $x$ such that $\varphi_x(x) \downarrow$?**

**Answer: Yes, such a function exists.**

**Construction:**

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

```
f(x) = {
  x²     if ϕₓ(x) ↓
  0      if ϕₓ(x) ↑
}
```

**Verification:**

1. **f is total:** For every $x \in \mathbb{N}$, either $\varphi_x(x) \downarrow$ or $\varphi_x(x) \uparrow$, so $f(x)$ is defined.
2. **f satisfies the condition:** By construction, $f(x) = x^2$ whenever $\varphi_x(x) \downarrow$.
3. **f is not computable:** Suppose f were computable. Then we could decide the halting problem:

   ```
   For input x:
     compute f(x)
     if f(x) = x² then ϕₓ(x) ↓
     else ϕₓ(x) ↑
   ```

   This would make $K = \{x : \varphi_x(x) \downarrow\}$ decidable, contradicting its undecidability.

Therefore, such a function f exists.

## Exercise 3

### Proof that $\bar{K} \leq_m A$ where $A = \{x \mid \varphi_x \text{ is total}\}$

We need to find a total computable function f such that:

```
x ∈ K̄ ⟺ f(x) ∈ A
```

**Construction:**

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

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

This can be implemented as:

```
g(x,y) = μz. H(x,x,z)
```

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 $x \in \bar{K}$: $\varphi_x(x) \uparrow$, so $g(x,y) = 0$ for all y, hence $\varphi_{s(x)}$ is the constant 0 function (total), so $s(x) \in A$.
- If $x \in K$: $\varphi_x(x) \downarrow$, so $g(x,y) \uparrow$ for all y, hence $\varphi_{s(x)}$ is everywhere undefined (not total), so $s(x) \notin A$.

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

## Exercise 4

**Classification of $B = \{x \in \mathbb{N} : inc(W_x) = E_x\}$**

where $inc(X) = X \cup \{x + 1 : x \in X\}$.

**B is saturated:** $B = \{x \mid \varphi_x \in \mathcal{B}\}$ where $\mathcal{B} = \{f \mid inc(dom(f)) = cod(f)\}$.

**B is not r.e.:** We use Rice-Shapiro theorem. Consider the identity function id.

- $dom(id) = \mathbb{N}$
- $inc(dom(id)) = inc(\mathbb{N}) = \mathbb{N} \cup \{x+1 : x \in \mathbb{N}\} = \mathbb{N}$
- $cod(id) = \mathbb{N}$

So $inc(dom(id)) = cod(id)$, hence $id \in \mathcal{B}$.

Consider the finite function $\theta = \{(0,2)\}$.

- $dom(\theta) = \{0\}$
- $inc(dom(\theta)) = \{0\} \cup \{1\} = \{0,1\}$
- $cod(\theta) = \{2\}$

Since $\{0,1\} \neq \{2\}$, we have $\theta \notin \mathcal{B}$.

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

**$\bar{B}$ is not r.e.:** Consider the empty function ∅.

- $inc(dom(\emptyset)) = inc(\emptyset) = \emptyset$
- $cod(\emptyset) = \emptyset$

So $inc(dom(\emptyset)) = cod(\emptyset)$, hence $\emptyset \in \mathcal{B}$.

For any function $f \notin \mathcal{B}$, consider $\theta = \emptyset \subseteq f$. We have $\theta \in \mathcal{B}$.

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

Wait, this reasoning is incorrect. Let me reconsider $\bar{B}$.

Consider $f(x) = x + 2$. Then:

- $dom(f) = \mathbb{N}$
- $inc(dom(f)) = \mathbb{N}$
- $cod(f) = \{2,3,4,...\}$

Since $\mathbb{N} \neq \{2,3,4,...\}$, we have $f \notin B$.

Consider $\theta = \{(0,2)\} \subseteq f$:

- $inc(dom(\theta)) = \{0,1\}$
- $cod(\theta) = \{2\}$

Since $\{0,1\} \neq \{2\}$, we have $\theta \notin B$.

This suggests we need a different approach. Actually, by Rice's theorem, since B is saturated and non-trivial, 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 $C_x = \{x : f(x) \in W_x\}$ is not saturated for injective total computable f

Since f is injective and total computable, define $g : \mathbb{N} \to \mathbb{N}$ by:

```
g(x) = some index for the constant function that outputs f(x)
```

More precisely, by s-m-n theorem, $\exists$ total computable g such that $\varphi g(x)(y) = f(x)$ for all y.

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

This means $\varphi_e(y) = f(e)$ for all y, so $W_e = \mathbb{N}$ and $E_e = \{f(e)\}$.

Since $f(e) \in E_e = \{f(e)\}$ and $W_e = \mathbb{N}$, we have $f(e) \in W_e$, so $e \in C_x$.

Now consider any $e' \neq e$ such that $\varphi_e' = \varphi_e$ (such e' exists since there are infinitely many indices for each function).

We have $\varphi_e' = \varphi_e$, so they compute the same function, but:

- $e \in C_x$ since $f(e) \in W_e = W_e'$
- For $e' \in C_x$, we need $f(e') \in W_e' = \{f(e)\}$

Since $f$ is injective and $e \neq e'$, we have $f(e) \neq f(e')$. So $f(e') \notin \{f(e)\} = W_{e'}$, hence $e' \notin C_x$.

Therefore, $\varphi_e = \varphi_{e'}$, $e \in C_x$, but $e' \notin C_x$, showing $C_x$ is not saturated.