Computability Exam Solutions

July 11, 2012

Exercise 1

URM^s variant with JS(m,n,t) instruction

The URM^s machine eliminates S(n) and J(m,n,t), adding JS(m,n,t) which:

- Compares contents of registers m and n
- If they coincide, jumps to instruction t
- Otherwise, increments register m and executes the next instruction

Relationship between C^s and C:

C^s = C (equal computational power)

Proof:

1. C^s ⊆ C: Every URM^s-computable function is URM-computable.

The JS(m,n,t) instruction can be simulated in standard URM:

```
JS(m,n,t) simulation:
  J(m,n,t)     // if rm = rn, jump to t
  S(m)         // otherwise increment rm
  // continue to next instruction
```

Since URM^s has Z(n) and T(m,n), and JS can be simulated, any URM^s program can be converted to a URM program.

2. C ⊆ C^s: Every URM-computable function is URM^s-computable.

We need to simulate S(n) and J(m,n,t) using available URM^s instructions.

Simulating S(n):

```
S(n) simulation:
  T(n,k)       // copy rn to working register k
  Z(n)         // clear rn
  JS(n,k,end)  // if rn = rk, jump to end (rn = rk = 0)
  // this loop never executes since rn starts at 0
end: // at this point rn = rk + 1 (incremented from JS operations)
```

Actually, this is more complex. Let me use a different approach:

To increment register n, we can use a loop that compares with a counter:

```
// To compute S(n): increment rn by 1
T(n,k)       // save original value
Z(n)         // clear rn
JS(n,k,end)  // while rn ≠ rk, increment rn (this gives rn = rk + 1)
end:
```

Simulating J(m,n,t):

```
J(m,n,t) simulation:
JS(m,n,t)    // if rm = rn, jump to t
T(m,k)       // otherwise, restore rm (since JS incremented it)
Z(m)
// copy back the original value using the saved copy
```

This requires careful register management but is achievable.

Conclusion: $C^s = C$ (equal computational power).

Exercise 2

Theorem: A is recursive ⟺ ∃ total computable f : ℕ → ℕ such that x ∈ A ⟺ f(x) > x

Proof:

(⇒) If A is recursive, then such f exists.

Since A is recursive, χ_a is computable. Define:

```
f(x) = x + χₐ(x)
```

Then f is total and computable, and:

- If $x \in A$: $f(x) = x + 1 > x$
- If $x \notin A$: $f(x) = x + 0 = x \not> x$

Therefore $x \in A \iff f(x) > x$.

(⇐) If such f exists, then A is recursive.

Given total computable f such that $x \in A \iff f(x) > x$, we can compute χ_a:

```
χₐ(x) = {
  1  if f(x) > x
  0  if f(x) ≤ x
}
```

Since f is computable and comparison is computable, χ_a is computable, so A is recursive.

Exercise 3

Classification of $A = \{x : \exists y \in W_x.\ x < f(y)\}$

where f is total computable with infinite image.

A is r.e.:

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

This searches for $y \in W_x$ such that $x < f(y)$.

A is not recursive:

Since img(f) is infinite, we can define an increasing sequence $\{a_i\}_{i \in \mathbb{N}} \subseteq \text{img}(f)$.

We reduce from the halting problem. Define $g : \mathbb{N}^2 \to \mathbb{N}$ by:

```
g(x,y) = {
  a large value > x  if x ∈ K
  ↑                  if x ∉ K
}
```

More precisely, since f has infinite image, for each x we can find some value in img(f) that is > x.

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

- If $x \in K$: $W_{s(x)}$ contains large values, so $\exists y \in W_{s(x)}$ with $s(x) < f(y)$, hence $s(x) \in A$
- If $x \notin K$: $W_{s(x)} = \emptyset$, so $\nexists y \in W_{s(x)}$, hence $s(x) \notin A$

This reduction shows $K \leq_m A$, so A is not recursive.

$\bar{A}$ is not r.e.: Since A is r.e. but not recursive, $\bar{A}$ is not r.e.

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

Exercise 4

Classification of $B = \{x : \varphi_x(0)\uparrow \vee \varphi_x(0) = 0\}$

B is r.e.:

```
scB(x) = 1(μt. S(x,0,0,t))
```

This searches for evidence that $\varphi_x(0) = 0$. If $\varphi_x(0)\uparrow$, the search never terminates, which is correct since $x \in B$ in this case too.

Actually, let me be more careful. We want:

```
x ∈ B ⟺ ϕₓ(0) ↑ ∨ ϕₓ(0) = 0
```

The semi-characteristic function should converge iff $x \in B$.

If $\varphi_x(0) = 0$, then we can detect this by finding a computation that halts with output 0.
If $\varphi_x(0) \uparrow$, then $x \in B$, but we can't detect this directly.

Actually, let's reconsider. We have:

```
x ∈ B ⟺ ϕₓ(0) ≠ n for any n > 0
```

This is equivalent to saying $x \notin B \Longleftrightarrow \varphi_x(0) = n$ for some $n > 0$.

So $\bar{B} = \{x : \varphi_x(0) = n \text{ for some } n > 0\}$ is r.e.:

```
scB̄(x) = 1(μ⟨n,t⟩. n > 0 ∧ S(x,0,n,t))
```

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

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

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

- If $x \notin K$: $\varphi_{s(x)}(0) = 0$, so $s(x) \in B$
- If $x \in K$: $\varphi_{s(x)}(0) \uparrow$, so $s(x) \in B$

This doesn't work since both cases give $s(x) \in B$.

Let me try differently:

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

- If $x \in K$: $\varphi_{s(x)}(0) = 1 \neq 0$ and $\varphi_{s(x)}(0) \downarrow$, so $s(x) \notin B$
- If $x \notin K$: $\varphi_{s(x)}(0) \uparrow$, so $s(x) \in B$

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

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

Exercise 5

Proof using Rice-Shapiro theorem

Given $\mathcal{A} \subseteq \mathcal{C}$ with $0 \notin \mathcal{A}$ and $1 \in \mathcal{A}$ (where 0, 1 are constant functions), and $A = \{x : \varphi_x \in \mathcal{A}\}$, we show either A is not r.e. or $\bar{A}$ is not r.e.

Since $1 \in \mathcal{A}$ and $0 \notin \mathcal{A}$, we have two cases to consider for Rice-Shapiro:

Case 1: A is not r.e. Consider the constant 1 function: $1 \in \mathcal{A}$. Consider any finite subfunction $\theta \subseteq 1$. Since 1 is the constant function, any finite $\theta \subseteq 1$ is either empty or maps some finite domain to $\{1\}$.

The empty function $\emptyset$ has $\mathrm{cod}(\emptyset) = \emptyset$, and whether $\emptyset \in \mathcal{A}$ depends on the specific set $\mathcal{A}$.

If $\emptyset \notin \mathcal{A}$, then we have $1 \in \mathcal{A}$ and finite $\theta = \emptyset \subseteq 1$ with $\theta \notin \mathcal{A}$. By Rice-Shapiro, A is not r.e.

Case 2: $\bar{A}$ is not r.e. Consider the constant 0 function: $0 \notin \mathcal{A}$, so $0 \in \bar{\mathcal{A}}$. Consider finite subfunctions $\theta \subseteq 0$. Again, $\theta = \emptyset$ is a finite subfunction.

If $\emptyset \in \mathcal{A}$, then $\emptyset \notin \bar{\mathcal{A}}$, so we have $0 \in \bar{\mathcal{A}}$ and finite $\theta = \emptyset \subseteq 0$ with $\theta \notin \bar{\mathcal{A}}$. By Rice-Shapiro, $\bar{A}$ is not r.e.

Conclusion: In either case ($\emptyset \in \mathcal{A}$ or $\emptyset \notin \mathcal{A}$), we get that either A is not r.e. or $\bar{A}$ is not r.e.