Computability Exam Solutions

April 3, 2012

Exercise 1

Definition of $A \leq_m B$ (many-one reducibility)

Given sets $A, B \subseteq \mathbb{N}$, we say that $A \leq_m B$ (A is many-one reducible to B) if there exists a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all $x \in \mathbb{N}$:

```
x ∈ A ⟺ f(x) ∈ B
```

a. Proof of transitivity: If $A \leq_m B$ and $B \leq_m C$ then $A \leq_m C$

Since $A \leq_m B$, $\exists$ total computable $f : \mathbb{N} \to \mathbb{N}$ such that $\forall x$: $x \in A \iff f(x) \in B$.

Since $B \leq_m C$, $\exists$ total computable $g : \mathbb{N} \to \mathbb{N}$ such that $\forall y$: $y \in B \iff g(y) \in C$.

Define $h = g \circ f$. Since f and g are total and computable, h is total and computable.

For any $x \in \mathbb{N}$:

```
x ∈ A ⟺ f(x) ∈ B ⟺ g(f(x)) ∈ C ⟺ h(x) ∈ C
```

Therefore $A \leq_m C$ via reduction function h.

b. Proof: If $A \neq \mathbb{N}$ then $\emptyset \leq_m A$

Since $A \neq \mathbb{N}$, $\exists a \in \mathbb{N}$ such that $a \notin A$.

Define $f : \mathbb{N} \to \mathbb{N}$ by $f(x) = a$ for all x.

Since f is the constant function with value a, f is total and computable.

For any $x \in \mathbb{N}$:

```
x ∈ ∅ ⟺ False ⟺ a ∉ A ⟺ f(x) ∉ A
```

Therefore $\emptyset \leq_m \bar{A}$ via f. But we want $\emptyset \leq_m A$.

Let me reconsider. We want f such that $x \in \emptyset \iff f(x) \in A$. Since $\emptyset$ is empty, we need $f(x) \notin A$ for all x.

Since $A \neq \mathbb{N}$, pick any $a \notin A$. Define $f(x) = a$. Then:

```
x ∈ ∅ ⟺ False ⟺ a ∉ A ⟺ f(x) ∉ A
```

This gives $\emptyset \leq_m \bar{A}$, not A.

Actually, for $\varnothing \leq_m A$ to hold, we need: $x \in \varnothing \iff f(x) \in A$. Since no x is in $\varnothing$, we need $f(x) \notin A$ for all x. This is possible when $A \neq \mathbb{N}$.

Wait, let me think again. We have $\varnothing$ = the empty set. For $\varnothing \leq_m A$, we need a function f such that: $x \in \varnothing \iff f(x) \in A$

Since $x \in \varnothing$ is always false, we need $f(x) \in A$ to be always false, i.e., $f(x) \notin A$ for all x.

Since $A \neq \mathbb{N}$, there exists some element $a \notin A$. Define $f(x) = a$ for all x. Then $f(x) \notin A$ for all x, so the equivalence holds.

Therefore $\varnothing \leq_m A$.

Exercise 2

Question: Does there exist a quasi-total computable function $f : \mathbb{N} \to \mathbb{N}$ such that $f \subseteq \chi_k$?

A function f is quasi-total if it is undefined on a finite set of points.

Answer: No, such a function cannot exist.

Proof: Suppose f is quasi-total, computable, and $f \subseteq \chi_k$. Since f is quasi-total, ∃ finite set F such that f is defined on $\mathbb{N} \setminus F$.

Since $f \subseteq \chi_k$, whenever f(x) is defined, $f(x) = \chi_k(x)$.

This means that for all $x \in \mathbb{N} \setminus F$, we can compute $\chi_k(x) = f(x)$.

Now consider the algorithm:

```
For input x:
  if x ∈ F:
    // F is finite, so membership is decidable
    compute χₖ(x) by brute force (check if ϕₓ(x) ↓)
  else:
    return f(x) = χₖ(x)
```

This would give us a total algorithm for computing χ_k, contradicting the fact that K is not recursive.

Therefore, no such quasi-total computable function f exists.

Exercise 3

Classification of $B = \{\pi(x,y) : P_x(x) \downarrow$ in less than y steps$\}$

B is r.e.:

```
scB(z) = 1(μt. let (x,y) = π⁻¹(z) in H(x,x,t) ∧ t < y)
```

This searches for evidence that $P_x(x)$ terminates in fewer than y steps.

B is recursive:

```
χB(z) = let (x,y) = π⁻¹(z) in:
        if y = 0 then 0  // no computation terminates in < 0 steps
        else χH(x,x,y-1)  // check if terminates in exactly y-1 steps or fewer
```

Since H is decidable and π^{-1} is computable, χB is computable.

B̄ is recursive: Since B is recursive, B̄ is also recursive.

Final classification: B and B̄ are both recursive.

Exercise 4

Classification of A = {x ∈ ℕ : ∃k > 0. φ_x symmetric in [0,2k]}

A function f is symmetric in [0,2k] if dom(f) ⊇ [0,2k] and ∀y ∈ [0,k]: f(y) = f(2k-y).

A is r.e.:

```
scA(x) = 1(μ⟨k,t⟩. k > 0 ∧ ∀y≤k ∀s≤t [S(x,y,f(y),s) ∧ S(x,2k-y,f(2k-y),s) → f(y) = f(2k-y)])
```

This can be made more precise using the step predicate S to verify that φ_x is defined on [0,2k] and satisfies the symmetry condition.

A is not recursive: The set A is saturated since it expresses a property of functions. By Rice's theorem, since A is non-trivial (neither empty nor the whole set), A is not recursive.

To see A ≠ ∅: The constant function $\varphi_e(x) = 0$ is symmetric in any interval [0,2k].
To see A ≠ ℕ: The identity function is not symmetric in [0,2] since id(0) = 0 ≠ 2 = id(2).

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 5

Second Recursion Theorem

For every total computable function f : ℕ → ℕ, there exists e_0 ∈ ℕ such that:

```
ϕₑ₀ = ϕf(e₀)
```

Proof that g(x) = e_0 if φ_x total, e_1 otherwise is not computable

where e_0 is an index for ∅ and e_1 is an index for the constant 1 function.

Proof: Suppose g were computable. Define h : ℕ → ℕ by h(x) = g(x).

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

Case 1: φ_e is total.

Then $g(e) = e_0$, so $\varphi_e = \varphi_{e0} = \varnothing$ (everywhere undefined).

But this contradicts φ_e being total.

Case 2: φ_e is not total.

Then $g(e) = e_1$, so $\varphi_e = \varphi_{e1} = 1$ (constant 1 function).

But the constant 1 function is total, contradicting φ_e not being total.

Both cases lead to contradictions, so g cannot be computable.