Final Exam Review Problems

Problem 1. Show that the following functions are register machine computable: $n!$, n^m and $\max\{n, m\}$.

Solution. I will leave you to think about writing these register machine programs.

Problem 2(a). Prove that $A \leq_m A$ for every set A.

2(b). Prove that if $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$.

2(c). Let $E = \{2n : n \in \mathbb{N}\}$ be the set of even numbers and $O = \{2n+1 : n \in \mathbb{N}\}$ be the set of odd numbers. Prove that $E \leq_m O$ and $O \leq_m E$. Explain why this example shows that $\leq_m$ is not an anti-symmetric relation.

Solution. For 2(a), use the identity function $f(n) = n$.

$$a \in A \Leftrightarrow f(n) = n \in A$$

For 2(b), let f and g be the functions witnessing $A \leq_m B$ and $B \leq_m C$ respectively. That is, $n \in A \Leftrightarrow f(n) \in B$ and $n \in B \Leftrightarrow g(n) \in C$. Let $h = g \circ f$ be the composition of f and g. Since f and g are total computable functions, h is also a total computable function. It follows that $A \leq_m C$ because

$$a \in A \Leftrightarrow f(n) \in B \Leftrightarrow g(f(n)) = (g \circ f)(n) \in C$$

For 2(c), the function $f(n) = n + 1$ works to show both $E \leq_m O$ and $O \leq_m E$ because $n \in E \Leftrightarrow n + 1 \in O$ and $n \in O \Leftrightarrow n + 1 \in E$. Recall that an anti-symmetric relation is a binary relation R with the property that if $R(i, j)$ and $R(j, i)$ both hold, then $i = j$. For $\leq_m$, we have $E \leq_m O$ and $O \leq_m E$ but $E \neq O$, so $\leq_m$ is not antisymmetric.

Problem 3. Define a new relation $\equiv_m$ between sets as follows:

$$A \equiv_m B \Leftrightarrow A \leq_m B \text{ and } B \leq_m A.$$

Prove that $\equiv_m$ is an equivalence relation. That is, $\equiv_m$ is reflexive, symmetric and transitive.

Solution. Reflexive. $A \equiv_m A$ because $A \leq_m A$ by Problem 2(a).

Symmetric. Assume $A \equiv_m B$ holds, and we show $B \equiv_m A$ holds. Since $A \equiv_m B$, we have $A \leq_m B$ and $B \leq_m A$. Therefore, both $B \leq_m A$ and $A \leq_m B$ hold, which means $B \equiv_m A$.

Transitive. Assume $A \equiv_m B$ and $B \equiv_m C$, and we show $A \equiv_m C$. Since $A \equiv_m B$, we know $A \leq_m B$ and $B \leq_m A$. Since $B \equiv_m C$, we know $B \leq_m C$ and $C \leq_m B$. Putting these together, using the transitivity of $\leq_m$ from Problem 2(b), we have

$$A \leq_m B \text{ and } B \leq_m C \Rightarrow A \leq_m C$$
$$C \leq_m B \text{ and } B \leq_m A \Rightarrow C \leq_m A$$

Together $A \leq_m C$ and $C \leq_m A$ imply $A \equiv_m C$.

Problem 4(a). Prove that $K_1 \leq_m K_0$ and that $K_0 \leq_m K$.

4(b). Using our results from class, show that $K \equiv_m K_0 \equiv_m K_1$.

Solution. For 4(a), first consider $K_1 \leq_m K_0$. Let $f(e) = \langle e, 0 \rangle$. Since our pairing function is computable, f is computable. This shows $K_1 \leq_m K_0$ by

$$e \in K_1 \Leftrightarrow \varphi_e(0) \downarrow \Leftrightarrow \langle e, 0 \rangle = f(e) \in K_0$$

Next, consider $K_0 \leq_m K$. Remember that each number x codes a pair $\langle x_0, x_1 \rangle$. Define a partial computable function $f(x, y)$ by

$$f(x,y) = \begin{cases} 1 & \text{if } \varphi_{x_0}(x_1) \downarrow \text{ where } x = \langle x_0, x_1 \rangle \\ \uparrow & \text{otherwise} \end{cases}$$

Note that f is computable because it takes inputs x and y, splits x into its coded pair $\langle x_0, x_1 \rangle$, decodes the x_0-th register machine from x_0 and runs it on input x_1. If this halts, f outputs 1 and otherwise it runs forever waiting for $\varphi_{x_0}(x_1)$ to halt. Note that the input y is ignored, so the value of the function $f(x, y)$ (and whether or not it halts) depends only on x.

By the s-m-n theorem, there is a total computable function $s(x)$ such that $\varphi_{s(x)}(y) = f(x, y)$ as partial functions. (Namely, on any input y, either both $\varphi_{s(x)}(y)$ and $f(x, y)$ diverge or both converge to the same value.) Consider the function $f(x, y)$ for a fixed value of x. Because $f(x, y)$ does not depend on y, we have the following property: $f(x, y) \downarrow$ for a single value of y if and only if $f(x, y) \downarrow$ for all values y.

By the s-m-n theorem, there is a computable function $s(x)$ such that for all x, $\varphi_{s(x)}(y) = f(x, y)$ as partial functions. We claim that the function $s(x)$ gives the reduction of K_0 to K.

$$\begin{aligned} \langle e, n \rangle \in K_0 &\Leftrightarrow \varphi_e(n) \downarrow \\ &\Leftrightarrow f(\langle e, n \rangle, y) \downarrow \text{ for all } y \\ &\Leftrightarrow \varphi_{s(\langle e,n \rangle)}(y) \downarrow \text{ for all } y \\ &\Leftrightarrow \varphi_{s(\langle e.n \rangle)}(s(\langle e, n \rangle)) \downarrow \\ &\Leftrightarrow s(\langle e, n \rangle) \in K \end{aligned}$$

Problem 5. Consider the set $\text{Tot} = \{e : \varphi_e \text{ is total}\} = \{e : \varphi_e(n) \downarrow \text{ for all } n\}$. Prove that Tot is not computable.

Solution. You can solve this problem in a variety of different ways using the techniques from the past couple of weeks.

Method 1. Assume for a contradiction that Tot is computable. This means that the characteristic function χ_{Tot} defined by

$$\chi_{\text{Tot}}(e) = \begin{cases} 1 & \text{if } \varphi_e(n) \downarrow \text{ for all } n \\ 0 & \text{otherwise} \end{cases}$$

is computable. Therefore, we can define a computable function $g(e)$ by

$$g(e) = \begin{cases} \varphi_e(e) + 1 & \text{if } \chi_{\text{Tot}}(e) = 1 \\ 0 & \text{if } \chi_{\text{Tot}}(e) = 0 \end{cases}$$

Note that if $\chi_{\text{Tot}}(e) = 1$, then φ_e is total, so in particular, $\varphi_e(e) \downarrow$. Therefore, g is a total computable function.

To finish the proof, we need to derive our contradiction. Since g is a computable function, it has an index, say $g = \varphi_i$. Since g is total, we know φ_i is total and hence $\chi_{\text{Tot}}(i) = 1$. Therefore, $g(i) = \varphi_i(i) + 1 \neq \varphi_i(i)$, which contradicts $g = \varphi_i$.

Method 2. For this method, we use the s-m-n theorem to give a reduction $K \leq_m \text{Tot}$ showing that Tot is not computable. Define a partial computable function $g(e, n)$ by

$$g(e, n) = \begin{cases} 1 & \text{if } \varphi_e(e) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Note that the value of $g(e, n)$, as well as whether it converges, does not depend on n. By the s-m-n theorem, there is a total computable function $s(e)$ such that for all e and n, $\varphi_{s(e)} = g(e, n)$ as partial functions. Note that if $\varphi_e(e) \downarrow$, then $g(e, n) \downarrow$ for all n, so $\varphi_{s(e)}(n) \downarrow$ for all n. On the other hand, if $\varphi_e(e) \uparrow$, then $g(e, n) \uparrow$ for all n and hence $\varphi_{s(e)}(n) \uparrow$ for all n. We can now show that the function $s(x)$ gives us the reduction:

$$\begin{aligned} e \in K &\Leftrightarrow \varphi_e(e) \downarrow \\ &\Leftrightarrow \varphi_{s(e)}(n) \downarrow \text{ for all } n \\ &\Leftrightarrow s(e) \in \text{Tot} \end{aligned}$$

Method 3. As a third method, we can solve this problem using Rice's theorem. Notice that $\text{Tot} \neq \emptyset$ because there are total computable functions – for example, $f(x) = x + 1$ is a total computable function. On the other hand, $\text{Tot} \neq \mathbb{N}$ because there are partial computable functions which are not total – for example, the function f such that $f(n) \uparrow$ for all n is partial computable but not total. Therefore, if we can show that Tot is an index set, it will follow from Rice's theorem that Tot is not computable. To see Tot is an index set, suppose $e \in Tot$ and $i \sim e$. That is, φ_e is total and $\varphi_i = \varphi_e$ as partial functions. Since $\varphi_e(n) \downarrow$ for all n, it follows that $\varphi_i(n) \downarrow$ for all n because $\varphi_e = \varphi_i$. So, φ_i is total and $i \in \text{Tot}$ as required.

Problem 6. Consider the following partial computable function.

$$g(e) = \begin{cases} \varphi_e(e) + 1 & \text{if } \varphi_e(e) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Prove that there is no (total) computable function f such that f is an extension of g.

Solution. Suppose for a contradiction that there is a total computable function f such that $f(e) = g(e)$ for all e such that $g(e) \downarrow$. Since f is a computable function, it has an index –

say $f(x) = \varphi_i(x)$. Since f is total, φ_i is total and hence $\varphi_i(i) \downarrow = f(i)$. Because $\varphi_i(i) \downarrow$, we have $g(i) = \varphi_i(i) + 1$. Since $g(i) \downarrow$ and f is an extension of g, $f(i) = g(i)$. We now have the following contradiction:

$$\varphi_i(i) = f(i) = g(i) = \varphi_i(i) + 1$$

Problem 7(a). Consider the set $\text{Ext} = \{e : \varphi_e \text{ has a total computable extension}\}$. Prove that Ext is an index set.

7(b). Prove that Ext is not computable.

Solution. For 7(a), assume that $e \in \text{Ext}$ and $i \sim e$. We need to show that $i \in \text{Ext}$. Since $e \sim i$, we know $\varphi_e = \varphi_i$ as partial functions. In particular, $\text{domain}(\varphi_e) = \text{domain}(\varphi_i)$ and for all $n \in \text{domain}(\varphi_e)$, $\varphi_e(n) = \varphi_i(n)$.

Since $e \in \text{Ext}$, there is a total computable function f such that f is an extension of φ_e. This means that for all $n \in \text{domain}(\varphi_e)$, $\varphi_e(n) = f(n)$. However, since $\text{domain}(\varphi_i) = \text{domain}(\varphi_e)$, we have that for all $n \in \text{domain}(\varphi_i)$, $\varphi_i(n) = \varphi_e(n) = f(n)$. Therefore, f is also an extension of φ_i and so $i \in \text{Ext}$.

For 7(b), we would like to apply Rice's theorem. We already know Ext is an index set, so it suffices to show Ext$=\neq \emptyset$ and Ext$\neq \mathbb{N}$. Problem 7(a) shows that Ext$\neq \mathbb{N}$. To see Ext$=\neq \emptyset$, notice that if f is any total computable function (for example, $f(x) = x+1$), then every index for f is in Ext because f is a total computable extension of itself (so f has a total computable extension). Therefore, Rice's theorem applies and we conclude that Ext is not computable.

Problem 8(a). Consider the set $\text{Fin} = \{e : \varphi_e \text{ has a finite domain}\}$. Prove that Fin is an index set.

8(b). Prove that Fin is not computable.

Solution. For 8(a), assume that $e \in \text{Fin}$ and $i \sim e$. We need to show that $i \in \text{Fin}$. Since $e \sim i$, we know $\varphi_e = \varphi_i$ as partial functions. In particular, $\text{domain}(\varphi_e) = \text{domain}(\varphi_i)$. Since $e \in \text{Fin}$, $\text{domain}(\varphi_e)$ is finite. Therefore, $\text{domain}(\varphi_i)$ is finite and hence $i \in \text{Fin}$.

For 8(b), we apply Rice's theorem. By 8(a), Fin is an index set. We know $\text{Fin} \neq \emptyset$ because the function f such that $f(n) \uparrow$ for all n is partial computable and has empty (and hence finite) domain. Therefore every index for f is in Fin. On the other hand, $\text{Fin} \neq \mathbb{N}$ because there are total computable functions (such as $f(x) = x + 1$). Any such function has infinite domain and hence each of its indices is not in Fin. Therefore, Rice's theorem applies to Fin and we conclude Fin is not computable.

Problem 9. Let A be a nonempty c.e. set. Prove that there is a total computable function g such that $A = \text{range}(g)$.

Solution. Since $A \neq \emptyset$, we can fix an element $a \in A$. Since A is c.e., there is a partial computable function φ_e such that $A = \text{domain}(\varphi_e)$. We define a total computable function $g(x)$ as follows. We interpret each input x as a code for a pair $\langle x_0, x_1 \rangle$. Then we compute

$\varphi_{e,x_0}(x_1)$. That is, we run the algorithm for $\varphi_e(x_1)$ for x_0 many steps. If this computation converges (i.e hits a halt instruction), we output x_1. If the computation doesn't converge, we output a. That is,

$$g(\langle x_0, x_1 \rangle) = \begin{cases} x_1 & \text{if } \varphi_{e,x_0}(x_1)\downarrow \\ a & \text{otherwise} \end{cases}$$

We need to show that range(g) = domain(φ_e), or equivalently range$(g) = A$. First, suppose $m \in$ range(g) and we show $m \in A$. There are two possibilities for m. First, we could have $m = a$, in which case $m \in A$ because $a \in A$. Second, we could have $m = x_1$ in some pair $\langle x_0, x_1 \rangle$ such that $\varphi_{e,x_0}(x_1)\downarrow$. In this case, $m = x_1 \in$ domain(φ_e) and so $m \in A$.

Next, assume $m \in A$ and we show $m \in$ range(g). Since $m \in A =$ domain(φ_e), we know there is some number of computation steps s such that $\varphi_{e,s}(m)\downarrow$. By the definition of g, we have $g(\langle s, m \rangle) = m$ because $\varphi_{e,s}(m)\downarrow$. Therefore, $m \in$ range(g) as required.

Problem 10. Let A be an infinite c.e. set. Prove that there is a total computable function g such that g is one-to-one and $A =$ range(g).

Solution.