Midterm (Answer Key)
CS 590: Theory of Computation
12 March (Lecture No. 22)

This is a 50 minute, closed book examination—though you may bring a single 8.5x11 inch (or A4) sheet of paper with whatever you want written on it. There are 100 possible points, seven questions, and five pages.
Show your work. Partial credit will be given for incorrect answers, provided that the logic behind the answer is clear and shows an understanding of the subject matter. More complete answers (e.g. proofs) will earn more points.

Question 1 (10 points) Suppose that $F$ is a finite set. Is it recursive? Recursively enumerable?

**Answer** Every finite set is recursive, and every recursive set is r.e. So, this set is both. Suppose that $F = \{a_1, \ldots, a_k\}$. Here is an algorithm for $\chi_A$:

```
Input $x$
If $x = a_1$
    then return 1
elseif $x = a_2$
    then return 1
elseif ...
    elseif $x = a_k$
    then return 1
else return 0
```

Question 2 (20 points) Consider the set $\text{Fin} = \{\epsilon : W_\epsilon \text{ is finite}\}$. Is Fin recursive? Is it recursively enumerable?

**Answer** Fin is clearly a non-trivial index set, since whenever $\epsilon \in \text{Fin}$ and $\phi_\epsilon = \phi_a$ then $W_\epsilon = W_a$ and so $W_a$ is finite which implies that $a \in \text{Fin}$.

So, by Rice’s theorem, Fin is not recursive.

Also, if it were r.e., then by the Rice-Shapiro theorem every extension of any finite function in Fin would also have an index in Fin. But, all the functions in Fin are, by definition, finite. So any infinite extension of one cannot be in Fin. This contradicts the Rice-Shapiro theorem, showing that Fin is not r.e., either.
Question 3 (15 points) Consider the set \( \text{Cof} = \{ e : \overline{W_e} \text{ is finite} \} \). It is a fact that \( K \leq_m \text{Cof} \) but that \( \text{Cof} \not\leq_m K \).

Now, let \( f \) be a total function such that \( \text{Cof} = \{ f(0), f(1), \ldots \} \) (such a function does exist). Is \( f \) computable? Why or why not?

Answer No. If \( f \) were computable, then \( \text{Cof} \) would be r.e., since \( f \) would recursively enumerate it. But then \( \text{Cof} \leq_m K \), since \( K \) is r.e. complete. This contradicts what we are given.

Question 4 (10 points) Let \( S = \{ e : e \not\in W_e \} \). Is \( S \) r.e.? Why or why not?

Answer \( S \) is not r.e., since it is defined by diagonalizing over the r.e. sets.

Suppose to the contrary that \( S \) is r.e. Then there is some \( a \) such that \( S = W_a \). Is \( a \in W_a \)?

If \( a \in W_a \), then by definition of \( S \) we know that \( a \not\in W_a \), which is a contradiction. If \( a \not\in W_a \), then we know that \( a \in S \) and therefore \( a \in W_a \), which is another contradiction.

So, there can be no such \( a \), and hence \( S \) cannot be r.e.

Notice that \( S \) is not and index set, so the Rice-Shapiro theorem does not apply.

An even easier answer would be to point out that \( S = \overline{K} \), and we know that \( \overline{K} \) is not r.e., since if it were that would imply that \( K \) is recursive.

Question 5 (15 points) Show that \( K \leq_m \text{TOT} \).

Answer By the smm theorem, define a total, computable function \( s(x) \) as:

\[
\phi_{s(x)}(y) = \begin{cases} 
0 & \text{if } x \in K \\
\uparrow & \text{otherwise}
\end{cases}
\]

\( x \in K \) implies that \( \phi_{s(x)} = 0 \), which it total, so that \( s(x) \in \text{TOT} \). \( x \not\in K \) implies that \( \phi_{s(x)} \uparrow \) for all inputs, so that it is not total, and therefore \( s(x) \not\in \text{TOT} \). In other words, \( x \in K \) iff \( s(x) \in \text{TOT} \), which shows that \( s : K \leq_m \text{TOT} \) as was desired.
Question 6 (10 points) Here is a fallacious “proof” that $\overline{K} \leq_m K$.

By the smn theorem, define a total, computable function $s(x)$ as:

$$\phi_{s(x)}(y) = \begin{cases} 
  y & \text{if } x \in \overline{K} \\
  \uparrow & \text{otherwise}
\end{cases}$$

If $x \in \overline{K}$, then $\phi_{s(x)}$ is the identity function, and so $\phi_{s(x)}(s(x)) = s(x)$ which implies $s(x) \in K$. If $x \notin \overline{K}$, then $\phi_{s(x)} \uparrow$ for all inputs, so that in particular $\phi_{s(x)}(s(x)) \uparrow$ and $s(x) \notin K$. In other words, $x \in \overline{K}$ iff $s(x) \in K$, which shows that $s : \overline{K} \leq_m K$ as was claimed.

How do you know that the proof must be in error? What is the error?

**Answer**

If it were correct, then $\overline{K}$ would be r.e. by downward closure of r.e. sets under m-reduction. But if that were true, then $K$ would be recursive, which it cannot be by diagonalization.

The error is that the predicate “$x \in \overline{K}$” is not even partially decidable. You would have to verify that $\phi_{s(x)} \uparrow$, and this cannot be done by an algorithm. So, the smn theorem does not apply to this function. This means that the function $s$ is not well-defined—in fact, it cannot exist.

Question 7 (20 points) Give one example of each of the following, and describe why your answer is correct.

1. A decidable predicate
2. A recursively enumerable set which is not recursive
3. A finite function
4. A non-total function which extends the finite function from your last answer
5. A computable, non-total function

**Answer** Several answers are possible. Here are some samples.

1. “is $x$ an even number” is decidable because a simple algorithm exists to take input $x$ and determine whether or not $x$ is even.
2. $K$ is a non-recursive r.e. set. It is r.e. because it’s partial characteristic function is computed by the following algorithm:

\begin{align*}
\text{input } x \\
\text{Simulate } P_x \text{ on input } x \\
\text{If it converges, return 1}
\end{align*}

$K$ is not recursive, because if it were, the following function would be computable which it cannot be by diagonalization:

$$d(x) = \begin{cases} 
\phi_x(x) + 1 & \text{if } x \in K \\
0 & \text{otherwise}
\end{cases}$$

3. Any function which converges on a finite number of inputs, such as the function $f(x) \uparrow$ for all $x$, is finite.

A more interesting example would be $f'(x) = 0$ if $x = 0$.

4. Any finite function extends the function $f$ from the last question. For example, the function $g(x) = x$ if $x \leq 1$ works since for every $x$ where $f(x) \downarrow$ (there are none) it is the case that $f(x) = g(x)$.

Also, $f' \subseteq g$, since for all $x$ where $f'(x) \downarrow$, $f'(x) = g(x)$ (namely, $f(0) = g(0) = 0$).

Consider also $g'(x) = x$ if $x$ is even. This is another non-total function (since $g'(x) \uparrow$ when $x$ is odd) which extends $f$ and $f'$. However, $g'$ is not a finite function.

5. All of the above function ($f, f', g, g'$) are computable and non-total. They are computable, since simple algorithms exist for them. They are non-total since each has at least one value for which it is undefined.