Lecture #10: First Hard Problems
Exercises and Review

Questions for Review

The "Halting Predicate" is Not Computable

1. Define (using reasonably simple English, and a bit of mathematical notation) the "halting predicate," HALT : $\mathbb{N}^2 \to \{0, 1\}$.
2. A significant result about this predicate, proved during the lecture is that this is ***not*** Turing-computable.
   (a) Identify the proof technique (something you learned about in MATH 271 or 273, if not before that) that is being used here.
   (b) As part of the proof, an assumption was used to establish that a macro could be used in programs, that could compute the HALT predicate, and a very short, and simple program was written and used to obtain a contradiction. Give (or, at least, describe) the program that was used in this argument.
   (c) Finally, describe the contradiction that was obtained, to complete the proof that the HALT predicate is not computable.

Parameter Theorem

3. Define the partial function $\Phi^{(n)} : \mathbb{N}^{n+1} \to \mathbb{N}$, and the function $\Phi_y^{(n)} : \mathbb{N}^n \to \mathbb{N}$ for a positive integer k and for $y \in \mathbb{N}$.
4. State the ***Universality Theorem***.
5. State the ***Parameter Theorem***, and explain briefly how this is proved.

Computations with Sets

6. Define the ***characteristic function*** of a set $S \subseteq \mathbb{N}^n$, for a positive integer n.
7. Define a ***computable set***. What are these also called, in older references?
8. Define a ***computably enumerable set***. What else are these called, in older references?
9. Sketch a proof that every ***computable*** set is ***computable enumerable***. That is, in a sentence or two, say how this is established.
10. Suppose that both a set $S \subseteq \mathbb{N}$ and its complement, $\overline{S}$, are both computably enumerable. What else can be concluded about S? Sketch a proof of this (or, at least, identify the proof technique used).
11. Suppose that $S, T \subseteq \mathbb{N}$ and both S and T are computably enumerable. What can be concluded about $S \cup T$ and $S \cap T$?
12. Define the set W_n.
13. State the ***Enumeration Theorem*** and explain, briefly, why this theorem is true.
14. Define the set K.
15. Sketch a proof that K is computable enumerable. That is, in a sentence or two, say how this is proved.
16. Sketch a proof that the complement $\overline{K}$ of K is ***not*** computably enumerable.
17. What else does the above result imply about the set K?
18. After consideration of the above, suppose that a set $S \subseteq \mathbb{N}$ is computably enumerable. What can — or ***cannot*** — necessarily be concluded about $\overline{S}$?
19. State three alternative — but provably equivalent — alternative characterizations of nonempty computably enumerable subsets of $\mathbb{N}$.

Practice Problems

These problems — which continue activities in the lecture presentation — will not be discussed during the lecture, and solutions for these problems will not generally be made available. They can be used as "practice" problems that can help you to practice skill considered in the lecture presentation for Lecture #10.

1. Let

$$\text{Total} = \{n \in \mathbb{N} \mid \text{the } \mathcal{S}\text{-program } \mathcal{P}, \text{ such that } \#(\mathcal{P}) = n, \text{ computes a } \textbf{\textit{total}} \text{ function } f_n : \mathbb{N} \to \mathbb{N} \text{ (of one variable)}\}.$$

The goal of this first question is to prove that the set "Total" is not computably enumerable.

(a) Assume — to obtain a contradiction — that the set "Total" is computably enumerable. Using a result from the lecture, show that there must exist a ***computable total function*** $g : \mathbb{N} \to \mathbb{N}$ such that "Total" is the range of g, that is,

$$\text{Total} = \{g(n) \mid n \in \mathbb{N}\}.$$

Note, then, that

$$g(0), g(1), g(2), g(3), \ldots$$

is an enumeration of the encodings of all $\mathcal{S}$-programs that compute total functions of one variable.[1]

(b) Consider the function $h : \mathbb{N} \to \mathbb{N}$ such that

$$h(x) = \Phi^{(1)}(x, g(x)) + 1.$$

Sketch a (very brief) proof that h is a computable total function.

(c) Now give a "Diagonalization" argument, using what has been established, above, to prove that h ***cannot*** be a computable total function.

(d) Explain why it follows that the set "Total" is not computably enumerable. (This should be easy.)

2. Prove that if $R, S \subseteq \mathbb{N}$ and R and S are both computably enumerable, then so is $R \cup S$.

Hint: This problem was mentioned during Lecture #10, immediately after a result — whose proof is included in the supplemental material for the lecture, and whose proof can be modified in order to solve this problem. It involves a proof technique that you might (or might not) have seen in a prerequisite for this course.

3. By the end of Lecture #10 it had been proved that a ***nonempty*** set $S \subseteq \mathbb{N}$ was computably enumerable if and only if S was the range of some computable *total* function $f : \mathbb{N} \to \mathbb{N}$.

Things change if f is required to satisfy an additional property.

(a) Suppose, again, that f is a computable total function — but also that f is ***increasing***, that is, $f(n+1) > f(n)$ for every integer $n \geq 0$. Prove that if S is the range of f, that is, $S = \{f(n) \mid n \in \mathbb{N}\}$, then S is a ***computable*** set.

(b) Suppose, in addition, that f is both increasing and ***primitive recursive***, and S is the range of f. Show that S is a ***primitive recursive*** set.

[1] $\mathcal{S}$-programs that end with the unlabelled dummy statement "$Y \leftarrow Y$" do not need to be considered here, because each of these computes the same total function as one of the programs that is included in this listing.

The final problems in this exercise are intended to help to explain the decision to restrict attention to subsets of $\mathbb{N}$ instead of considering subsets of $\mathbb{N}^n$, for all positive integers n. The material here is based on a discussion of this in Davis, Sigal and Weyuker's text, *Computability, Complexity and Languages*.

Let $\mathcal{C}$ be a set (or "class") of partial or total functions $f : \mathbb{N}^n \to \mathbb{N}$ (for a positive integer n). $\mathcal{C}$ is said to be a ***PRC class*** if $\mathcal{C}$ includes the initial functions and is closed under both composition and primitive recursion.

Three different PRC classes can now be identified using results established, so far, in CPSC 513:

- the set of all primitive recursive functions,
- the set of all computable total functions, and
- the set of all computable (partial or total) functions.

4. Prove that if $S_1, S_2 \subseteq \mathbb{N}^n$ and $\mathcal{C}$ is a PRC class of functions such that S_1 and S_2 are in $\mathcal{C}$ then
   (a) $S_1 \cap S_2$ is in $\mathcal{C}$,
   (b) $S_1 \cup S_2$ is in $\mathcal{C}$, and
   (c) $\overline{S_1} = \{(x_1, x_2, \ldots, x_n) \in \mathbb{N}^n \mid (x_1, x_2, \ldots, x_n) \notin S_1\}$ is in $\mathcal{C}$.

5. Suppose $\mathcal{C}$ is a PRC class, $S \subseteq \mathbb{N}^n$, and

$$T = \{[x_1, x_2, \ldots, x_n] \mid (x_1, x_2, \ldots, x_n) \in S\} \subseteq \mathbb{N}.$$

   Prove that S is in $\mathcal{C}$ if and only if T is also in $\mathcal{C}$.

The above result establishes that when considering the "computability" of a subset S of $\mathbb{N}^n$ it suffices to consider the "computability" of the corresponding set

$$T = \{[x_1, x_2, \ldots, x_n] \mid (x_1, x_2, \ldots, x_n) \in S\} \subseteq \mathbb{N}$$

instead. Hence — to simplify presentations — we generally restrict attention to subsets of $\mathbb{N}$, when we consider the computability of sets.