CPSC 513 — Assignment #2
Solutions for Question #1

Recall that, for $n \geq \mathbb{N}$, Φ_n is the (partial or total) function that is computed by the program with encoding n. The problems on this assignment concern the set

$$\text{Monotone} = \{n \in \mathbb{N} \mid \Phi_n \text{ is total and } \Phi_n(x) \leq \Phi_n(x+1) \text{ for all } x \in \mathbb{N}\}.$$

In this question you were asked to assume — in order to obtain a contradiction — that Monotone is a recursive set.

(a) You were first asked to use this to prove that the function $f : \mathbb{N} \to \mathbb{N}$ is a computable total function, where $f(x) < f(x+1)$ for all $x \in \mathbb{N}$ and

$$\text{Monotone} = \{f(0), f(1), f(2), f(3), \dots\}.$$

Solution: Since Monotone is (by assumption) a recursive set its characteristic function $p_{\text{Monotone}} : \mathbb{N} \to \{0, 1\}$ such that, for $n \in \mathbb{N}$,

$$p_{\text{Monotone}}(n) = \begin{cases} 1 & \text{if } n \in \text{Monotone} \\ 0 & \text{otherwise} \end{cases}$$

is a computable total function. Consider the following $\mathcal{L}$ program — which uses a macro to compute this function:

$$\begin{array}{ll}
 & Y \leftarrow 0 \\
[A_1] & Z_3 \leftarrow p_{\text{Monotone}}(Y) \\
 & \text{IF } Z_3 \neq 0 \text{ GOTO } B_1 \\
 & Y \leftarrow Y + 1 \\
 & \text{GOTO } A_1 \\
[B_1] & \text{IF } X_1 \neq 0 \text{ GOTO } C_1 \\
 & \text{GOTO } D_1 \\
[C_1] & X_1 \leftarrow X_1 - 1 \\
 & Y \leftarrow Y + 1 \\
 & \text{GOTO } A_1
\end{array}$$

It is reasonably easy to prove the following by induction on n: For all $n \in \mathbb{N}$, if this program is executed with input n then the statement with label B_1 is executed exactly $n+1$ times. Furthermore, if $1 \leq i \leq n$ then, at the i^{th} execution of this statement, X_1 has value $n+1-i$ and Y has value $f(i-1)$.

It follows that, at the $n+1^{\text{st}}$ execution of this statement, X_1 has value 0 and Y has value $f(n)$ — so that the test fails and the statement that follows it is reached and executed. Since there is no statement with label D, the program now terminates — returning the value $f(n)$ on input n.

Thus f is a computable function.

(b) You were next asked to use this to prove that the function $g : \mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$,

$$g(n) = \sum_{i=0}^{n} \phi_{f(i)}(n)$$

is a a computable *monotone* total function — that is, $g(n)$ and $g(n+1)$ are defined and $g(n) \leq g(n+1)$ for all $n \in \mathbb{N}$.

Solution: Consider the following $\mathcal{L}$ program — which uses a macro to evaluate the function f that was considered in part (a).

$$\begin{array}{ll}
 & Y \leftarrow 0 \\
 & Z_1 \leftarrow 0 \\
[A_1] & Z_2 \leftarrow f(Z_1) \\
 & Z_3 \leftarrow \Phi_{Z_2}(X_1) \\
 & Z_4 \leftarrow Y + Z_3 \\
 & Y \leftarrow Z_4 \\
 & Z_4 \leftarrow X_1 - Z_1 \\
 & \text{IF } Z_4 \neq 0 \text{ GOTO } B_1 \\
 & \text{GOTO } C_1 \\
[B_1] & Z_1 \leftarrow Z_1 + 1 \\
 & \text{GOTO } A_1
\end{array}$$

It is not difficult to prove the following by induction on n: For all $n \in \mathbb{N}$, if this program is executed with input n then the statement

$$\text{IF } Z_4 \neq 0 \text{ GOTO } B_1$$

is executed exactly n times. Furthermore, if $1 \leq i \leq n+1$ then, at the i^{th} execution of this statement, Z_1 has value $i-1$, Z_4 has value $n-i+1$, and Y has value

$$g(i-1) = \sum_{j=0}^{i-1} \Phi_{f(j)}(n).$$

When proving this one should note that, since f is a total function, every execution of the statement with label A_1 halts. Every execution of the statement *after* that one halts, as well, because $\Phi_{f(i)}$ is also a total function for all $i \in \mathbb{N}$.

Note that on the $n+1^{\text{st}}$ execution of the statement Z_4 has value 0 so that the test fails, and the next statement is executed. Since there is no statement with label C_1 this causes the execution of the program to halt, with $g(n)$ returned as output.

Thus (since the program does halt on every input n) it follows that g is a total computable function.

To see that this is also monotone note that, for every integer $x \geq 0$,

$$\begin{aligned}
g(x+1) - g(x) &= \sum_{i=0}^{x+1} \phi_{f(i)}(x+1) - \sum_{i=0}^{x} \phi_{f(i)}(x) \\
&= \phi_{f(x+1)}(x+1) + \sum_{i=0}^{x} (\phi_{f(i)}(x+1) - \phi_{f(i)}(x)) \qquad \text{(reordering terms)} \\
&\geq \sum_{i=0}^{x} (\phi_{f(i)}(x+1) - \phi_{f(i)}(x)) \qquad \text{(since } \phi_{f(x+1)}(x+1) \geq 0\text{)} \\
&\geq \sum_{i=0}^{x} 0 \qquad \text{(since } \phi_{f(i)} \text{ is a monotone function for all } i \in \mathbb{N}\text{)} \\
&\geq 0
\end{aligned}$$

— implying (by definition) that g is a monotone function, as required.

(c) Finally, you were asked to give a proof using ***diagonalization*** to show that Monotone is *not* a recursive set after all.

Solution: Consider the function $f_C : \mathbb{N} \to \mathbb{N}$ such that

$$f_C(x) = g(x) + 1$$

for all $x \in \mathbb{N}$. f_C is certainly a computable function, since g is, and f_C is also monotone, since

$$f_C(x_1) - f_C(x) = g(x+1) - g(x) \geq 0$$

for all $x \in \mathbb{N}$.

It follows that $f_C = \phi_{f(n)}$ for some number $n \in \mathbb{N}$.

However, it now follows that

$$
\begin{aligned}
f_C(n) &= g(n) + 1 && \text{(by the definition of } f_C\text{)} \\
&= 1 + \sum_{i=0}^{n} \phi_{f(i)}(n) && \text{(by the definition of } g\text{)} \\
&\geq \phi_{f(n)}(n) + 1 && \text{(since } \phi_{f(i)}(n) \geq 0 \text{ for } 0 \leq i \leq n-1\text{)} \\
&= f_C(n) + 1 && \text{(since } f_C = \phi_{f(n)}\text{).}
\end{aligned}
$$

This is certainly impossible, so a ***contradiction*** has been obtained. Since the only assumption that was made was that Monotone is a recursive set, it follows that this set is *not* recursive.