CPSC 513 — Assignment #2
Solutions for Question #2

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

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

This question also concerns the set

$$\text{Total} = \{n \in \mathbb{N} \mid \Phi_n \text{ is total}\}$$

that was considered during the lecture on "Diagonalization and Reducibility."

(a) You were first asked to give an $\mathcal{L}$-program such that, for inputs $x_1, x_2 \in \mathbb{N}$,

- if $\Phi_{x_2}(0), \Phi_{x_2}(1), \ldots, \Phi_{x_2}(x_1)$ are all defined then this program computes and returns
$$\Phi_{x_2}(0) + \Phi_{x_2}(1) + \cdots + \Phi_{x_2}(x_1);$$
- this program fails to halt on inputs x_1 and x_2 otherwise.

Solution Consider the following program.

$$
\begin{array}{ll}
 & Y \leftarrow 0 \\
 & Z_1 \leftarrow 0 \\
[A_1] & Z_2 \leftarrow \Phi_{X_2}(Z_1) \\
 & Z_3 \leftarrow Z_2 + Y \\
 & Y \leftarrow Z_3 \\
 & Z_3 \leftarrow X_1 - Z_1 \\
 & \text{IF } Z_3 \neq 0 \text{ GOTO } B_1 \\
 & \text{GOTO } C_1 \\
[B_1] & Z_1 \leftarrow Z_1 + 1 \\
 & \text{GOTO } A_1
\end{array}
$$

The following is not hard to prove by induction on n: For all $n, x_2 \in \mathbb{N}$, if each of the values

$$\Phi_{x_2}(0), \Phi_{x_2}(1), \ldots, \Phi_{x_2}(n)$$

are defined, and this program is executed with inputs n and x_2, then the statement

IF $Z_3 \neq 0$ GOTO B_1

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

$$\Phi_{x_2}(0) + \Phi_{x_2}(1) + \cdots + \Phi_{x_2}(i-1),$$

Z_1 has value $i-1$, and Z_3 has value $n+1-i$. Consequently Y has the desired output value and Z_3 has value 0 at the $n+1^{\text{st}}$ execution of this statement. The test will fail and the next statement will be reached and executed. Since there is no statement with label C_1 this will cause the execution of the program to halt with the desired output being returned.

Suppose, instead, that one or more of the values

$$\Phi_{x_2}(0), \Phi_{x_2}(1), \ldots, \Phi_{x_2}(n)$$

is *not* defined. Let $m \in \mathbb{N}$ such that $0 \leq m \leq n$ and m is the *smallest* value such that $\Phi_{x_2}(m)$ is undefined.

Notice that if $m = 0$ then the program fails to halt when it is executed with inputs n and x_2 because the statement with label A_1 is reached and fails to halt the first time it is executed.

Suppose, instead, that $1 \leq m \leq n$. Then, if the program is executed with inputs n and x_2 then, once again, it can be argued that the statement with label A_1 is executed exactly $m+1$ times. Indeed, it can be proved by induction on i that if $1 \leq i \leq m$ then the variable Z_1 has value $i-1$ and the variable Z_3 has value $n-i+1 \geq n-m+1 \geq 1$ at the i^{th} execution of this statement.

In particular, Z_1 has value $m-1$ and Z_3 has a positive value at the m^{th} execution of this statement — so that the test will pass once again, and the statement with label B_1 will be executed after that — increasing the value of Z_1 after that. The statement with label A_1 will be reached and executed, for a final time, two steps after that — and the fail will fail to halt, because of this execution of this statement will never end.

Thus this program has the properties listed in the question.

Note: Since this program exists it has encoding p for some (fixed by unknown) natural number p.

(b) You were next asked to prove that Total $\preceq_m$ Monotone.

Consider the function $f : \mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$,

$$f(n) = s_1^1(n, p)$$

where s_1^1 is the function mentioned in the Parameter Theorem and the constant p is the encoding of the program given in part (a). Since s_1^1 is primitive recursive it should be clear that f is also a primitive recursive function — so that if is certainly a computable total function.

Suppose that $n \in$ Total. Then, for all $x \in \mathbb{N}$,

$$\begin{aligned}\Phi_{f(n)}(x) &= \Phi^{(1)}(x, f(n)) \\ &= \Phi^{(1)}(x, s_1^1(n, p)) \\ &= \Phi^{(2)}(x, n, p) \\ &= \Phi_n(0) + \Phi_n(1) + \cdots + \Phi_n(x) \in \mathbb{N},\end{aligned}$$

since $n \in$ Total, so that Φ_n is a program that computes a total function. Thus $f(n) \in$ Total. Furthermore, it follows by the above that for all $x \in \mathbb{N}$,

$$\Phi_{f(n)}(x+1) - \Phi_{f(n)}(x) = \Phi_{x_2}(x+1) \geq 0,$$

so that $\Phi_{f(n)}$ is a monotone function. Thus $f(n) \in$ Monotone in this case.

Suppose, on the other hand, that $n \neq$ Total. Then Φ_n is not a total function. Let x be the smallest natural number such that $\Phi_n(x)$ is undefined. Then (arguing as above once again)

$$\Phi_{f(n)}(x) = \Phi^{(2)}(x, n, p)$$

is undefined as well, because Φ_p fails to halt when executed on inputs x and n. Thus $\Phi_{f(n)}$ is not a total function, so that it is not monotone, either: $f(n) \notin$ Monotone.

Thus $x \in$ Total if and only if $f(x) \in$ Monotone and, since f is a computable total function it follows that

$$\text{Total} \preceq_m \text{Monotone}$$

as claimed.

(c) Finally you were asked what can be concluded about the set Monotone.

Solution: Recall (from Lecture #9) that the set Total is not recursively enumerable. It follows by the above reduction that Monotone is not recursively enumerable either.