HW 08

Elizabeth Hunt (A02364151)

November 11, 2023

1 Problem One

Let m be the number of statements that don't produce an effect on Y i.e. $\{Y \leftarrow Y, X1 \leftarrow X1+1, \cdots\}$, n be the number of statements $Y \leftarrow Y-1$, and o be the number of statements $Y \leftarrow Y+1$. A straightline program of length k must have $m+n+o=k$ instructions. Then, the terminal snapshot ends with $Y = o-n+0m$. Thus, Y is at a maximum if and only if $o=k$, thus $Y=k$, and at a minimum if and only if $o=n$, thus $Y=0$ (we cannot have $n>o$ as by definition of L, for a valid program P Y is a non-negative number).

2 Problem Two

Let P be a program in L that computes f. Then f is partially computable by definition, and is partially computable in $L++$ trivially since $L++$ is a "superset" of L. ($L++ \Leftarrow L$).
Then let Q be a program in $L++$ that computes f, f is partially computable if we can convert Q into L. For each instruction $i \in Q$, if i is in the form $V \leftarrow k | k \in N$ then we replace i with the sequence of instructions and unique labels (with the notation $\cdots$ representing a reapeating instruction until the instruction is the following index), else write i ($L++ \Rightarrow L$).

```
1. [A1] IF V != 0 GOTO B1
2.      GOTO C1
3. [B1] V <- V - 1
4.      GOTO A1
5. [C1] V <- V + 1
...     V <- V + 1
(5 + k - 1). V <- V + 1
```

And thus any program in $L++$ is also partially computable.

3 Problem Three

For $n=0$, $i(n,x) = f^0(x) = x$ is primitive recursive since it can be represented as a projection function $u_1^1(x) = x$. If f^n is primitive recursive, then so is f^{n+1}, as f^{n+1} is a finite composition $f(f^n)$. By induction, for any $n \in N$, $i(n,x)$ is primitive recursive, and thus computable.

4 Problem Four

4.1 One

Let $m \in N$ represent the number of compositions in a function such that the set containing all n -ary functions with "composition number" m are of the form

$f(x_1, \cdots, x_n) = g(f_1(x_1, \cdots, x_n), \cdots, f_j(x_1, \cdots, x_n))$ where each f_i is in the set of functions with a "composition number" $\leq m - 1$. Except for the case in which $m = 0$ - the initial functions; which are all of the form k or $x_i + k$.
Then, let $F^m | m \in N$ represent the set of all functions of composition number m or less, and assume that $\forall f \in F^m$ are of either form k or $x_i + k$. Then, $g \in F^{m+1} | m \in N$'s elements are $g(x_1, \cdots, x_n) = f(f_1(x_1, \cdots, x_n), \cdots, f_j(x_1, \cdots, x_n))$ where f is an initial function, by definition, and each $f_i | 1 \leq i \leq j \in F^m$.

1. If f is the Zero function g is in the form $k = 0$
2. If f is the Successor function g is some $f_i(\cdots)+1$ which by assumption is thus $(x_i+k)+1 \Rightarrow x_i + k$ or $k + 1 \Rightarrow k$.
3. If f is the Projection function g is some $f_i(\cdots)$ which by assumption is $(x_i + k)$ or k.

Finally, by induction, $\forall f(x_1, \cdots, x_n) \in \texttt{COMP}$, f is thus in the form k or $x_i + k | 1 \leq i \leq n$.

4.2 Two

By what we showed in One, $f(x_1, \cdots, x_n)$ is dependent on a single $x_i \in x_1, \cdots, x_n$ and produces $r = (x_i + k)$ or $r = k$. Then $f(y_1, \cdots, y_n) = (y_i + k) | k$ for the same i. So if for all $i | 1 \leq i \leq n$, $x_i \leq y_i$ then $(y_i + k) \geq (x_i + k)$ or $k = k$. Thus in both cases $f(y_1, \cdots, y_n) \geq f(x_1, \cdots, x_n)$ and is thus monotonic.

4.3 Three

Yes. Firstly, every $f \in \texttt{COMP}$ is primitive recursive, just without the recursion, so it is a subset of the set of all primitive recursive functions.
Consider the (proven in class) primitive recursive function $f(x_1, x_2) = x_1 * x_2$, and the equivalent $g \in \texttt{COMP}$. $g(x_1, x_2)$ is equivalent to $x_1 + k$, $x_2 + k$, or k. Then there all elements of $\{f(x_1, x_2 + 1), f(x_1 + 1, x_2), f(x_1, x_2)\} | x_1, x_2 > 1$ cannot be present in $\{g(x_1, x_2 + 1), g(x_1 + 1, x_2), g(x_1, x_2)\}$ for the same x_1, x_2 due to the dependence relation, which is a contradiction to f and g being equivalent.

4.4 Four

Yes. By the fact that all primitive recursive functions are computable, `COMP` being a set of primitive recursive functions from Three, is a subset of computable functions. However, we also showed in Three that there exists a computable (primitive recursive) function which is not in `COMP`, so `COMP` is not equivalent to the set of all computable functions.

5 Problem Five

1. Let $a(x_1, x_2) = x_1 + x_2$ and $m(x_1, x_2) = x_1 * x_2$ which are primitive recursive by proofs in class.
2. Let $t(x_1)$ be the composition of the successor function s 10 times on the zero function z function: $t(x_1) = s(s(\cdots(n(x_1))))$, and is primitive recursive.
3. Let $v(x_1)$ be $v(x_1) = a(x_1, a(x_1, a(x_1, a(x_1, u_1^1(x_1)))))$, which is primitive recursive.
4. Let $w(x_1)$ be $w(x_1) = a(t(x_1), v(x_1))$, which is primitive recursive.
5. Let $y(x_1)$ be $y(x_1) = m(m(x_1, x_1), s(s(n(x_1))))$, which is primitive recursive.
6. Then $f(x_1)$ is $a(w(x_1), y(x_1))$, which is primitive recursive.