Lecture #13: A Computable Total Function That is Not Primitive Recursive

Lecture Presentation

Expectations

This material is ***for interest only:*** Students will not be required to apply this material to solve problems on assignments, and students will not be tested on this.

Another Function To Consider

Definition: Let $f : \mathbb{N} \to \mathbb{N}$ be an unary function and let n be a positive integer. Then the ***n-fold application of f, $f^{\langle n \rangle}$***, is the function $f^{\langle n \rangle} : \mathbb{N} \to \mathbb{N}$ that is defined as follows:

(a) $f^{\langle 1 \rangle}(x) = f(x)$ for all $x \in \mathbb{N}$ — so that $f^{\langle 1 \rangle} = f$.

(b) For every positive integer n, and for $x \in \mathbb{N}$,

$$f^{\langle n+1 \rangle}(x) = f(f^{\langle n \rangle}(x)).$$

Note: If f is a total function then it is easily proved, by induction on n, that $f^{\langle n \rangle} : \mathbb{N} \to \mathbb{N}$ is a total function, for every positive integer n.

Definition: For all $n \in \mathbb{N}$, let $F_n : \mathbb{N} \to \mathbb{N}$ be defined as follows.

(a) $F_0(x) = x + 1$ for all $x \in \mathbb{N}$.

(b) $F_{n+1}(x) = F_n^{\langle x+1 \rangle}(x)$ for all $x, n \in \mathbb{N}$.

Exercise: Suppose that, for a fixed number $n \in \mathbb{N}$, the function $F_n : \mathbb{N} \to \mathbb{N}$ is primitive recursive. (Since F_0 is the "successor" function, this is true when $n = 0$.)

(a) Prove that the function $g : \mathbb{N}^2 \to \mathbb{N}$ such that, for all $x, y \in \mathbb{N}$,

$$g(x, y) = F_n^{\langle y \rangle}(x)$$

is primitive recursive.

(b) Use this to prove that the function $F_{n+1} : \mathbb{N} \to \mathbb{N}$ is primitive recursive, as well.

This is can be used to show that the function $F_n : \mathbb{N} \to \mathbb{N}$ is primitive recursive, for every natural number n.

Note that, since $F_0^{\langle 1 \rangle}(x) = F_0(x) = x + 1$, it is easily proved by induction, on y, that

$$F_0^{\langle y \rangle}(x) = x + y$$

for all $x, y \in \mathbb{N}$ such that $y \geq 1$. Thus

$$F_1(x) = F_0^{\langle x+1 \rangle}(x) = 2x + 1, \tag{1}$$

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

$$F_1^{\langle 1 \rangle}(x) = F_1(x) = 2x + 1,$$
$$F_1^{\langle 2 \rangle}(x) = F_1(F^{\langle 1 \rangle}(x)) = F_1(2x + 1) = 4x + 3,$$

and it an be proved by induction on y that

$$F_1^{\langle y \rangle}(x) = 2^y \cdot x + 2^y - 1$$

for all $x, y \in \mathbb{N}$ such that $y \geq 1$. Thus

$$F_2(x) = F_1^{\langle x+1 \rangle}(x) = 2^{x+1} \cdot x + 2^{x+1} - 1 > 2^x \tag{2}$$

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

One can continue this to see that the functions $F_3, F_4, F_5, \ldots$ keep growing faster and faster.

Lemma #1:

(a) $F_{i+1}(x) \geq F_i(x)$ for all $i, x \in \mathbb{N}$ — and, furthermore, $F_{i+1}(x) > F_i(x)$ if $x \geq 1$.

(b) $F_i(x) > x$ for all $i, x \in \mathbb{N}$.

(c) For all $i, x, y \in \mathbb{N}$, if $x > y$ then $F_i(x) > F_i(y)$.

How This is Proved:

Proof:

Definition: Let $g : \mathbb{N} \to \mathbb{N}$ be a unary total function and, for a positive integer k, let $h : \mathbb{N}^k \to \mathbb{N}$. We say that $\boldsymbol{g}$ ***eventually dominates*** $\boldsymbol{h}$ if there exists a number $N \in \mathbb{N}$ such that

$$g(\max(x_1, x_2, \ldots, x_k)) > h(x_1, x_2, \ldots, x_k)$$

for all $x_1, x_2, \ldots, x_k \in \mathbb{N}$ such that $\max(x_1, x_2, \ldots, x_k) \geq N$.

Equivalent Definition: for g, k and h as above, $\boldsymbol{g}$ ***eventually dominates*** $\boldsymbol{h}$ if and only if

$$g(\max(x_1, x_2, \ldots, x_k)) > h(x_1, x_2, \ldots, x_k)$$

for all but finitely many choices of the sequence of values $x_1, x_2, \ldots, x_k \in \mathbb{N}$.

Theorem #2: If $f : \mathbb{N}^k \to \mathbb{N}$ is a primitive recursive function then there exists a number $i \in \mathbb{N}$ such that F_i eventually dominates f.

How This is Proved:

Proof:

Finally, the Function: Let $A : \mathbb{N} \to \mathbb{N}$ such that

$$A(x) = F_x(x)$$

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

- This can be thought of as *another* simplified version of Ackerman's function.
- It is possible to show that A is a total computable function.
- It can be concluded, from Theorem #2, that the function A is not primitive recursive — because it is *not* eventually dominated by F_i for any fixed number $i \in \mathbb{N}$. Thus it is not primitive recursive because it "grows too fast", as claimed.