bounded minimization

One useful way of generating more primitive recursive functions from existing ones is through what is known as *bounded summation* and *bounded product*. Given a primitive recursive function $f:\mathbb{N}^{m+1}\to\mathbb{N}$, define two functions $f_s, f_p:\mathbb{N}^{m+1}\to\mathbb{N}$ as follows: for $\boldsymbol{x}\in\mathbb{N}^m$ and $y\in\mathbb{N}$:

$$f_s(\boldsymbol{x},y):=\sum_{i=0}^{y} f(\boldsymbol{x},i)$$

$$f_p(\boldsymbol{x},y):=\prod_{i=0}^{y} f(\boldsymbol{x},i)$$

These are easily seen to be primitive recursive, because they are defined by primitive recursion. For example,

$$f_s(\boldsymbol{x},0)=f(\boldsymbol{x},0),\quad \text{and}\quad f_s(\boldsymbol{x},n+1)=g(\boldsymbol{x},n,f_s(\boldsymbol{x},n)),$$

where $g(\boldsymbol{x},n,y)=\mathrm{add}(f(\boldsymbol{x},n),y)$, which is primitive recursive by functional composition.

Definition. We call f_s and f_p functions obtained from f by *bounded sum* and *bounded product* respectively.

Using bounded summation and bounded product, another useful class of primitive recursive functions can be generated:

Definition. Let $f:\mathbb{N}^{m+1}\to\mathbb{N}$ be a function. For each $y\in\mathbb{N}$, set

$$A_f(\boldsymbol{x}, y) := \{z \in \mathbb{N} \mid z \leq y \text{ and } f(\boldsymbol{x}, z) = 0\}.$$

Define

$$f_{bmin}(\boldsymbol{x}, y) := \begin{cases} \min A_f(\boldsymbol{x}, y) & \text{if } A_f(\boldsymbol{x}, y) \neq \emptyset, \\ s(y) & \text{otherwise.} \end{cases}$$

f_{bmin} is called the function obtained from f by *bounded minimization*, and is usually denoted

$$\mu z \leq y\,(f(\boldsymbol{x}, z) = 0).$$

Proposition 1. *If $f : \mathrm{N}^{m+1} \to \mathrm{N}$ is primitive recursive, so is f_{bmin}.*

Proof. Define $g := \operatorname{sgn} \circ f$. Then

$$g(\boldsymbol{x}, y) := \begin{cases} 0 & \text{if } f(\boldsymbol{x}, y) = 0, \\ 1 & \text{otherwise.} \end{cases}$$

As f is primitive recursive, so is g, since the sign function sgn is primitive recursive (see this entry (`http://planetmath.org/ExamplesOfPrimitiveRecursiveFunctions`)).

Next, the function g_p obtained from g by bounded product has the following properties:

- if $g_p(\boldsymbol{x}, y) = 1$, then $g_p(\boldsymbol{x}, z) = 1$ for all $z < y$,
- if $g_p(\boldsymbol{x}, y) = 0$, then $g_p(\boldsymbol{x}, z) = 0$ for all $z \geq y$.

Finally, the function $(g_p)_s$ obtained from g_p by bounded sum has the property that, when applied to $(\boldsymbol{x}, y)$, counts the number of $z \leq y$ such that $g_p(\boldsymbol{x}, z) = 1$. Based on the property of g_p, this count is then exactly the least $z \leq y$ such that $g_p(\boldsymbol{x}, z) = 1$. This means that $(g_p)_s = f_{bmin}$ for all $(\boldsymbol{x}, y) \in \mathbb{N}^{m+1}$. Since g_p is primitive recursive, so is $(g_p)_s$, or that f_{bmin} is primitive recursive. ∎