primitive recursive functions without primitive recursion

Canonical name	PrimitiveRecursiveFunctionsWithoutPrimitiveRecursion
Date of creation	2013-03-22 19:08:09
Last modified on	2013-03-22 19:08:09
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	7
Author	CWoo (3771)
Entry type	Example
Classification	msc 03D20

What sorts of functions can be built from the simplest primitive recursive functions (the initial functions) using functional composition alone? In this entry, we list some useful examples:

To begin with, we list the initial functions:

1. (zero function) $z(x) = 0$,
2. (successor function) $s(x) = x + 1$,
3. (projection functions) $p_i^n(x_1, \dots, x_n) = x_i$ for $i = 1, \dots, n$; in particular, we have the identity function $\mathrm{id}(x) = x$, which is just p_1^1.

With the help of functional composition, more functions can be derived:

1. (addition by a fixed number n) $s_n(x) = x + n$, which can be obtained by repeated application of the successor function s:

$$s_n := \underbrace{s \circ s \circ \cdots \circ s}_{n \text{ times}},$$

2. (constant functions) $\mathrm{const}_1(x) = 1$, which is just $s(z(x))$; more generally, $\mathrm{const}_n(x) = n$ is $s_n(z(x))$, where s_n is defined previously.

Next, we list some properties derivable using functional composition which are preserved by primitive recursiveness.

1. (permutation of variables) if $f(x_1, \dots, x_n)$ is primitive recursive, then so is any function g obtained from f by permuting the variables x_i:

$$g(x_1, \dots, x_n) = f(x_{\sigma(1)}, \dots, x_{\sigma(n)}),$$

where σ is a permutation on $\{1, \dots, n\}$;

2. (removing a variable) if $f(x_1, \dots, x_n, x_{n+1})$ is primitive recursive, then so is g defined by

$$g(x_1, \dots, x_n) := f(x_1, \dots, x_n, x_n);$$

3. (adding a variable) if $f(x_1, \dots, x_n)$ is primitive recursive, then so is g defined by

$$g(x_1, \dots, x_n, x_{n+1}) := f(x_1, \dots, x_n).$$

Proof. All of the above can be proved by appropriately substituting the suitable projection functions:

1. For each $i = 1, \dots, n$, let $h_i = p^n_{\sigma(i)}$. Then each h_i is primitive recursive, and therefore $g = f(h_1, \dots, h_n)$ is primitive recursive also.

2. For each $i = 1, \dots, n$, let $h_i = p^n_i$, and $h_{n+1} = p^n_n$. Then each h_i is primitive, and therefore $g = f(h_1, \dots, h_{n+1})$ is primitive recursive also.

3. For each $i = 1, \dots, n$, let $h_i = p^{n+1}_i$. Then each h_i is primitive recursive, and therefore $g = f(h_1, \dots, h_n)$ is primitive recursive also.

$\square$

As a corollary, we see that primitive recursiveness is preserved under generalized composition, in the following sense:

Corollary 1. *If $g_i : \mathbb{N}^{k_i} \to \mathbb{N}$, where $i = 1, \dots, n$, and $h : \mathbb{N}^n \to \mathbb{N}$ are primitive recursive, then the function f, given by*

$$f(x_1, \dots, x_m) = h(g_1(x_{t_1(1)}, \dots, x_{t_1(k_1)}), \dots, g_n(x_{t_n(1)}, \dots, x_{t_n(k_n)})),$$

where each t_i is a function on $\{1, \dots, k_i\}$, and $m \geq \max\{k_1, \dots, k_n\}$, is also primitive recursive.

Proof. Define $h_i(x_1, \dots, x_m) := g_i(x_{t_i(1)}, \dots, x_{t_i(k_i)})$. Then by repeated applications of the properties listed above, we see that h_i is primitive recursive. Hence $f = h(h_1, \dots, h_n)$ is also primitive recursive. $\square$