examples of primitive recursive functions

Canonical name	ExamplesOfPrimitiveRecursiveFunctions
Date of creation	2013-03-22 19:05:06
Last modified on	2013-03-22 19:05:06
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	17
Author	CWoo (3771)
Entry type	Example
Classification	msc 03D20
Related topic	ExamplesOfPrimitiveRecursivePredicates

Starting from the simplest primitive recursive functions, we can build more complicated primitive recursive functions by functional composition and primitive recursion. In this entry, we have listed some basic examples using functional composition alone. In this entry, we list more basic examples, allowing the use of primitive recursion:

1. $\operatorname{add}(x,y)=x+y$: $\operatorname{add}(x,0)=\operatorname{id}(x)$, and $\operatorname{add}(x,n+1)=s(\operatorname{add}(x,n))$
2. $\operatorname{mult}(x,y)=xy$: $\operatorname{mult}(x,0)=z(x)$, and $\operatorname{mult}(x,n+1)=\operatorname{add}(x,\operatorname{mult}(x,n))$.
3. $p_2(x)=x^2$, which is just $\operatorname{mult}(x,x)$; more generally, $p_{m+1}(x)=\operatorname{mult}(x,p_m(x))$, which is primitive recursive by induction on m
4. $\exp_m(x)=m^x$: $\exp_m(0)=s(0)$, and $\exp_m(n+1)=\operatorname{mult}(\operatorname{const}_m(n),\exp_m(n))$
5. $\exp(x,y)=x^y$: $\exp(x,0)=\operatorname{const}_1(x)$, and $\exp(x,n+1)=\operatorname{mult}(x,\exp(x,n))$
6. $\operatorname{fact}(x)=x!$: $\operatorname{fact}(0)=s(0)$, and $\operatorname{fact}(n+1)=\operatorname{mult}(s(n),\operatorname{fact}(n))$
7. $$\operatorname{sub}_1(x)=x\dot{-}1:=\begin{cases}0 & \text{if } x=0,\\ x-1 & \text{otherwise,}\end{cases}$$
   built using z and s: $\operatorname{sub}_1(0)=z(0)$, and $\operatorname{sub}_1(n+1)=s(\operatorname{sub}_1(n))$;
8. more generally, $\operatorname{sub}_m(x)=x\dot{-}m$ may be defined: $\operatorname{sub}_m=\operatorname{sub}_1^m$.
9. $\operatorname{sub}(x,y)=x\dot{-}y$: $\operatorname{sub}(x,0)=\operatorname{id}(x)$, and $\operatorname{sub}(x,n+1)=\operatorname{sub}_1(\operatorname{sub}(x,n))$.
10. $\operatorname{diff}(x,y)=|x-y|:=\operatorname{sub}(x,y)+\operatorname{sub}(y,x)$
11. $$d_0(x):=\begin{cases}1 & \text{if } x=0,\\ 0 & \text{otherwise.}\end{cases}$$
    built using const_1 and z: $d_0(0)=\operatorname{const}_1(0)$, and $d_0(n+1)=z(d_0(n))$.
12. more generally,
    $$d_m(x):=\begin{cases}1 & \text{if } x=m,\\ 0 & \text{otherwise.}\end{cases}$$
    is primitive recursive, for it is $d_0(\operatorname{diff}(x,\operatorname{const}_m(x)))$.

13. even more generally,

$$d_S(x) := \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{otherwise.} \end{cases}$$

where $S = \{a_1, \ldots, a_m\}$, is primitive recursive, for it is $d_{a_1} + \cdots + d_{a_m}$.

14.
$$\mathrm{sgn}(x) := \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{otherwise.} \end{cases}$$

which is just $\mathrm{sub}(\mathrm{const}_1(x), d_0(x))$.

15.
$$\mathrm{rem}(x, y) := \begin{cases} 0 & \text{if } y = 0, \\ x \bmod y & \text{otherwise,} \end{cases}$$

where $x \bmod y$ is the remainder of $x \div y$. Suppose $y \neq 0$. Then $0 \bmod y = 0$. In addition,

$$(n+1) \bmod y = \begin{cases} 0 & \text{if } \mathrm{diff}(s(n \bmod y), y) = 0, \\ s(n \bmod y) & \text{otherwise,} \end{cases}$$

Then $\mathrm{rem}(0, y) = z(y)$, and

$$\begin{aligned} \mathrm{rem}(n+1, y) &= \mathrm{sgn}(y)(\mathrm{rem}(n, y) + 1)\,\mathrm{sgn}(|\,\mathrm{rem}(n, y) + 1 - y|) \\ &= \mathrm{mult}(\mathrm{sgn}(y), \mathrm{mult}(s(\mathrm{rem}(n, y)), \mathrm{sgn}(\mathrm{diff}(s(\mathrm{rem}(n, y)), y)))) \\ &= g(y, \mathrm{rem}(n, y)) \end{aligned}$$

where $g(y, x) := \mathrm{mult}(\mathrm{sgn}(y), \mathrm{mult}(s(x), \mathrm{sgn}(\mathrm{diff}(s(x), y))))$. The reason for including $\mathrm{sgn}(y)$ is to account for the case when $y = 0$.

16.
$$q(x, y) = \begin{cases} \text{quotient of } x \div y & \text{if } y \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

To see that q is primitive recursive, we use equation

$$x = yq(x, y) + \mathrm{rem}(x, y)$$

obtained from the division algorithm for integers. Then

$$yq(x, y) + \mathrm{rem}(x, y) + 1 = x + 1 = yq(x + 1, y) + \mathrm{rem}(x + 1, y).$$

Simplify and we get

$$y(q(x+1,y) - q(x,y)) = \mathrm{rem}(x,y) + 1 - \mathrm{rem}(x+1,y).$$

Thus, by the previous example, we get

$$q(n+1,y) = \begin{cases} q(n,y)+1 & \text{if } \mathrm{rem}(n,y)+1 = y, \\ q(n,y) & \text{otherwise.} \end{cases}$$

Therefore, $q(0,y) = z(y)$, and

$$q(n+1,y) = \mathrm{sgn}(y)(q(n,y) + \mathrm{sgn}(\mathrm{diff}(s(\mathrm{rem}(n,y)),y)))$$

where $\mathrm{sgn}(y)$ takes the case $y = 0$ into account.

Remarks.

- All of the functions above are in fact examples of elementary recursive functions.
- Example 3(m) above is a special case of a more general phenomenon. Recall that a subset $S \subseteq \mathbb{N}^n$ is called *primitive recursive* if its characteristic function φ_S is primitive recursive. If we take $S = \{m\}$, then $\varphi_S = d_m$. Furthermore, a predicate $\Phi(\boldsymbol{x})$ over $\mathbb{N}^k$ is *primitive recursive* if the corresponding set $S(\Phi) := \{\boldsymbol{x} \in \mathbb{N}^k \mid \Phi(\boldsymbol{x})\}$ is primitive recursive.
- The technique of bounded maximization may be used to prove the primitive recursiveness of the quotient and the reminder functions in examples 3(o) and 3(p). See `http://planetmath.org/BoundedMaximization`this entry for more detail.