rec.1 Examples of Primitive Recursive Functions

cmp:rec:exa:sec

Here are some examples of primitive recursive functions:

1. Constants: for each natural number n, the function that always returns n primitive recursive function, since it is equal to $\mathrm{succ}(\mathrm{succ}(\dots\mathrm{succ}(\mathrm{zero}(x))))$.
2. The identity function: $\mathrm{id}(x) = x$, i.e. P^1_0
3. Addition, $x + y$
4. Multiplication, $x \cdot y$
5. Exponentiation, x^y (with 0^0 defined to be 1)
6. Factorial, $x! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot x$
7. The predecessor function, $\mathrm{pred}(x)$, defined by
$$\mathrm{pred}(0) = 0, \quad \mathrm{pred}(x + 1) = x$$
8. Truncated subtraction, $x \dot{-} y$, defined by
$$x \dot{-} 0 = x, \quad x \dot{-} (y + 1) = \mathrm{pred}(x \dot{-} y)$$
9. Maximum, $\max(x, y)$, defined by
$$\max(x, y) = x + (y \dot{-} x)$$
10. Minimum, $\min(x, y)$
11. Distance between x and y, $|x - y|$

In our definitions, we'll often use constants n. This is ok because the constant function $\mathrm{const}_n(x)$ is primitive recursive (defined from zero and succ). So if, e.g., we want to define the function $f(x) = 2 \cdot x$ can obtain it by composition from $\mathrm{const}_n(x)$ and multiplication as $f(x) = \mathrm{const}_2(x) \cdot P^1_0(x)$. We'll make use of this trick from now on. explanation

You'll also have noticed that the definition of pred does not, strictly speaking, fit into the pattern of definition by primitive recursion, since that pattern requires an extra argument. It is also odd in that it does not actually $\mathrm{pred}(x)$ in the definition of $\mathrm{pred}(x + 1)$. But we can define $\mathrm{pred}'(x, y)$ by

$$\mathrm{pred}'(0, y) = \mathrm{zero}(y) = 0$$
$$\mathrm{pred}'(x + 1, y) = P^3_0(x, \mathrm{pred}'(x, y), y) = x$$

and then define pred from it by composition, e.g., as $\mathrm{pred}(x) = \mathrm{pred}'(P^1_0(x), \mathrm{zero}(x))$.

Problem rec.1. Show that

$$f(x,y) = \left. 2^{2^{\cdot^{\cdot^{\cdot^{2^x}}}}} \right\} y \text{ 2's}$$

is primitive recursive.

Problem rec.2. Show that $d(x, y) = \lfloor x/y \rfloor$ (i.e., division, where you disregard everything after the decimal point) is primitive recursive. When $y = 0$, we stipulate $d(x, y) = 0$. Give an explicit definifion of d using primitive recursion and composition. You will have detour through an axiliary function—you cannot use recursion on the arguments x or y themselves.

The set of primitive recursive functions is further closed under the following two operations:

1. Finite sums: if $f(x, \vec{z})$ is primitive recursive, then so is the function

$$g(y, \vec{z}) = \sum_{x=0}^{y} f(x, \vec{z}).$$

2. Finite products: if $f(x, \vec{z})$ is primitive recursive, then so is the function

$$h(y, \vec{z}) = \prod_{x=0}^{y} f(x, \vec{z}).$$

For example, finite sums are defined recursively by the equations

$$g(0, \vec{z}) = f(0, \vec{z}), \quad g(y + 1, \vec{z}) = g(y, \vec{z}) + f(y + 1, \vec{z}).$$

We can also define boolean operations, where 1 stands for true, and 0 for false:

1. Negation, $\mathrm{not}(x) = 1 \dot{-} x$
2. Conjunction, $\mathrm{and}(x, y) = x \cdot y$

Other classical boolean operations like $\mathrm{or}(x, y)$ and $\mathrm{ifthen}(x, y)$ can be defined from these in the usual way.

Photo Credits

Bibliography