



planetmath.org

Math for the people, by the people.

more examples of primitive recursive functions

Canonical name	MoreExamplesOfPrimitiveRecursiveFunctions
Date of creation	2013-03-22 19:06:02
Last modified on	2013-03-22 19:06:02
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	13
Author	CWoo (3771)
Entry type	Example
Classification	msc 03D20

With bounded minimization, bounded maximization, and properties of primitive recursive predicates, many more examples of primitive recursive functions can now be exhibited. We list some of the most common ones:

1. $\text{NxtPrm}(n)$ is the smallest prime number greater than or equal to n .

Let $\Phi_1(y)$ be the predicate “ y is prime” and $\Phi_2(n, y)$ the predicate “ $n \leq y$ ”. Then

$$\text{NxtPrm}(n) = \mu y (\Phi_1(y) \wedge \Phi_2(n, y)),$$

where μ is the minimization operator. Since both Φ_1 and Φ_2 are primitive recursive, NxtPrm is recursive. From <http://planetmath.org/ProofThatThereAreInfinitelyManyPrimes>, proof of the infinitude of primes, two conclusions can be made:

- NxtPrm is a total function, and
- the search for y need not be unbounded: let p_1, \dots, p_k be all the prime numbers less than n (suppose $n > 2$), then $p = p_1 \cdots p_k + 1$ does not divide any of the p_i , so must be at least n . Let q be some prime such that $q|p$. Then $q \neq p_i$ for any $i = 1, \dots, k$, which means that $n \leq q$. So what we have shown is that q is the most we need to search to find the next prime after n . Since $q \leq p = p_1 \cdots p_k + 1 \leq n! + 1$, we may reformulate the above expression:

$$\text{NxtPrm}(n) = \mu y \leq (n! + 1) (\Phi_1(y) \wedge \Phi_2(n, y)).$$

Since $n! + 1$ is primitive recursive, we conclude that NxtPrm is primitive recursive by one application of bounded minimization and functional composition.

2. $\text{Pr}(n)$ is the n -th prime number, and $\text{Pr}(0) := 1$. Again, this is a total function. To see that Pr is primitive recursive, we simply note that $\text{Pr}(n + 1) = \text{NxtPrm}(\text{Pr}(n))$.
3. $\text{div}(x, y)$, which is 1 if $x|y$, and 0 otherwise, is primitive recursive. The predicate $\Phi(x, y, z)$ “ $x = yz$ ” is primitive recursive, so that $f(x, y) := \mu z \leq x \Phi(x, y, z)$ is primitive recursive. $f(x, y)$ returns $z \leq x$ if $yz = x$, and $x + 1$ otherwise. Thus “ $f(x, y) \leq x$ ” is a primitive recursive

predicate, and its characteristic function is easily seen to be $\text{div}(x, y)$, hence primitive recursive.

Note that the least z such that $x = yz$ is also the *only* z satisfying the equation. An alternative, more direct (without bounded minimization) way to prove that div is primitive recursive is by noticing that $\text{div}(x, y) = 1 - \text{sgn}(\text{rem}(x, y))$, where sgn is the sign function, and rem is the remainder function, both of which are primitive recursive. Since $-$ is primitive recursive, so is div .

Remark. $\text{div}(x, y)$ is also the characteristic function of the predicate “ $x|y$ ”, which shows that “ x divides y ” is a primitive recursive predicate.

4. $[x^{1/y}]$, which returns the largest z (bounded by x) such that $z^y \leq x$, is primitive recursive, since it can be obtained by an application of bounded maximization to the predicate “ $z^y \leq x$ ”.
5. $\text{lo}(x, y)$, which returns the largest z (bounded by y) such that $x^z | y$, the integer version of the logarithm function, is primitive recursive, for $\text{lo}(x, y)$ is the largest z such that $\text{div}(x^z, y) = 1$, and div has been shown to be primitive recursive in the previous example. To make lo a total function, we also set $\text{lo}(x, y) := 0$ if $x \in \{0, 1\}$.
6. $F(n)$ is the n -th Fibonacci number. The Fibonacci numbers are defined using a slight variation of primitive recursion: $F(0) = F(1) = 1$, and $F(n + 2) = F(n + 1) + F(n)$. To show that F is primitive recursive, first define a function g as follows:

$$g(n) = \exp(2, F(n)) \exp(3, F(n + 1)).$$

Then $F(n) = \text{lo}(2, g(n))$, and $F(n + 1) = \text{lo}(3, g(n))$. Moreover, $g(0) = 6$, and

$$\begin{aligned} g(n + 1) &= \exp(2, F(n + 1)) \exp(3, F(n + 2)) \\ &= \exp(2, F(n + 1)) \exp(3, F(n + 1) + F(n)) \\ &= \exp(2, \text{lo}(3, g(n))) \exp(3, \text{lo}(3, g(n)) + \text{lo}(2, g(n))). \end{aligned}$$

Thus g is defined via primitive recursion from multiplication, exponentiation, and lo , all of which primitive recursive. Therefore, $g(n)$, and consequently, $F(n) = \text{lo}(2, g(n))$, is primitive recursive. The type of recursion used in defining the Fibonacci numbers is known as course-of-values recursion.

7. $\gcd(x, y)$ is the greatest common divisor of x and y (for convenience, we set $\gcd(x, 0) = \gcd(0, y) := 0$). In other words, the \gcd function is defined by two cases:

$$\gcd(x, y) := \begin{cases} 0 & \text{if } xy = 0, \\ z & \text{if } xy \neq 0 \text{ and } z \text{ is the largest number } \leq x \text{ with } z|x \text{ and } z|y. \end{cases}$$

Since both $z|x$ and $z|y$ are primitive recursive (from previous example), as well as the predicate “ $xy \neq 0$ ”, so is their conjunction. Let $c(x, y, z)$ be the corresponding characteristic function. By taking the bounded maximum, the second case of \gcd is primitive recursive. Since the first case of \gcd is clearly primitive recursive, hence \gcd itself is primitive recursive.

8. As a result, the predicate “ x and y are coprime” is primitive recursive, as it has the same characteristic function as the predicate “ $\gcd(x, y) = 1$ ”, and $\gcd(x, y)$ has just been shown to be primitive recursive.
9. Euler <http://planetmath.org/EulerPhi> function ϕ -function is primitive recursive. Let $c(x, y)$ be the characteristic function of the primitive recursive predicate “ x and y are coprime”. Consider the bounded sum

$$f(x, y) = \sum_{i=0}^y c(x, i),$$

which is primitive recursive. Hence $\phi(x) = f(x, x)$ is too. Note that $\phi(0) = 0$.

Remark. It is not hard to show that, all of the functions above are in fact elementary recursive.