The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 5, Problem 5.5

- Recursion is the key to this definition. We can easily write the base case as

$$f(0, z_1, \ldots, z_n) = \text{is\_zero}(P(0, z_1, \ldots, z_n))$$

  and can then use the step to write

$$f(n+1, z_1, \ldots, z_n) = \begin{cases} f(n, z_1, \ldots, z_n) & \text{if } f(n, z_1, \ldots, z_n) \leq n \\ n+1 & \text{if } P(n+1, z_1, \ldots, z_n) \\ n+2 & \text{otherwise} \end{cases}$$

  The reader will recall that the ordering of the cases is significant. Here we use recursion to check the smaller values of the first argument; if the returned value is within the range (that is, a solution was found), it is returned as the answer; otherwise the larger value $n+1$ is tried; and if that also fails, then the first value outside the range, namely $n+2$, is returned.
- A simple affair if we remember to think in terms of strings as well as numbers:

$$(x \leq y) = \text{is\_zero}(\text{minus}(x, y))$$

- The key is to use multiplication, which we know to be primitive recursive. Note that x divides y evenly if and only if there exists a z such that $x \cdot z = y$. We need to define the equality predicate first, since we do not have one yet; to avoid confusion, let us denote it by equal(x, y); we have equal$(x, y) = \text{is_zero}(\text{con}_2(\text{minus}(x, y), \text{minus}(y, x)))$. Now we can just write

$$x|y = \exists z/y[\text{mult}(x, z) = y]$$

- Now we use the quantifiers and the division predicate just defined:

$$\text{is\_prime}(x) = \forall y \le x[2 \le |y| < |x| \to not(y|x)]$$

- We take the 0th prime to be 2 (thus $\text{prime}(0) = 2$, $\text{prime}(1) = 3$, $\text{prime}(2) = 5$, etc.) We need a number of helper functions. Basically, to construct the $n + 1$st prime, we start from the nth prime and search upwards, testing each successive number for primality. We know that the search need not go beyond some simple function of the nth prime; trivially, $\text{prime}(n + 1)$ is at most $\text{prime}(n)! + 1$, although in fact one can prove that $\text{prime}(n + 1)$ is at most $\text{prime}^2(n)$. We have not seen a primitive recursive definition for the strict inequality, but it is easy to define: $(x < y) = (x \le y) \wedge not(eq(x, y))$. We shall also need the function to establish the search range; if we use factorial plus one, we can write $fp1(x) = \text{Succ}(fact(x))$, with $fact(0) = 1$ and $fact(n + 1) = mult(n + 1, fact(n))$, where we omitted the use of the projection functions. Finally, we shall need a search for the smallest element in a range, i.e., a function

$$h(x, z_1, \dots, z_n) = \min y \le x\, P(y, z_1, \dots, z_n)$$

which returns the smallest y (prefix of x) such that the predicate holds, or returns $\text{Succ}(x)$ if no such y exists. This function is much easier to define than it would seem (see also the solution to Exercise 5.10): we set

$$h(0, z_1, \dots, z_n) = lev'(P(0, z_1, \dots, z_n))$$

and

$$h(x + 1, z_1, \dots, z_n) = \begin{cases} h(x, z_1, \dots, z_n) & \text{if } h(x, z_1, \dots, z_n) \le x \\ \text{Succ}(x) & \text{if } P(\text{Succ}(x), z_1, \dots, z_n) \\ \text{Succ}(\text{Succ}(x)) & \text{otherwise} \end{cases}$$

Now we can write $\text{prime}(x)$ as

$$\begin{cases} \text{prime}(0) = 2 \\ \text{prime}(n + 1) = \min y \le fp1(\text{prime}(n))[(y > \text{prime}(n) \wedge \text{is\_prime}(y)] \end{cases}$$