Lab03-Recursive Function

CS363-Computability Theory, Xiaofeng Gao, Spring 2016

∗ Please upload your assignment to FTP or submit a paper version on the next class
∗ If there is any problem, please contact: nongeek.zv@gmail.com

∗ Name:Zhang Yu Peng   StudentId: 5130309468   Email: 845113336@qq.com

1. Show that the following functions are primitive recursive:

   (a) $half(n) = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even,} \\ \frac{n-1}{2}, & \text{if } n \text{ is odd.} \end{cases}$

   Proof:

$$half(0) = 0$$

$$half(x+1) = \begin{cases} \frac{x}{2}, & \text{if } x \text{ is even,} \\ \frac{x+1}{2}, & \text{if } x \text{ is odd.} \end{cases} = \begin{cases} half(x), & \text{if } x \text{ is even,} \\ half(x)+1, & \text{if } x \text{ is odd.} \end{cases}$$

   So, the function is primitive recursive.

   (b) $\max\{x_1, x_2, \cdots, x_n\}$ = the maximum of x_1, x_2, $\cdots$, x_n.
   Proof:
   For $n = 2$, $\max\{x_1, x_2\} = x + (y \dot{-} x)$, it's computable.
   Assume that for $n = k$, the function is computable, then for $n = k+1$, $\max\{x_1, ...x_k, x_{k+1}\} = \max\{\max\{x_1, ...x_k\}, x_{k+1}\}$.

   (c) $f(x)$ = the sum of all prime divisors of x.
   Proof:
   $f(x) = \sum_{y<x+1} yPr(y)div(y, x)$, so the function is computable.

   (d) $g(x) = x^x$.
   Proof:
   $g(x) = power(x, x)$, so the function is computable.

2. Show the computability of the following functions by minimalisation.

   (a) $f^{-1}(x)$, if $f(x)$ is a total injective computable function.
   Proof:
   $f^{-1}(x) = \mu y(f(y) - x = 0)$
   So, the function is computable.

   (b) $f(a) = \begin{cases} \text{the least non-negative integral root of } p(x) - a \ (a \in \mathbb{N}), \\ \text{undefined if there's no such root,} \end{cases}$

   where $p(x)$ is a polynomial with integer coefficients.
   Proof:
   $f(a) = \mu y(p(x) - a = 0)$
   So, the function is computable.

(c) $f(x,y) = \begin{cases} x/y & \text{if } y \neq 0 \text{ and } y|x, \\ \text{undefined} & \text{otherwise.} \end{cases}$

Proof: $f(x,y) = \mu z(mult(y,z) - x = 0$

So, the function is computable.

3. Let $\pi(x,y) = 2^x(2y+1) - 1$. Show that π is a computable bijection from $\mathbb{N}^2$ to $\mathbb{N}$, and that the functions π_1, π_2 such that $\pi(\pi_1(z), \pi_2(z)) = z$ for all z are computable.

Proof:

First, we proof the injective: suppose that $\pi(x_1,y_1) = 2^{x_1}(2y_1+1)-1, \pi(x_2,y_2) = 2^{x_2}(2y_2+1)-1$, when $\pi(x_1,y_1) = \pi(x_2,y_2)$, we have $2^{x_1}(2y_1+1)-1 = 2^{x_2}(2y_2+1)-1$, which is $\frac{2^{x_2}}{2^{x_1}} = \frac{2y_1+1}{2y_2+1}$. We can easily see that the right side of the equation is an odd number, so if the equation is right, that means $2^{x_2-x_1}$ is an odd number, so $2^{x_2-x_1} = 1, x_1 = x_2$, then $y_1 = y_2$. So the function is injective.

Second, we proof the surjective: if z is an even number, then $\pi(0, z/2) = z$ can map to z. if z is an odd number, $z+1$ is an even number, let 2^x is the max number that $2^x|(z+1)$, so $\pi(x, (\frac{z+1}{2^x} - 1)/2) = z$, so the function is surjective.

So, the function is bijective.

Third, due to the results above, we can see $\pi_1(z) = (\mu y < (z+1)(div(power(2,y), z+1) = 0)\dot{-}1)$, $\pi_2(z) = (qt(2, qt(power(2, \pi_1(z))), z+1)\dot{-}1)$, so the function is computable.

4. Show that the following function is primitive recursive (with the help of $\pi(x,y)$, perhaps):

$$\begin{aligned} f(0) &= 1, \\ f(1) &= 1, \\ f(n+2) &= f(n) + f(n+1). \end{aligned}$$

Proof:

Define $g(n) = power(2, f(n))power(3, f(n+1))$ and $\pi_3(z) = (\mu y < (z+1)(div(power(3,y), z+1) = 0)\dot{-}1)$, so $f(n) = \pi_1(g(n)), f(n+1) = \pi_3(g(n))$.

Then,

$$g(0) = 6$$

$$g(n+1) = power(2, f(n+1))power(3, f(n+2)) = power(2, f(n+1))power(3, f(n+1)+f(n)) = power(2, \pi_3(g(n)))power(3, \pi_3(g(n)) + \pi_2(g(n)))$$

So, the function is primitive recursion.

5. Coding Technology.

Any number $x \in \mathbb{N}$ has a unique expression as

(1) $x = \sum_{i=0}^{\infty} \alpha_i 2^i$, with $\alpha_i = 0$ or 1, for all i.

Hence, if $x > 0$, there are unique expressions for x in the forms

(2) $x = 2^{b_1} + 2^{b_2} + \ldots + 2^{b_l}$, with $0 \leq b_1 < b_2 < ... < b_l$ and $l \geq 1$, and

(3) $x = 2^{a_1} + 2^{a_1+a_2+1} + \ldots + 2^{a_1+a_2+\ldots+a_k+k-1}$. (The expression (3) is a way of regarding x as coding the sequence $(a_1, a_2, \ldots, a_l)$ of numbers)

Show that each of the functions α, l, b, a defined below is computable.

(a) $\alpha(i,x) = \alpha_i$ as in the expression (1);

Proof:

$\alpha(i,x) = rm(2, qt(power(2,i),x))$
So the function is computable.

(b) $l(x) = \begin{cases} l \text{ as in (2)}, & \text{if } x > 0, \\ 0 & \text{otherwise;} \end{cases}$
Proof:
$l(x) = \sum_{i<x+1}(\alpha(i,x))$
So, the function is computable.

(c) $b(i,x) = \begin{cases} b_i \text{ as in (2)}, & \text{if } x > 0 \text{ and } 1 \leq i \leq l, \\ 0 & \text{otherwise;} \end{cases}$
Proof: $b(i,x) = \mu z < (x+1)(\sum_{j<z+1} \alpha(j,x) - i = 0) sg(x) sg(i) \bar{sg}(i + 1 \dot{-} l(x))$
so, the function is computable.

(d) $a(i,x) = \begin{cases} a_i \text{ as in (3)}, & \text{if } x > 0 \text{ and } 1 \leq i \leq l, \\ 0 & \text{otherwise;} \end{cases}$
Proof:
We can see that $x = 2^{a_1}(1 + 2^{a_2+1}(...(1 + 2^{a_l+1})))$
Define $g(x) = (\mu y < (x)(div(power(2,y),x) = 0) \dot{-} 1)$, $h(x) = qt(power(2, g(x)), x)$
Define

$$F(x,0) = 2x + 1$$

$$F(x, i+1) = h(F(x,i) \dot{-} 1)$$

So, $F(x, i \dot{-} 1) = 2^{a_1}(1 + 2^{a_2+1}(...(1 + 2^{a_l+1})))$,then:

$$a(i,x) = g(qt(2, F(x, i \dot{-} 1) \dot{-} 1)) sg(x) sg(i) \bar{sg}(i + 1 \dot{-} l(x))$$