Lógica y Computabilidad

FCEyN - UBA

Primer Cuatrimestre 2013

Práctica 8: Funciones Primitivas Recursivas

1. Sean ψ y ϕ funciones numéricas totales de una y dos variables respectivamente. Analizar cuáles de las siguientes definiciones de f son definiciones por recursión primitiva.

   a) $f(x,0) = 17.$
   $f(x, y+1) = f(0, \phi(x,y)).$

   b) $f(x,0) = \psi(x).$
   $f(x, y+1) = f(x,y) + \phi(y,x).$

   c) $f(x,0) = \phi(0,x).$
   $f(x, y+1) = \phi(f(x,y), y+1).$

   Para cada una de las definiciones que representen una recursión primitiva, especificar las funciones g y h asociadas a partir de las cuales se obtiene f por recursión primitiva.

2. Probar que cada una de las siguientes funciones es primitiva recursiva.

   a) $cte_k(x) = c$, donde k es un natural.

   b) $max(x,y) = \begin{cases} x & \text{si } x \geq y \\ y & \text{si } x < y \end{cases}$

   c) $min(x,y) = \begin{cases} x & \text{si } x \leq y \\ y & \text{si } x > y \end{cases}$

   d) $par(x,y) = \begin{cases} 1 & \text{si } x \textit{ es par} \\ 0 & \text{si } x \textit{ es impar} \end{cases}$

   e) $hf(x) = \lfloor \frac{x}{2} \rfloor.$

   f) $sqrt(x) = \lfloor \sqrt{x} \rfloor.$

g) $psq(x) = \begin{cases} 1 & \text{si } x \; es \; un \; cuadrado \; perfecto \\ 0 & \text{en otro caso} \end{cases}$

3. Sean ψ y ϕ funciones primitivas recursivas de una y dos variables, respectivamente. Mostrar que cada una de las funciones siguientes es también primitiva recursiva.

   a) La función f_1 de una variable, donde $f_1(0) = \psi(0) + 1$,
   $f_1(1) = \psi(\psi(1) + 1) + 1$, y en general:
   $f_1(x) = \psi(\psi(\ldots(\psi(x) + 1)\ldots) + 1) + 1.$
   donde la cantidad de veces que aparece ψ es $x + 1$.

   b) La función f_2 de dos variables, donde $f_2(x, 0) = \phi(x, 0)$, $f_2(x, 1) = \phi(\phi(x, 1), 0)$, y en general:
   $f_2(x, y) = \phi(\phi(\phi(\ldots\phi(\phi(x, y), y - 1)\ldots 2), 1), 0)$.

4. Usar las definiciones por sumas y/o productos acotados para establecer la recursividad primitiva de cada una de las funciones siguientes. Suponer que g es una función primitiva recursiva de una variable.

   a) $f(y) =$ el número de valores de i en el intervalo $0 \leq i \leq y$ para los cuales $g(i) > 3$.

   b) $f(x, y) = \begin{cases} 1 & \text{si } g(i+1) > g(i) \text{ para todo } x \leq i \leq y \\ 0 & \text{en otro caso} \end{cases}$

   c) $f(w, x, y) = \begin{cases} 1 & \text{si } x \leq y \text{ y } w \text{ es el mayor entre } g(x), g(x+1), \ldots, g(y) \\ 0 & \text{en otro caso} \end{cases}$

5. Sea g una función primitiva recursiva de $n + 1$ variables, s, t funciones primitivas recursivas de 1 variable. Probar que las siguientes funciones son primitivas recursivas.

   a) $f_1(x_1, \ldots, x_n, y) = max_{0 \leq i \leq y}(g(x_1, \ldots, x_n, i))$.

   b) $f_2(x_1, \ldots, x_n, y) = \begin{cases} max_{s(y) \leq i \leq t(y)}(g(x_1, \ldots, x_n, i)) & \text{si } s(y) \leq t(y) \\ 0 & \text{en otro caso} \end{cases}$.

6. Probar que las funciones dadas a continuación son primitivas recursivas. Pueden usarse como funciones auxiliares, las dadas en la clase teórica o las ya calculadas anteriormente.

   a) $shr(x, n) = \lfloor \frac{x}{2^n} \rfloor$.

   b) $lg(x) = \begin{cases} 0 & \text{si } x = 0 \\ \lfloor log_2(x) \rfloor + 1 & \text{en otro caso} \end{cases}$

c) $dig(x,n)$ = el n-ésimo dígito en la representación binaria de x , contando desde la derecha y comenzando con 0. Así, $dig(13,0) = 1, dig(13,1) = 0, dig(13,2) = 1, dig(13,3) = 1, dig(13,4) = 0$, etc.

d) $wgt(x)$ = el número de unos en la representación binaria de x.

7. Probar que las siguientes funciones son primitivas recursivas:

   a) $f(n)$ es el último dígito del desarrollo decimal de n.

   b) $f(n)$ es el primer dígito del desarrollo decimal de n.

8. Probar que las siguientes funciones son primitivas recursivas:

   a) $G(n,m)$ es la cantidad de números primos entre n y m.

   b) $G(n,m) = f^n(m)$, donde f es primitiva recursiva.

9. Determinar el número de Gödel del programa siguiente:

$$\begin{array}{l} [A] \quad X_1 \leftarrow X_1 + 1 \\ \text{if} \quad X_1 \neq 0 \quad \text{go to } [A] \end{array}$$

10. a) ¿Qué programa tiene número de Gödel 0?

    b) ¿Qué programa tiene número de Gödel 1?

    c) ¿Qué programa tiene número de Gödel 10?

    d) ¿Qué programa tiene número de Gödel 86399?

11. Dos programas son equivalentes si para iguales entradas devuelven iguales salidas, i.e. ambos representan la misma función computable. Dado un natural g, dar una forma de obtener infinitos valores distintos $g_1, g_2, g_3, \ldots$ tales que para todo i el programa cuyo número de Gödel es g_i sea equivalente al programa cuyo número de Gödel es g.

12. Probar que la función de Fibonacci es primitiva recursiva. Esta función está definida por

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

13. Probar que la siguiente función es primitiva recursiva.

$$\begin{aligned} H(0) &= 0 \\ H(n+1) &= 1 + \prod_{i=1}^{n} H(i) \end{aligned}$$

14. De las funciones definidas en los últimos cuatro ejercicios de la práctica anterior. ¿Cuáles son primitivas recursivas?