Práctica 2 - Funciones primitivas recursivas

Lógica y Computabilidad

Verano 2007

Ejercicio 1. Sean las funciones totales $\varphi : \mathbb{N}^2 \to \mathbb{N}$ y $\psi : \mathbb{N} \to \mathbb{N}$. Sabiendo que la suma es una función primitiva recursiva, analizar si las siguientes definiciones de f son definiciones por recursión primitiva a partir de φ y ψ.

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

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

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

d. $f(x, 0) = \varphi(0, x)$
   $f(x, y + 1) = \varphi(f(x, y), y + 1)$

Para cada una de las definiciones que representen una recursión primitiva, especificar las funciones asociadas al esquema de recursión primitiva a partir del cual se obtiene f.

Ejercicio 2. Probar que las siguientes funciones son primitivas recursivas, mostrando que pueden obtenerse a partir de las funciones iniciales o a través de composición o recursión primitiva:

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

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

c. $\mathrm{par}(x) = \begin{cases} 1 & \text{si } x \text{ es par} \\ 0 & \text{si } x \text{ es impar} \end{cases}$

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

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

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

Ejercicio 3. Sean $\varphi : \mathbb{N}^2 \to \mathbb{N}$ y $\psi : \mathbb{N} \to \mathbb{N}$ funciones primitivas recursivas. Mostrar que las siguientes funciones también son primitivas recursivas.

a. La función $f_1 : \mathbb{N} \to \mathbb{N}$ definida como:

$$\begin{array}{rcl} f_1(0) &=& 0 \\ f_1(1) &=& 1^1 \\ f_1(2) &=& 2^2 \\ f_1(3) &=& 3^{3^3} \\ &\vdots& \\ f_1(n) &=& \underbrace{n^{n^{n^{\cdot^{\cdot^{\cdot^{n}}}}}}}_{n \text{ veces}} \end{array}$$

b. La función $f_2 : \mathbb{N} \to \mathbb{N}$ definida como:

$$\begin{array}{rcl} f_2(0) &=& \psi(0) \\ f_2(1) &=& \psi(\psi(1)+1))+1 \\ &\vdots& \\ f_2(x) &=& \underbrace{\psi(\psi(\ldots(\psi}_{x+1 \text{ veces}}(x)+1)\ldots)+1)+1 \end{array}$$

c. La función $f_3 : \mathbb{N}^2 \to \mathbb{N}$ definida como:

$$\begin{array}{rcl} f_3(x,0) &=& \varphi(x,0) \\ f_3(x,1) &=& \varphi(\varphi(x,1),0) \\ &\vdots& \\ f_3(x,y) &=& \underbrace{\varphi(\varphi(\varphi(\ldots\varphi(\varphi}_{y+1 \text{ veces}}(x,y),y-1),\ldots 2),1),0) \end{array}$$

Ejercicio 4. Usar las definiciones por sumas y/o productos acotados para establecer la recursividad primitiva de cada una de las siguientes funciones. Suponer que $g : \mathbb{N} \to \mathbb{N}$ es una función primitiva recursiva.

a. $f(x,y) = \#\{i : 0 \leq i \leq x \wedge g(i) > y\}$

b. $f(x,y) = \begin{cases} 1 & \text{si } g(i+1) > g(i) \text{ para todos los valores de } i \text{ tal que } x \leq i \leq y \\ 0 & \text{en otro caso} \end{cases}$

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

Ejercicio 5. Probar que las funciones f_1 y f_2 del ejercicio 8 de la práctica de funciones computables son primitivas recursivas, siempre que g, s y t lo sean.

Ejercicio 6. Probar que las funciones dadas a continuación son primitivas recursivas. Pueden usarse como funciones auxiliares las dadas en la clases teóricas y prácticas o las ya calculadas anteriormente.

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

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

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

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

e. Pr(n, m) es la cantidad de números primos entre n y m.

Ejercicio 7. Mostrar que la función $f : \mathbb{N} \to \mathbb{N}$ tal que $f(n) = [1, \ldots, n]$ es primitiva recursiva.

Ejercicio 8. Mostrar que la función cant : $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ definida como

$$\text{cant}(y, [x_1, \ldots, x_n]) = \text{cantidad de apariciones de } y \text{ en la lista } [x_1, \ldots, x_n]$$

para todo $x_1, \ldots, x_n \in \mathbb{N}$, $x_n \neq 0$ es primitiva recursiva.

Ejercicio 9. Para $n, x_1, \ldots, x_n \in \mathbb{N}$, $x_i \neq 0$ para todo i se define Sort$([x_1, \ldots, x_n]) = [y_1, \ldots, y_n]$, donde la secuencia $[y_1, \ldots, y_n]$ es una permutación de $[x_1, \ldots, x_n]$ tal que $y_1 \leq y_2 \leq \cdots \leq y_n$. Mostrar que Sort es primitiva recursiva.

Ejercicio 10. Mostrar que la función de Fibonacci

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

es primitiva recursiva.

Ejercicio 11. Sea $f : \mathbb{N} \to \mathbb{N}^+$ una función primitiva recursiva. Mostrar que la función map definida como

$$\text{map}([x_1, \ldots, x_n]) = [f(x_1), \ldots, f(x_n)]$$

para todo $x_1, \ldots, x_n \in \mathbb{N}$, $x_n \neq 0$ es primitiva recursiva.