Lógica y computabilidad
Práctica 8: *Funciones recursivas primitivas*

En esta práctica se puede usar que una función determinada es primitiva recursiva si se probó o dejó como ejercicio en clase. Y por supuesto, en un ejercicio dado todo lo que ya se probó en ejercicios anteriores también puede usarse directamente.

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) $\begin{cases} f(0,x) &= 17 \\ f(y+1,x) &= f\big(0,\phi(x,y)\big) \end{cases}$

b) $\begin{cases} f(0,x) &= \psi(x) \\ f(k+1,x) &= f(k,x)+\phi(x,k) \end{cases}$

c) $\begin{cases} f(x,0) &= \phi(0,x) \\ f(x,y+1) &= \phi(f(x,y),y+1). \end{cases}$

Para cada una de las definiciones que representen una recursión primitiva, especificar las funciones $g(t,y,x_1,\ldots,x_n)$ y $h(x_1,\ldots,x_n)$ a partir de las cuales se obtiene $f(t,x_1,\ldots,x_n)$ por recursión primitiva.

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

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

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

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

d) $\text{half}(x) = \left\lfloor \frac{x}{2} \right\rfloor$.

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

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

3) Sean ψ y ϕ funciones recursivas primitivas 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

$$f_1(0) = \psi(0) + 1,$$

$$f_1(1) = \psi\big(\psi(1) + 1)\big) + 1,$$

y en general

$$f_1(t) = \psi\Bigg(\psi\Big(\ldots\psi\big(\psi(t)+1\big)+1\ldots\Big)+1\Bigg)+1,$$

donde la cantidad de veces que aparece ψ en la expresión es $t+1$. Por ejemplo,

$$f_1(3) = \psi\Bigg(\psi\Big(\psi\big(\psi(3)+1\big)+1\Big)+1\Bigg)+1.$$

b) La función f_2 de dos variables, donde

$$f_2(0, x) = \phi(0, x),$$

$$f_2(1, x) = \phi(0, \phi(1, x)),$$

y en general:

$$f_2(t, x) = \phi\Bigg(0, \phi\Bigg(1, \phi\Big(2, \ldots\phi\big(t-1, \phi(t, x)\big)\ldots\Big)\Bigg)\Bigg).$$

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) $f(y)$ devuelve la cantidad de valores de i en el intervalo $0 \leq i \leq y$ para los cuales $g(i) > 3$, es decir,

$$f(y) = \#\{i \in \mathbb{N}_0 \colon 0 \leq i \leq y, g(i) > 3\}.$$

b)

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

c)

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

5) Sea g una función recursiva primitiva de $n+1$ variables. Se define la función f a partir de g por:

$$f(t, x_1, \ldots, x_n) = \max_{0 \leq i \leq t} \{g(i, x_1, \ldots, x_n)\},$$

de modo que f puede pensarse obtenida a partir de g por una operación de «máximo acotado» respecto de la primera variable.

a) Probar que la función f es primitiva recursiva.

b) Generalizar el esquema de definición dado de modo que se permita que los límites superior e inferior de i sean funciones recursivas primitivas arbitrarias de los argumentos de f, es decir:

$$f(t, x_1, \ldots, x_n) = \max_{\alpha(t,x_1,\ldots,x_n) \leq i \leq \beta(t,x_1,\ldots,x_n)} \{g(i, x_1, \ldots, x_n)\}.$$

(Usar la convención que asigna a f el valor 0 si el límite inferior excede al límite superior, es decir, si el máximo se toma para $i \in \emptyset$). Mostrar que la función resultante f es también recursiva primitiva.

6) Usar minimización acotada para probar que las funciones de los incisos a),b),d) y e) del ejercicio 2) son primitivas recursivas.

7) Sea la función h de una variable definida tal que $h(n)$ es el $n+1$-ésimo dígito significativo en la representación decimal de $\sqrt{2}$. De modo que $h(0) = 1$, $h(1) = 4$, $h(2) = 1$, $h(3) = 4$, etc. Probar que h es primitiva recursiva. (Sugerencia: considerar la función axiliar g, donde $g(n)$ es el número natural representado por los primeros $n+1$ dígitos significativos en la representación de $\sqrt{2}$. Así, $g(0) = 1$, $g(1) = 14$, $g(2) = 141$, $g(3) = 1414$, etc. Usar minimización acotada para probar que g es primitiva recursiva.

8) Probar que las funciones dadas a continuación son primitivas recursivas. Pueden usarse como funciones auxiliares, las dadas en clase las ya estudiadas en esta guía. [1]

[1] Tener en cuenta que el nombre y/o el orden de las variables no necesariamente indica el rol que cada una tiene en la recursión primitiva. Por ejemplo, $f(x,y) = y^x$, $f(y,x) = x^y$, $f(x,t) = t^x$, $f(t,u) = u^t$, etc., todas representan la misma función. Y como vale $x^{y+1} = x^y \cdot x$, pero también $y^{x+1} = y^x \cdot y$, etc., la variable de la recursión es a veces y, otras x, t, u, etc.

a) $\mathrm{shr}(t,x) = \left\lfloor \frac{x}{2^t} \right\rfloor$

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

c) $\mathrm{dig}(x,n)$, definida como el n-ésimo dígito en la representación binaria de x , contando desde la derecha. Es decir, si

$$x = b_N b_{N-1} \dots b_2 b_1 b_{0|2},$$

entonces $\mathrm{dig}(x,n) = b_n$ si $n \leq N$ y $\mathrm{dig}(x,n) = 0$ si $n > N$.

Por ejemplo, $\mathrm{dig}(13,0) = 1, \mathrm{dig}(13,1) = 0, \mathrm{dig}(13,2) = 1, \mathrm{dig}(13,3) = 1, \mathrm{dig}(13,4) = 0$, etc.

d) $\mathrm{wgt}(x)$ definida como cantidad de unos que figuran en la representación binaria de x.

9) Probar que las siguientes funciones son recursivas primitivas:

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

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

10) Probar que las siguientes funciones son recursivas primitivas:

a) $Q(m,n)$ es la cantidad de números primos p tales que $m \leq p \leq n$.

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

11) Sea $F(n)$ la sucesión/función de Fibonacci, dada por

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

Probar que $F(n)$ es recursiva primitiva.

(Sugerencia: considerar la función $f(n) = \langle F(n+1), F(n) \rangle$.)

12) Sea $\mathscr{T}$ la clase de todas las funciones totales (de cualquier ariedad finita).
a) Probar que $\mathscr{T}$ es PRC.
b) Probar que las funciones recursivas primitivas son funciones totales.

13) Si $\mathscr{P}$ es el conjunto de las funciones estrictamente parciales (es decir, las que no son totales) e $\mathscr{I}$ es el conjunto de todas las funciones iniciales, probar que $\mathscr{P} \cup \mathscr{I}$ no es una clase PRC.

14) Sea $n > 0$ y $\mathscr{C}_n$ la clase de todas las funciones totales de **a lo sumo** n variables.

a) Mostrar que existen funciones recursivas primitivas que no pertenecen a $\mathscr{C}_n$.

b) Probar que $\mathscr{C}_n \cup \mathscr{I}$ no es una clase PRC.