Práctica 6 -Funciones primitivas recursivas y clases *PRC*-

Ejercicio 1. Mostrar que, dado un k fijo, la función constante $f(x) = k$ puede definirse usando las funciones iniciales y composición (*sin* usar recursión primitiva).

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

$$f_1(x,y) = x + y \qquad f_2(x,y) = x \cdot y \qquad f_3(x,y) = x^y \qquad f_4(x,y) = \underbrace{x^{x^{x^{\cdot^{\cdot^{\cdot^{x}}}}}}}_{y \text{ veces}}$$

$$g_1(x) = x \dot{-} 1 \qquad g_2(x,y) = x \dot{-} y \qquad g_3(x,y) = \lfloor x/y \rfloor \qquad g_4(x,y) = x \bmod y$$

Observación: $x \dot{-} y = 0$ cuando $y > x$; y se asume que $g_3(x,0) = 0$ y $g_4(x,0) = x$.

Ejercicio 3. Sea $\mathcal{C}_i$ la clase de funciones iniciales, es decir, aquella que contiene a:

$$n(x) = 0 \qquad s(x) = x + 1 \qquad u_i^n(x_1,\dots,x_n) = x_i \text{ para cada } n \in \mathbb{N} \text{ e } i \in \{1,\dots,n\}$$

y sea $\mathcal{C}_c$ la (mínima) clase que extiende a $\mathcal{C}_i$ y se encuentra cerrada por composición, i.e., si $f, g_1, \dots g_m$ están en $\mathcal{C}_c$, entonces $h(x_1,\dots,x_n) = f(g_1(x_1,\dots,x_n),\dots,g_m(x_1,\dots,x_n))$ también lo está.

a. Demostrar que para toda $f : \mathbb{N}^n \to \mathbb{N}$, f está en $\mathcal{C}_c$ sii existe $k \geq 0$ tal que, o bien sucede $f(x_1,\dots,x_n) = k$, o bien para algún i fijo, se tiene $f(x_1,\dots,x_n) = x_i + k$.

b. Mostrar que existe una función primitiva recursiva que no está en $\mathcal{C}_c$.

Ejercicio 4. Llamamos *predicado* a cualquier función $p : \mathbb{N}^n \to \{0,1\}$, escribimos $p(a_1,\dots,a_n)$ en lugar de $p(a_1,\dots,a_n) = 1$ y decimos, informalmente, en ese caso, que "$p(a_1,\dots,a_n)$ es verdadero". Mostrar que los predicados $\leq$, $\geq$, $=$, $\neq$, $<$ y $>$: $\mathbb{N}^2 \to \{0,1\}$ están en cualquier clase *PRC*.

Ejercicio 5. Sea $\mathcal{C}$ una clase *PRC*, sean $f_1,\dots,f_k, g : \mathbb{N}^n \to \mathbb{N}$ funciones en $\mathcal{C}$ y sean también $p_1,\dots,p_k : \mathbb{N}^n \to \{0,1\}$ predicados disjuntos en $\mathcal{C}$ (i.e., no sucede $p_i(a_1,\dots,a_n) = p_j(a_1,\dots,a_n) = 1$ con $i \neq j$ para ningún $(a_1,\dots,a_n) \in \mathbb{N}^n$). Mostrar que también está en $\mathcal{C}$ cualquier función h que cumpla:

$$h(x_1,\dots,x_n) = \begin{cases} f_1(x_1,\dots,x_n) & \text{si } p_1(x_1,\dots,x_n) \\ \quad\vdots & \\ f_k(x_1,\dots,x_n) & \text{si } p_k(x_1,\dots,x_n) \\ g(x_1,\dots,x_n) & \text{si no} \end{cases}$$

Ejercicio 6. Sea $\mathcal{C}$ una clase *PRC*, y sean $f : \mathbb{N}^n \to \mathbb{N}$ y $g_1, g_2 : \mathbb{N}^{n+2} \to \mathbb{N}$ funciones en $\mathcal{C}$. Mostrar que también está en $\mathcal{C}$ cualquier h que cumpla:

$$h(x_1,\dots,x_n,t) = \begin{cases} f(x_1,\dots,x_n) & \text{si } t = 0 \\ g_1(x_1,\dots,x_n,k,h(x_1,\dots,x_n,t-1)) & \text{si } t = 2 \cdot k + 1 \\ g_2(x_1,\dots,x_n,k,h(x_1,\dots,x_n,t-1)) & \text{si } t = 2 \cdot k + 2 \end{cases}$$

Ejercicio 7. Sea $\mathcal{C}$ una clase *PRC* y sea $p : \mathbb{N}^{n+1} \to \{0,1\}$ un predicado en $\mathcal{C}$. Mostrar que también están en $\mathcal{C}$ las siguientes funciones:

$$
\begin{aligned}
\text{cantidad}_p(x_1,\ldots,x_n,y,z) &= |\{t \mid y \le t \le z \wedge p(x_1,\ldots,x_n,t)\}| \\
\text{todos}_p(x_1,\ldots,x_n,y,z) &= \begin{cases} 1 & \text{si } (\forall t : y \le t \le z) p(x_1,\ldots,x_n,t) \\ 0 & \text{si no} \end{cases} \\
\text{alguno}_p(x_1,\ldots,x_n,y,z) &= \begin{cases} 1 & \text{si } (\exists t : y \le t \le z) p(x_1,\ldots,x_n,t) \\ 0 & \text{si no} \end{cases} \\
\text{minimo}_p(x_1,\ldots,x_n,y,z) &= \begin{cases} \text{mín}\{t \mid y \le t \le z \wedge p(x_1,\ldots,x_n,t)\} & \text{si } z \ge y \\ 0 & \text{si no} \end{cases} \\
\text{maximo}_p(x_1,\ldots,x_n,y,z) &= \begin{cases} \text{máx}\{t \mid y \le t \le z \wedge p(x_1,\ldots,x_n,t)\} & \text{si } z \ge y \\ 0 & \text{si no} \end{cases} \\
\text{unico}_p(x_1,\ldots,x_n,y,z) &= \begin{cases} u & \text{si } \{u\} = \{t \mid y \le t \le z \wedge p(x_1,\ldots,x_n,t)\} \\ z+1 & \text{si no} \end{cases}
\end{aligned}
$$

Ejercicio 8. Mostrar, usando las funciones del ejercicio anterior, que las siguientes funciones están en toda clase PRC:

$$
\begin{aligned}
\text{divide}(x,y) &= \begin{cases} 1 & \text{si } x \text{ divide a } y \\ 0 & \text{si no} \end{cases} \\
\text{primo}(x) &= \begin{cases} 1 & \text{si } x \text{ es un número primo} \\ 0 & \text{si no} \end{cases} \\
\text{raiz}(x,y) &= \lfloor \sqrt[x]{y} \rfloor \\
\text{nprimo}(n) &= k \text{ sii } k \text{ es primo y hay sólo } n \text{ primos positivos menores que } k
\end{aligned}
$$

Ejercicio 9. Considerar la codificación de pares de naturales dada por $\langle x,y\rangle = 2^x(2y+1) \dot{-} 1$.

1. Mostrar que esta codificación es biyectiva.
2. Mostrar que las funciones proyectoras $l, r : \mathbb{N} \to \mathbb{N}$ tales que $l(\langle x,y\rangle) = x$ y $r(\langle x,y\rangle) = y$ están en toda clase PRC.

Ejercicio 10. Mostrar que *fib*, la función de Fibonacci, está en toda clase PRC, donde:

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

Ejercicio 11. Demostrar que toda clase PRC se encuentra cerrada por *recursión mutua*. Es decir, dada $\mathcal{C}$, una clase PRC y dadas f_1, f_2, g_1 y g_2 funciones en $\mathcal{C}$, mostrar que también están en $\mathcal{C}$ las funciones h_1 y h_2 que cumplen:

$$
\begin{aligned}
h_1(x_1,\ldots,x_n,t) &= \begin{cases} f_1(x_1,\ldots,x_n) & \text{si t} = 0 \\ g_1(h_1(x_1,\ldots,x_n,t-1), h_2(x_1,\ldots,x_n,t-1), x_1,\ldots,x_n,t) & \text{si no} \end{cases} \\
h_2(x_1,\ldots,x_n,t) &= \begin{cases} f_2(x_1,\ldots,x_n) & \text{si t} = 0 \\ g_2(h_2(x_1,\ldots,x_n,t-1), h_1(x_1,\ldots,x_n,t-1), x_1,\ldots,x_n,t) & \text{si no} \end{cases}
\end{aligned}
$$

Ejercicio 12. Sea $\mathcal{C}_{i+p}$ la clase de funciones que extiende a la clase de funciones iniciales $\mathcal{C}_i$ con la función codificadora de pares $\langle \cdot,\cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ y las proyectoras $l, r : \mathbb{N} \to \mathbb{N}$ y sea $\mathcal{C}_{Ack}$ la (mínima) clase que incluye a $\mathcal{C}_{i+p}$ y se encuentra cerrada por composición y por *iteración de funciones unarias*, i.e., si $f : \mathbb{N} \to \mathbb{N}$ está en $\mathcal{C}_{Ack}$, entonces también está $h(n,x) = f^{(n)}(x)$ (recordar que $f^{(n)} = \underbrace{f \circ f \circ \cdots \circ f}_{n}$). Demostrar que $\mathcal{C}_{Ack}$ coincide con la clase de funciones primitivas recursivas.

Ejercicio 13. Considerar la codificación de secuencias finitas de números naturales dada por $[a_0, \ldots, a_{n-1}] = \Pi_{i=0}^{n-1} \text{nprimo}(i)^{a_i}$, donde nprimo es la función definida en el Ejercicio 8.

a. Mostrar que la codificación dada forma una biyección entre el conjunto de secuencias finitas que no terminan en cero y los números naturales mayores que cero.

b. Determinar qué valor codifica la secuencia vacía y mostrar que las siguientes funciones están en toda clase *PRC*:

- $|\cdot| : \mathbb{N} \to \mathbb{N}$ tal que $|[a_0, \ldots, a_{n-1}]| = n$ (longitud)
- $\pi.(\cdot) : \mathbb{N}^2 \to \mathbb{N}$ tal que $\pi_i([a_0, \ldots, a_{n-1}]) = \begin{cases} a_i & \text{si } i < n \\ 0 & \text{si no} \end{cases}$ (proyección)
- $[\cdot] : \mathbb{N} \to \mathbb{N}$ tal que $[x]$ es la lista con único elemento x (creación)
- $\cdot \circ \cdot : \mathbb{N}^2 \to \mathbb{N}$ tal que $[a_1, \ldots, a_n] \circ [b_1, \ldots, b_m] = [a_1, \ldots a_n, b_1, \ldots b_m]$ (concatenación)
- $\text{sub} : \mathbb{N}^3 \to \mathbb{N}$ tal que $\text{sub}([a_1, \ldots, a_n], i, j) = [a_i, \ldots, a_j]$ (sublista)

c. Proponer una codificación de secuencias ρ : Listas $\to \mathbb{N}$ que forme una biyección entre los números naturales (incluyendo el cero) y el conjunto de *todas* las secuencias finitas de naturales tal que las funciones del punto b estén en toda clase *PRC*.

Ejercicio 14. Demostrar que toda clase *PRC* se encuentra cerrada por *recursión global* (*course-of-values recursion*). Es decir, dada $\mathcal{C}$, una clase *PRC*, y dada una función $f : \mathbb{N}^{n+1} \to \mathbb{N}$ en $\mathcal{C}$, mostrar que la función definida como

$$\begin{aligned} h(x_1, \ldots, x_n, 0) &= f([], x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n, t+1) &= f([h(x_1, \ldots, x_n, 0), \ldots, h(x_1, \ldots, x_n, t)], x_1, \ldots, x_n) \end{aligned}$$

también está en $\mathcal{C}$.

Ejercicio 15. Mostrar que las siguientes funciones están en toda clase *PRC*:

- $\text{cant} : \mathbb{N}^2 \to \mathbb{N}$, tal que $\text{cant}(x, [a_0, \ldots, a_{n-1}]) = k$ sii hay exactamente k apariciones de x en $[a_0, \ldots, a_{n-1}]$.
- $\text{elem} : \mathbb{N}^2 \to \{0, 1\}$, tal que $\text{elem}(x, [a_0, \ldots, a_{n-1}])$ sii $a_i = x$, para algún $i < n$.
- $\text{perm} : \mathbb{N}^2 \to \{0, 1\}$ tal que $\text{perm}(x, y) = 1$ sii x representa una permutación de y.
- $\max : \mathbb{N} \to \mathbb{N}$ tal que $\max([a_0, \ldots, a_{n-1}])$ es el máximo de todos los a_i (o es 0 si $n = 0$).
- $\text{sort} : \mathbb{N} \to \mathbb{N}$ tal que $\text{sort}(x)$ es una permutación de x ordenada de menor a mayor.
- $\text{aplanar} : \mathbb{N} \to \mathbb{N}$ tal que

$$\text{aplanar}([[a_0^1, \ldots, a_{n_1-1}^1], \ldots, [a_0^k, \ldots, a_{n_k-1}^k]]) = [a_0^1, \ldots, a_{n_1-1}^1, \ldots, a_0^k, \ldots, a_{n_k-1}^k]$$

Más sobre funciones computables

Ejercicio 16. Utilizando las funciones primitivo-recursivas $\text{STP}^{(n)}$ y $\text{SNAP}^{(n)} : \mathbb{N}^{n+2} \to \mathbb{N}$ vistas en clase, mostrar que las siguientes son funciones parciales computables:

$$f_1(\#p, x) = \begin{cases} 1 & \text{si } x \in \text{Dom}\, \Psi_p^{(1)} \\ \uparrow & \text{si no} \end{cases} \qquad f_2(\#p) = \begin{cases} 1 & \text{si Dom}\, \Psi_p^{(1)} \neq \emptyset \\ \uparrow & \text{si no} \end{cases}$$

$$f_3(\#p, y) = \begin{cases} 1 & \text{si } y \in \text{Im}\, \Psi_p^{(1)} \\ \uparrow & \text{si no} \end{cases} \qquad f_4(\#p, \#q) = \begin{cases} 1 & \text{si Dom}\, \Psi_p^{(1)} \cap \text{Im}\, \Psi_q^{(1)} \neq \emptyset \\ \uparrow & \text{si no} \end{cases}$$

Ejercicio 17. Sea $f : \mathbb{N} \to \mathbb{N}$ una función computable en tiempo polinomial (i.e., existe un programa p tal que $\Psi_p^{(1)}(x) = f(x)$ y tal que, para algún polinomio q, p no requiere más que $q(\lceil \log_2 x \rceil)$ pasos para terminar). Mostrar que en ese caso f es primitiva recursiva. ¿Sucede lo mismo si la cota es exponencial, doblemente exponencial, etc.? ¿Qué podemos decir, en general, sobre la complejidad temporal de una función computable que no sea primitiva recursiva?

Ejercicio 18. Un programa en el lenguage $\mathcal{F}$ es una secuencia finita de instrucciones que operan sobre un conjunto de variables $X_0, X_1, X_2, \ldots$. Hay tres instrucciones en este lenguaje: $X_i \leftarrow X_i + 1$, $X_i \leftarrow X_i \dot{-} 1$ y REP X_i p, con p un programa de $\mathcal{F}$, cuyo significado es equivalente a ejecutar k veces consecutivas el programa p donde k es el valor de X_i antes de ejecutar la instrucción (la cantidad de veces que p se ejecuta no varía si p modifica X_i).

Denotamos con $\tilde{\Psi}_p^{(n)}(x_1, \ldots, x_n) : \mathbb{N}^n \to \mathbb{N}$ a la función de n variables computada por el programa p; $\tilde{\Psi}_p^{(n)}(a_1, \ldots, a_n)$ corresponde al valor de X_0 al terminar la ejecución de p a partir de la asignación inicial $X_1 = a_1, \ldots, X_n = a_n$ (y el resto de las variables inicializadas en 0).

a. Dado el siguiente programa p determinar cuál de las afirmaciones sobre $\tilde{\Psi}_p^{(2)}(x_1, x_2)$ es correcta:

$$
\begin{array}{l}
p\text{:} \\
\quad \text{REP } X_1 \\
\qquad X_0 \leftarrow X_0 + 1 \\
\qquad \text{REP } X_2 \\
\qquad\quad X_3 \leftarrow X_3 + 1 \\
\quad \text{REP } X_3 \\
\qquad \text{REP } X_0 \\
\qquad\quad X_0 \leftarrow X_0 + 1
\end{array}
$$

- $\tilde{\Psi}_p^{(2)}(x_1, x_2) = 0$
- $\tilde{\Psi}_p^{(2)}(x_1, x_2) = x_1 x_2$
- $\tilde{\Psi}_p^{(2)}(x_1, x_2) = (x_1 x_2)^{x_1 x_2}$
- $\tilde{\Psi}_p^{(2)}(x_1, x_2) = 2^{x_1 x_2} x_1$
- $\tilde{\Psi}_p^{(2)}(x_1, x_2) \uparrow$

b. Demostrar que la clase de funciones $\mathcal{F}$-computables es una clase PRC.

c. Dado un $\mathcal{F}$-programa p, sea $f_p : \mathbb{N} \to \mathbb{N}$ la función que cumple $f_p([a_0, \ldots, a_n]) = [a'_0, \ldots, a'_n]$ donde a'_i es el valor de la variable X_i al finalizar la ejecución de p a partir de la asignación inicial $X_j = a_j$ si $0 \leq j \leq n$ y $X_k = 0$ para $k > n$. Demostrar que f_p es primitiva recursiva.

d. Usando el resultado anterior, mostrar que la clase de funciones $\mathcal{F}$-computables coincide con la clase de funciones primitivas recursivas.

e. Dar una codificación $\tilde{\#}$ de programas en $\mathcal{F}$ usando números naturales que sea biyectiva (y tal que su decodificación sea primitiva recursiva).

f. Mostrar que existe una función primitiva recursiva $t : \mathbb{N} \to \mathbb{N}$ tal que si $e = \tilde{\#}(p)$, entonces $\tilde{\Psi}_p^{(n)}(x_1, \ldots, x_n) = \Phi^{(n)}(x_1, \ldots, x_n, t(e))$ (intuitivamente, t convierte $\mathcal{F}$-programas en $\mathcal{S}$-programas).

g. Sea $\tilde{\Phi}^{(n)} : \mathbb{N}^{n+1} \to \mathbb{N}$ tal que $\tilde{\Phi}^{(n)}(x_1, \ldots, x_n, y) = \Phi^{(n)}(x_1, \ldots, x_n, t(y))$, donde t es la función del punto anterior (i.e., $\tilde{\Phi}^{(n)}$ es un intérprete $\mathcal{S}$-computable de programas en $\mathcal{F}$). Demostrar que $\tilde{\Phi}^{(n)}$ es una función *total*, computable pero que no es primitiva recursiva. (*Sugerencia:* Utilizar una diagonalización.)