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).

$I_k(x) = \overbrace{S(S(\ldots}^{k \text{ veces}}(n(x)))\ldots) = S^{(k)} \circ n$

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:

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

$$g_1(x) = x \dot{-} 1 \qquad g_2(x,y) = x \dot{-} y \qquad g_3(x,y) = \max\{x,y\} \qquad g_4(x,y) = \min\{x,y\}$$

Observaciones: Se asume que $f_4(x,0) = 1$. $\quad x \dot{-} y = \begin{cases} x - y & \text{si } y \leq x \\ 0 & \text{si } y > x \end{cases}$

$f_1(x,0) = \underbrace{u_1^1(x)}_{= I(x)}$ $\quad (x+y) = x + \overbrace{1 + 1 + \ldots + 1}^{y \text{ veces}}$

$f_1(x,t+1) = g(f_1(x,t),x,t)$, $g(F,x,t) = S(u_1^3(F,x,t))$

$f_2(x,0) = I_0(x)$ $\quad (x \cdot y) = \overbrace{x + x + \ldots + x}^{y \text{ veces}}$

$f_2(x,t+1) = g(f_2(x,t),x,t)$, $g(F,x,t) = f_1(u_1^3(F,x,t), u_2^3(F,x,t))$

$f_3(x,0) = I_1(x)$ $\quad (x^y) = \overbrace{x \cdot x \cdot \ldots \cdot x}^{y \text{ veces}}$

$f_3(x,t+1) = g(f_3(x,t),x,t)$, $g(F,x,t) = f_2(u_1^3(F,x,t), u_2^3(F,x,t))$

$f_4(x,0) = I_1(x)$ $\quad \overbrace{x^{x^{\cdot^{\cdot^{x}}}}}^{y \text{ veces}}$

$f_4(x,t+1) = g(f_4(x,t),x,t)$, $g(F,x,t) = f_3(u_1^3(F,x,t), u_2^3(F,x,t))$

$g_1(x,0) = I_0(x)$ $\quad (x \dot{-} 1)$

$g_1(x,t+1) = g(g_1(x,t),x,t)$, $g(G,x,t) = u_3^3(G,x,t)$

$g_2(x,0) = I(x)$ $\quad (x \dot{-} y) = x \overbrace{\dot{-} 1 \ldots \dot{-} 1}^{y \text{ veces}}$

$g_2(x,t+1) = g(g_2(x,t),x,t)$, $g(G,x,t) = g_1(u_1^3(G,x,t))$

$\max(x,y)$:

Sea $\alpha(x) = \begin{cases} 1 & \text{si } x=0 \\ 0 & cc \end{cases}$

$\alpha(0) = 1$

$\alpha(t+1) = g(\alpha(t), t)$, $g(A,t) = I_0(u_1^2(A,t))$] $x \dot{-} y \underset{\swarrow}{\overset{x \le y}{\leq}} 0$, $x \dot{-} y \overset{x>y}{\geq} 1$

Podemos usar α como un "$x \leq y$" porque $\alpha(x \dot{-} y) = \begin{cases} 1 & \text{si } x \leq y \\ 0 & cc \end{cases}$

A partir de acá me confundí y usé u_j^i.

$\Rightarrow \max(x,y) = \overbrace{\alpha(x \dot{-} y)}^{x \leq y} \cdot y + \overbrace{\alpha(\alpha(x \dot{-} y))}^{x>y} \cdot x$] Ojo al piojo que acabamos de descubrir como hacer funciones partidas

$\min(x,y) = \alpha(x \dot{-} y) \cdot x + \alpha(\alpha(x \dot{-} y)) \cdot y$]

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

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

y sea $\mathcal{C}_c$ la (mínima) clase que extiende a $\mathcal{C}_i$ y se encuentra cerrada por composición, i.e., si $f, g_1, \ldots g_m$ están en $\mathcal{C}_c$, entonces $h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ 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 $f(x_1, \ldots, x_n) = k$, o bien para algún i fijo, se tiene $f(x_1, \ldots, x_n) = x_i + k$.

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

a) Lo hacemos por inducción estructural

CB:

$n(x) = 0$, $k = 0$] ✓ , $s(x) = x+1$, $k=1$] ✓ , $u_i^n(\bar{x}) = x_i$, $k=0$] ✓

PI:

Sea $f : \mathbb{N}^n \to \mathbb{N}$, $h : \mathbb{N}^k \to \mathbb{N} \in \mathcal{C}_c$, $g_{1,\ldots,k} : \mathbb{N}^m \to \mathbb{N} \in \mathcal{C}_c$.

qvq: $f(\bar{x}) = k$ o para algún i fijo, se tiene $f(\bar{x}) = x_i + k$

$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x}))$

Dotamos, h y $g_{1,\dots,k}$ cumplen HI por pertenecer a C_c.

$\Rightarrow h(g_1(\bar{x}), \dots, g_k(\bar{x})) = k$ o $h(g_1(\bar{x}), \dots, g_k(\bar{x})) = x_i + k$, i fijo

1) $h(g_1(\bar{x}), \dots, g_k(\bar{x})) = k$

$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x})) = k \Rightarrow f(\bar{x}) = k$ ✓

2) $h(g_1(\bar{x}), \dots, g_k(\bar{x})) = g_i(\bar{x}) + k$

$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x})) = g_i(\bar{x}) + k$

Tenemos de nuevo 2 opciones:

2.1) $g_i(\bar{x}) = k'$

$\Rightarrow f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x})) = g_i(\bar{x}) + k = k' + k = k''$ ✓

2.2) $g_i(\bar{x}) = x_i + k'$

$\Rightarrow f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x})) = g_i(\bar{x}) + k = x_i + k' + k = x_i + k''$ ✓

Luego por inducción, f está en $C_c \Leftrightarrow \exists k \geq 0 / f(\bar{x}) = k$ o $f(\bar{x}) = x_i + k$.

b) $f(x) = x \dot{-} 1$ es P.R. pero no pertenece a C_c porque no la podés conseguir <u>solo</u> usando composición.

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

En esta parte vamos a usar mucho $\alpha(x) = \begin{cases} 1 & x=0 \\ 0 & cc \end{cases}$ definido en punto 2.

$x \leqslant y = \alpha(x \dot{-} y)$ / $x \geqslant y = \alpha(y \dot{-} x)$ / $x = y = (x \leqslant y) \cdot (y \leqslant x)$ / $x \neq y = \alpha(x = y)$

$x < y = \alpha(x \geqslant y)$ / $x > y = \alpha(x \leqslant y)$

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

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

Observar que h queda completamente determinada por este esquema.

Esto vale porque $h(\overline{x}) = \left[\sum_{i=1}^{k} f_i(\overline{x}) \cdot P_i(\overline{x})\right] + g(\overline{x}) \cdot \alpha\left(\sum_{i=1}^{k} P_i(\overline{x})\right)$

Como la suma, mult, g, f_i, P_i y todo lo que usé está en C, h es composición de cosas en $C \Rightarrow h$ está en C.

Ejercicio 6. a. Demostrar que el predicado $\text{par}(x) = \begin{cases} 1 & \text{si } x \text{ es par} \\ 0 & \text{si no} \end{cases}$ está en toda clase *PRC*.

$\text{par}(0) = 1$

$\text{par}(t+1) = g(\text{par}(t), t)$, $g(P,t) = \alpha(P)$

b. Demostrar que la función $f(x) = \lfloor x/2 \rfloor$ está en toda clase *PRC*.

$f(0) = 0$

$f(t+1) = g(f(t), t)$, $g(F,t) = \begin{cases} F & \text{si } \text{par}(t) \\ F+1 & cc \end{cases}$

c. 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:

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

Observar que h queda completamente determinada por este esquema.

La reescribimos como algo conocido y listo.

$h(\bar{x}, 0) = f(\bar{x})$] ✓ BELLISIMO CASO BASE

$h(\bar{x}, t+1) = g(h(\bar{x}, 0), \bar{x}, t)$ Vamos a tratar de meter a g_1 y a g_2 en g

$$g(H, \bar{x}, t) = \begin{cases} g_1(\bar{x}, k_1(t), H), & \text{si par}(t) \\ g_2(\bar{x}, k_2(t), H), & \text{cc} \end{cases}$$

$t\cancel{+1} = 2k\cancel{+1} \Rightarrow k = \frac{t}{2} \Rightarrow k_1(t) = \lfloor t/2 \rfloor$

$t\cancel{+1} = 2k+\cancel{2}^{+1} \Rightarrow k = \frac{t-1}{2} \Rightarrow k_2 = k_1 = \lfloor t/2 \rfloor$

BELLISIMO PASO RECURSIVO

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:

$$\text{cantidad}_p(x_1, \dots, x_n, y, z) = |\{t \mid y \le t \le z \wedge p(x_1, \dots, x_n, t)\}|$$

Observación: pueden usarse los operadores acotados (mín, Σ, $\forall$, $\exists$) vistos en la teórica.

$\text{cantidad}_p(\bar{x}, y, z) = \sum_{i=y}^{z} p(\bar{x}, i)$

$$\text{todos}_p(x_1, \dots, x_n, y, z) = \begin{cases} 1 & \text{si } (\forall t : y \le t \le z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases}$$

$\text{todos}_p(\bar{x}, y, z) = (\forall t)_{\le z}[t < y \vee p(\bar{x}, t)]$

$$\text{alguno}_p(x_1, \dots, x_n, y, z) = \begin{cases} 1 & \text{si } (\exists t : y \le t \le z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases}$$

$\text{alguno}_p(\bar{x}, y, z) = (\exists t)_{\le z}[t \ge y \wedge p(\bar{x}, t)]$

$$\text{minimo}_p(x_1,\dots,x_n,y,z) = \begin{cases} \text{mín}\{t \mid y \le t \le z \wedge p(x_1,\dots,x_n,t)\} & \text{si existe tal t} \\ 0 & \text{si no} \end{cases}$$

$$\text{minimo}_p(\bar{x},y,z) = \min_{t \le z} \left[t \ge y \wedge p(\bar{x},t) \right]$$

$$\text{maximo}_p(x_1,\dots,x_n,y,z) = \begin{cases} \text{máx}\{t \mid y \le t \le z \wedge p(x_1,\dots,x_n,t)\} & \text{si existe tal t} \\ 0 & \text{si no} \end{cases}$$

no existe nadie mayor que cumpla p

$$\text{maximo}_p(\bar{x},y,z) = \min_{t \le z} \left[t \ge y \wedge p(\bar{x},t) \wedge \neg\left((\exists t')_{\le z} \left[t' > t \wedge p(\bar{x},t') \right] \right) \right]$$

$$\text{unico}_p(x_1,\dots,x_n,y,z) = \begin{cases} u & \text{si } \{u\} = \{t \mid y \le t \le z \wedge p(x_1,\dots,x_n,t)\} \\ z+1 & \text{si no} \end{cases}$$

$$\text{unico}_p(\bar{x},y,z) = \begin{cases} \text{min}_p(\bar{x},y,z) & , \text{ si } \text{min}_p(\bar{x},y,z) = \text{max}_p(\bar{x},y,z) \overset{\exists \min}{\neq 0} \\ z+1 & , \text{ cc} \end{cases}$$

Ejercicio 8. Mostrar que las siguientes funciones están en toda clase *PRC*:

$$\text{cociente}(x,y) = \lfloor x/y \rfloor$$

Observación: Se asume que cociente$(x,0) = 0$ y resto$(x,0) = x$.

$$\text{cociente}(x,y) = \min_{t \le x} \left[(t+1)\cdot y > x \right]$$

$$\text{resto}(x,y) = x \bmod y$$

$x = y \cdot \text{cociente}(x,y) + \text{resto}(x,y) \Rightarrow \text{resto}(x,y) = x - y \cdot \text{cociente}(x,y)$

$$\text{resto}(x,y) = x - y \cdot \text{cociente}(x,y)$$

$$\text{divide}(x,y) = \begin{cases} 1 & \text{si } x \text{ divide a } y \\ 0 & \text{si no} \end{cases}$$

$$\text{divide}(x,y) = (\exists t)_{\le y} \left[t \cdot x = y \right]$$

$$\text{primo}(x) = \begin{cases} 1 & \text{si } x \text{ es un número primo} \\ 0 & \text{si no} \end{cases}$$

$\text{primo}(x) = \alpha\left((\exists t)_{<x}\left[t > 1 \wedge \text{divide}(t, x)\right]\right)$

$$\text{raiz}(x, y) = \begin{cases} \lfloor \sqrt[x]{y} \rfloor & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases}$$

$\text{raiz}(x,y) = \min\limits_{t \leq y} \left[t^{x+1} > y \right]$

$\text{nprimo}(n) = k$ sii k es primo y hay sólo $n-1$ primos positivos menores que k

$\text{nprimo}(n) = \min\limits_{t \leq n!} \left[\text{cantidad}_{\text{Primos}}(t, 1, t-1) = n-1 \right]$

Ejercicio 9. Considerar la codificación de pares de naturales dada por $\langle x, y\rangle = 2^x(2y+1) \dot{-} 1$. Mostrar que las funciones observadoras $l, r : \mathbb{N} \to \mathbb{N}$ tales que $l(\langle x, y\rangle) = x$ y $r(\langle x, y\rangle) = y$ están en toda clase *PRC*.

$\langle x,y \rangle$

$l(z) = \min\limits_{x \leq z} \left[(\exists y)_{\leq z} \left[z = \langle x, y \rangle \right] \right]$

$r(z) = \min\limits_{y \leq z} \left[(\exists x)_{\leq z} \left[z = \langle x, y \rangle \right] \right]$

Esto vale porque cada z tiene asociado exactamente un par <x,y>.

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

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

Definimos $FIB(x) = [fib(x), fib(x+1)]$

$\Rightarrow fib(x) = FIB(x)[1]$

$FIB(0) = [0, 1]$

$FIB(t+1) = g(FIB(t), t)$, $g(F, t) = [F[2], F[1] + F[2]]$

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:

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

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

Observar que h_1 y h_2 quedan completamente determinadas por el esquema de recursión mutua.

$H(\bar{x},t) = [h_1(\bar{x},t), h_2(\bar{x},t)]$

$\Rightarrow h_1(\bar{x},t) = H(\bar{x},t)[1]$ y $h_2(\bar{x},t) = H(\bar{x},t)[2]$

$H(\bar{x},0) = [f_1(\bar{x}), f_2(\bar{x})]$

$H(\bar{x},t+1) = g(H(\bar{x},t),\bar{x},t)$, $g(H,\bar{x},t) = [g_1(H[1],H[2],\bar{x},t), g_2(H[2],H[1],\bar{x},t)]$

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 observadoras $l, r : \mathbb{N} \to \mathbb{N}$ 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 $f^{(n)} = \underbrace{f \circ f \circ \cdots \circ f}_{n}$).

Que cambia meter a <.,.>, l(z) y r(z) enlas iniciales?
No se las podia obtener con recursion/composicion y listo?

a) Demostrar que $\mathcal{C}_{Ack} \subset PR$.

qvq: Toda $f \in C_{ACK}$ se puede obtener mediante composición / recursión de C_i

Lo hacemos por inducción

CB: C_i, $\langle\cdot,\cdot\rangle$, $l(\cdot)$ y $r(\cdot)$ son PR ✓

PI:

- COMPOSICION ∴ PR por definición

- ITERACION:

$h(x,n) = f^{(n)}(x)$

$h(x,0) = x$

$h(x,t+1) = g(h(x,t),x,t)$, $g(H,x,t) = f(H)$

Lo puedo escribir usando solo composición y recursión ✓

$\Rightarrow C_{ACK} \subset PR$

b) Observar que en $\mathcal{C}_{Ack}$ se tienen las funciones codificadoras de n-tuplas y sus observadoras.

Observado :)

c) Demostrar que si $f : \mathbb{N}^n \to \mathbb{N}$ y $g : \mathbb{N}^{n+2} \to \mathbb{N}$ pertenecen a la clase $\mathcal{C}_{Ack}$ y h se obtiene mediante el esquema de recursión primitiva a partir de f y g, entonces la función $s : \mathbb{N}^{n+1} \to \mathbb{N}$ definida por $s(\overline{x}, y) = \langle \overline{x}, y, h(\overline{x}, y)\rangle$ también pertenece a la clase $\mathcal{C}_{Ack}$.

HIP

$h(\bar{x}, 0) = f(\bar{x})$

$h(\bar{x}, t+1) = g(h(\bar{x}, t), \bar{x}, t)$

qvq: $s(\bar{x}, y) = \langle \bar{x}, y, h(\bar{x}, y)\rangle \in C_{Ack}$

Esto vale porque <*,*,*> pertenece a C_{Ack} y h(X,y) pertenece a C_{Ack} por haberlo conseguido mediante composicion/recursion de funciones que tambien pertenecian a C_{Ack} (f y g).

d) Concluir que $PR \subset \mathcal{C}_{Ack}$ y, por lo tanto, coinciden.

qvq: Toda función que se puede conseguir mediante composición/recursión de $c_i \in C_{ACK}$?

Ejercicio 13. Considerar la codificación de secuencias finitas de números naturales dada por $[a_1, \ldots, a_n] = \Pi_{i=1}^{n} \text{nprimo}(i)^{a_i}$, 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.

qvq: $\forall [a_1, \ldots, a_n]$ no terminada en 0 $\exists ! \prod_{i=1}^{n} \text{nPrimo}(i)^{a_i}$

Por un teorema de algebra, $z = p_1^{k_1} \cdot \ldots \cdot p_n^{k_n}$ y esa descomposición en primos es única. (Todos $k_i \neq 0$)
La lista es biyectiva con descomposición $\Rightarrow$ Como no puedo meter 0_s a derecha $\exists !$ lista para cada nro y cada nro tiene una unica lista.

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 $|[a_1, \ldots, a_n]| = n$ (longitud)
- $\cdot[i] : \mathbb{N} \to \mathbb{N}$ tal que $[a_1, \ldots, a_n][i] = \begin{cases} a_i & \text{si } 1 \leq i \leq n \\ 0 & \text{si no} \end{cases}$ (observador)
- $[\cdot] : \mathbb{N} \to \mathbb{N}$ 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)

$[\,] = 1$

- $|n| = \min_{t \leq n} \text{divide}(\text{nPrimo}(t), z) \wedge \neg(\exists t')_{\leq n} [t' > t \wedge \text{divide}(t', n)]$
- $n[i] = \min_{t \leq n} \neg \text{divide}\left(\text{nPrimo}(i)^{t+1}, n\right)$
- $[x] = 2^x$
- $l_1 \circ l_2 = l_1 \cdot \prod_{i=1}^{|l_2|} \text{nPrimo}(i + |l_1|)^{l_2[i]}$
- $\text{sub}(l, i, j) = \prod_{k=1}^{j-i} \text{nPrimo}(k)^{l[i]+k-1}$

Tomamos una secuencia cualquiera [x1,...,xn,0...,0] y la codificamos como la tupla <[x1,...,xn],#0>. Esto está buenardo, pero pero pero nosotros queremos tener una biyección con los naturales. Para eso nos alcanza con tener una biyección con las tuplas (porque las tuplas son biyectivas con los naturales). El tema con nuestra codificación berreta es que no hay ninguna lista que le corresponda a los pares de la forma <0,x>. Igual lo bueno es que podemos arreglar esto re isi, nomas tenemos que cambiar la tupla de <[x1,...,xn],#0> a <[x1,...,xn]-1,#0> y ya estamos para salir a las pistas rey.

Ejercicio 14. Utilicemos [] para referirnos a la codificación de secuencias dada en el punto 13.c.

a. 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 $\mathcal{C}$.

Observar que h queda completamente determinada por el esquema de recursión global.

$H(\bar{x}, 0) = [f([], \bar{x})] = [h(\bar{x}, 0)]$

$[h(0), \ldots h(t)] \circ$

$H(\bar{x}, t+1) = g(H(\bar{x}, t), \bar{x}, t) \quad , \quad g(H, \bar{x}, t) = H \circ \overbrace{f(H, \bar{x})}^{h(\bar{x}, t+1)}$

$h(\bar{x}, t) = H(\bar{x}, t)[|H(\bar{x}, t)|]$

b. Demostrar, a partir del ítem anterior, que dada $\mathcal{C}$, una clase *PRC*, y funciones $g_1 : \mathbb{N}^n \to \mathbb{N}$, $g_2 : \mathbb{N}^{n+2} \to \mathbb{N}$ en $\mathcal{C}$, la función definida como

$$\begin{aligned} h(x_1, \ldots, x_n, 0) &= g_1(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n, t+1) &= g_2([h(x_1, \ldots, x_n, 0), \ldots, h(x_1, \ldots, x_n, t)], x_1, \ldots, x_n, \underline{t}) \end{aligned}$$

único cambio que noté

también está en $\mathcal{C}$.

Observar que h queda completamente determinada por el esquema de recursión global.

NO ES LO MISMO?

Ejercicio 15. Demostrar que toda clase PRC se encuentra cerrada por *recursión doble*. Es decir, dadas $f : \mathbb{N}^3 \to \mathbb{N}$ y $g : \mathbb{N}^4 \to \mathbb{N}$ pertenecientes a $\mathcal{C}$, una clase PRC, demostrar que también está en $\mathcal{C}$ la función $h : \mathbb{N}^3 \to \mathbb{N}$ que cumple:

$$\begin{aligned} h(x,0,z) &= f(x,0,z) \\ h(x,y,0) &= f(x,y,0) \\ h(x,y+1,z+1) &= g(x,y,z,h(x,y,z)) \end{aligned}$$

Observar que h queda completamente determinada por el esquema de recursión doble.

Si te das cuenta, para calcular h(x,y,z) siempre vas a terminar llegando a alguno de los casos base con el valor h(x, y-M, z-M) con M=min(y,z).
Entonces vamos a hacer un estilo de buttom-up y vamos a agarrar ese resultado e ir consiguiendo desde ahi los h(x,y+1,z+1).
Veamos que h(x,y-M+1,z-M+1) = g(x,y-M,z-M,h(x,y-M,z-M)).
Por otro lado, para seguir aumentando y,z solo necesitamos aplicar g de forma iterativa hasta llegar al valor pedido. Como vimos que C está cerrado por iteración de funciones unarias, entonces si logramos escribir a h usando composición/iteracion unaria de cosas pertenecientes a C => h pertenece a C.

$h(x,y,z) = h'(x,y,z)$

$M = \min(y,z)$

$h'(x,y,z) = [x, y-M, z-M, f(x,y-M,z-M)]$

$g'(l) = [l[1], l[2]+1, l[3]+1, g(l[1], l[2], l[3], l[4])]$

$h(x,y,z) = g'^{(M)}(h'(x,y,z))[4]$