# Guía 1 - Funciones primitivas recursivas y clases PRC

#### Solución de un alumno

#### Verano 2021

#### Ejercicio 1

$$f(x) = k = s(s(s(...s(n(x))))) = s^k(n(x))$$

Siendo s(x) la función sucesor y n(x) la función cero.

#### Ejercicio 2

$$\begin{split} g_1(x) &= x \dot{-} 1 \\ g_1(0) &= 0 \\ g_1(t+1) &= t = u_2^2(g(t),t) \\ g_3(x,y) &= \max\{x,y\} = x \cdot (x > y) + y \cdot (x \le y) \end{split}$$

donde  $x \leq y$  es  $\alpha(\dot{x-y})$  y x > y es  $\neg \alpha(\dot{x-y})$ 

# Ejercicio 3

a)

Qvq todas las funciones de la clase  $C_c$  se pueden escribir como  $f(x_1,...,x_n)=k$  ó  $f(x_1,...,x_n)=x_i+k$  con un i fijo.

Demostremos por inducción estructural:

CASO BASE: Las funciones iniciales se ve que se pueden escribir así. Listo.

**PASO INDUCTIVO:** Sea h una función obtenida por composición de funciones de  $C_c$ , entonces h tiene la pinta de:

$$h(x_1, ..., x_n) = f(g_1(x_1, ..., x_n), ..., g_m(x_1, ..., x_n))$$

$$\Rightarrow h(x_1, ..., x_n) = g_i(x_1, ..., x_n) + k$$

$$\delta$$

$$h(x_1, ..., x_n) = k$$

El segundo caso ya cumple. En el primer caso:

$$h(x_1, ..., x_n) = g_i(x_1, ..., x_n) + k$$

$$\Rightarrow h(x_1, ..., x_n) = x_j + k' + k = x_j + k''$$

$$ó$$

$$h(x_1, ..., x_n) = k' + k = k''$$

Ambos casos cumplen.

 $f(x_1,...,x_n)=k$  está en  $C_c$  por ejercicio 1.

 $f(x_1,...,x_n)=x_i+k=u_i^n(x_1,...,x_n)+k$  entonces está en  $C_c$  por composición.

b)  $g_2(x,y) = \dot{x-y}$  se puede demostrar con a).

#### Ejercicio 4

## Ejercicio 6

$$par(x) = \begin{cases} 1 & si \ x \ es \ par \\ 0 & si \ no \end{cases}$$

$$par(0) = 1$$
$$par(x+1) = \neg par(x)$$

## Ejercicio 7

$$\begin{split} & cantidad_p(x_1,...,x_n,y,z) = \sum_{t=y}^z p(x_1,...,x_n,t) \\ & todos_p(x_1,...,x_n,y,z) = (cantidad_p(x_1,...,x_n,y,z) = z - y) \\ & alguno_p(x_1,...,x_n,y,z) = (cantidad_p(x_1,...,x_n,y,z) \geq 1) \\ & min(x_1,...,x_n,y,z) = \sum_{u=y}^z \prod_{t=0}^u \alpha(p(x_1,...,x_n,t)) \\ & minimo(x_1,...,x_n,y,z) = (min(x_1,...,x_n,y,z) \neq z + 1) \cdot min(x_1,...,x_n,y,z) \\ & unico_p(x_1,...,x_n,y,z) = [maximo(x_1,...,x_n,y,z) = minimo(x_1,...,x_n,y,z)] \cdot maximo(x_1,...,x_n,y,z) + \alpha(p(x_1,...,x_n,maximo(x_1,...,x_n,y,z))) \cdot (z + 1) \end{split}$$

## Ejercicio 8

$$cociente(x, y) = min_{t \le x}((t+1) \cdot y > x)$$

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

$$divide(x, y) = \alpha(resto(y, x))$$

$$primo(x) = \neg(\exists t)_{1 < t < x}(divide(t, x))$$

$$raiz(x) = min_{t < y}((t+1)^x > y)$$

#### Ejercicio 9

$$l(z) = min_{x \le z}((\exists y)_{\le z} z = < x, y >)$$
  
 $r(z) = min_{y < z}((\exists x)_{< z} z = < x, y >)$ 

#### Ejercicio 10

$$f(x) = \begin{cases} 0 & x = 0\\ 1 & x = 1\\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

Observamos que cada elemento de la secuencia utiliza dos anteriores, entonces podemos crear una función similar pero que nos devuelva dos valores:

$$F(0) = <0, 1 >$$

$$F(n+1) = < r(F(n)), r(F(n)) + l(F(n)) >$$

Entonces:

F(0) = <0, 1>

F(1) = <1, 1>

F(2) = <1, 2>

F(3) = <2, 3>

F(4) = <3, 5>

F(5) = <5, 8>

Y siempre vamos acumulando la suma de fibonacci a la derecha y a la izquierda nos guardamos un valor auxiliar (el anterior valor de la sucesión) para construir el siguiente.

Veamos que F es p.r.:

 $\langle \rangle, r(), l()$  y + son p.r., entonces F es p.r. por ser una recursiva primitiva de composiciones.

Entonces f(x) de fibonacci si la definimos por composición nos queda:

$$f(n) = r(F(n-1)) \cdot (n > 0)$$

#### Ejercicio 11

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

$$h_2(\bar{x},t) = \begin{cases} f_2(\bar{x}) & \text{si } t = 0\\ g_2(h_2(\bar{x},t-1),h1(\bar{x},t-1),\bar{x},t) & \text{si no} \end{cases}$$

Utilicemos la misma idea que el ejercicio de fibonacci, nos guardamos en la primer componente el valor de  $h_1$  y en la segunda el valor de  $h_2$ .

$$H(0,x) = \langle f_1(x), f_2(x) \rangle$$

$$H(n+1,x) = \langle g_1(l(H(n,x)), r(H(n,x)), ...), g_2(r(H(n,x)), l(H(n,x)), ...) \rangle$$

H(x,y) pertenece a la clase C por ser recursiva primitiva con composición de funciones que estaban en C

Entonces:

$$h_1(x,t) = l(H(t,x))$$

$$h_2(x,t) = r(H(t,x))$$

Y ambas están en C.

## Ejercicio 12

# Ejercicio 13

- a) INYECTIVA: Dos secuencias diferentes van a tener diferente factorización de primos (si no termina en ceros) entonces van a tener diferente número natural asociado (porque la factorización de primos es única para cada número natural).
  - SOBREYECTIVA: Todo número natural mayor que cero tiene factorización de primos, entonces tiene una secuencia asociada.

b)

 $|\cdot| = maximo_{x < \cdot} \{x : nprimo(x) | \cdot \}$ 

 $\cdot [i] = maximo\{x : nprimo(i)^x | \cdot \}$ 

 $[x] = nprimo(1)^x$ 

 $\cdot_1 \circ \cdot_2 = \cdot_1 \cdot \prod_{i=1}^m nprimo(i + |\cdot_1|)^{\cdot_2[i]}$ 

 $sub(s, i, j) = \prod_{z=1}^{j} nprimo(z)^{s[z+i]}$ 

c) Proponemos codificar una lista como:

$$[a_1,...,a_n] = < \prod_{i=1}^n nprimo(i)^{a_i}, n >$$

Siendo la parte izquierda la codificación de primos anterior y la parte derecha la longitud de la lista, así pudiendo guardar listas que terminen con ceros. Veamos que los naturales y las secuencias forman una biyección.

INYECTIVA: Dos secuencias diferentes van a tener diferente factorización de primos o diferente longitud. Cuando alguno de ambos es diferentes entonces la codificación de pares va a ser diferente (porque es inyectiva).

SOBREYECTIVA: como la codificación de pares es sobreyectiva, entonces existen a, b tal que  $c = \langle a, b \rangle$  entonces toda codificación tiene su propia secuencia asociada.

Ahora no importa si la secuencia termina con ceros, va a tener su número propio.

$$\begin{split} |\cdot| &= r(\cdot) \\ \cdot [i] &= l(\cdot)[i] \\ [\cdot] &= < [l(\cdot)], 1 > \\ \cdot_1 \circ \cdot_2 &= < l(\cdot_1) \circ l(\cdot_2), r(\cdot_1) + r(\cdot_2) > \\ sub(x, i, j) &= < sub(l(x)), j - i + 1 > \end{split}$$

## Ejercicio 14

a) Misma idea que fibonacci pero ahora nos tenemos que guardar todos los valores anteriores.

$$H(x,0) = [f([],x)]$$

$$H(x,t) = H(x,t-1) \circ [f(H(x,t-1),x)]$$

Entonces:

$$h(x,t) = H(x,t)[|H(x,t)|]$$

b) Casi igual al a).