Lógica y Computabilidad

Práctica 1: Funciones primitivas recursivas y clases PRC

2do cuatrimestre 2022

Ejercicio 1

Para construir una constante k, aplicamos la función s (sucesor) unas k veces, partiendo inicialmente de la función n que nos devuelve el 0.

$f(x) = k = (\underbrace{s \circ \cdots \circ s}_{k \text{ veces}} \circ n)(x) = s^k(n(x))$

Ejercicio 2

- $f_1(x, y) = suma(x, y) = x + y$
  $suma(x, 0) = u_1^1(x) = x$
  $suma(x, y + 1) = g(suma(x, y), x, y)$ donde $g(x_1, x_2, x_3) = s(u_1^3(x_1, x_2, x_3))$
  $\Rightarrow suma(x, y + 1) = s(suma(x, y))$
- $f_2(x, y) = prod(x, y) = x \cdot y$
  $prod(x, 0) = n(x) = 0$
  $prod(x, y + 1) = g(prod(x, y), x, y)$ donde $g(x_1, x_2, x_3) = suma(u_1^3(x_1, x_2, x_3), u_2^3(x_1, x_2, x_3))$
  $\Rightarrow prod(x, y + 1) = suma(prod(x, y), x)$
- $f_3(x, y) = pot(x, y) = x^y$
  $pot(x, 0) = s(n(x)) = 1$
  $pot(x, y + 1) = g(pot(x, y), x, y)$ donde $g(x_1, x_2, x_3) = prod(u_1^3(x_1, x_2, x_3), u_2^3(x_1, x_2, x_3))$
  $\Rightarrow pot(x, y + 1) = prod(pot(x, y), x)$
- $f_4(x, y) = \underbrace{x^{x^{\cdot^{\cdot^{x}}}}}_{y \text{ veces}}$
  $f_4(x, 0) = 1$
  $f_4(x, y + 1) = g(f_4(x, y), x, y)$ donde $g(x_1, x_2, x_3) = pot(u_2^3(x_1, x_2, x_3), u_1^3(x_1, x_2, x_3))$
  $\Rightarrow f_4(x, y + 1) = pot(x, f_4(x, y))$
  Esta función a veces se la llama “Power Tower” (Wikipedia)
- $g_1(x) = pred(x) = x \dot{-} 1$
  $pred(0) = n() = 0$ Permitimos utilizar la función nula n sin parámetros.
  $pred(x + 1) = g(pred(x), x)$ donde $g(x_1, x_2) = u_2^2(x_1, x_2) = x_2$
  $\Rightarrow pred(x + 1) = x$
- $g_2(x, y) = resta(x, y) = x \dot{-} y$
  $resta(x, 0) = u_1^1(x) = x$
  $resta(x, y + 1) = g(resta(x, y), x, y)$ donde $g(x_1, x_2, x_3) = pred(u_1^3(x_1, x_2, x_3))$
  $\Rightarrow resta(x, y + 1) = pred(resta(x, y))$

- $g_3(x, y) = \max\{x, y\}$

  $g_3(x, y) = suma(resta(x, y), y) = (x \dot{-} y) + y$

  Si $x \geqslant y$, entonces g_3 simplemente resta y suma y a un x que es más grande, y en efecto terminamos con x que era el máximo. En el otro caso $x < y$, al hacer la resta en $\mathbb{N}$ obtenemos $x \dot{-} y = 0$, luego al sumar y obtenemos nuevamente y que era el máximo.

- $g_4(x, y) = \min\{x, y\}$

  $g_4(x, y) = resta(suma(x, y), \max\{x, y\}) = x + y - \max\{x, y\}$

Ejercicio 3

a)

Para la ida ($\Rightarrow$) hacemos una demostración por inducción estructural. Primero probamos que todas las funciones iniciales cumplen la propiedad.

- Función nula

  $f(x) = n(x) = 0$. La función nula cae en el caso $f(x) = k$ donde $k = 0$.

- Función sucesor

  $f(x) = s(x) = x + 1$. La función sucesor cae en el caso $f(x) = x + k$ donde $k = 1$.

- Función proyector

  $f(x_1, \ldots, x_n) = u_i^n(x_1, \ldots, x_n) = x_i$. La función proyector cae en el caso $f(x_1, \ldots, x_n) = x_i + k$ donde $k = 0$.

Paso inductivo. Supongamos que existe $h_m \in \mathcal{C}_c$ generada a partir de m composiciones, tal que $h_m(x_1, \ldots, x_n) = k$ o bien $h_m(x_1, \ldots, x_n) = x_i + k$. Queremos ver si cualquier $h_{m+1} \in \mathcal{C}_c$ también cumple la propiedad. Para generar h_{m+1} componemos h_m con alguna función $f \in \mathcal{C}_c$.

- Caso $f(x) = n(x)$

  $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = n(h_m(x_1, \ldots, x_n)) = 0$.

  No importa la forma de h_m pues $n(x) = 0$ para cualquier x.

- Caso $f(x) = s(x)$

  - Caso $h_m(x_1, \ldots, x_n) = x_i + q$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = s(x_i + q) = x_i + q + 1 = x_i + k$ donde $k = q + 1$.
  - Caso $h_m(x_1, \ldots, x_n) = q$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = s(q) = q + 1 = k$ donde $k = q + 1$.

- Caso $f(x) = u_i^n(x)$

  Como $h_m : \mathbb{N}^n \to \mathbb{N}$, necesariamente $f(x) = u_1^1(x)$ para poder componer $f \circ h_m$.

  - Caso $h_m(x_1, \ldots, x_n) = x_i + k$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = u_1^1(x_i + k) = x_i + k$.
  - Caso $h_m(x_1, \ldots, x_n) = k$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = u_1^1(k) = k$.

- Caso $f(x) = x + r$

  - Caso $h_m(x_1, \ldots, x_n) = x_i + q$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = x_i + q + r = x_i + k$ donde $k = q + r$.
  - Caso $h_m(x_1, \ldots, x_n) = q$

    $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = q + r = k$ donde $k = q + r$.

- Caso $f(x) = k$

- Caso $h_m(x_1, \ldots, x_n) = x_i + q$
  $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = f(x_i + q) = k$.
- Caso $h_m(x_1, \ldots, x_n) = q$
  $h_{m+1} = f(h_m(x_1, \ldots, x_n)) = f(q) = k$.

Por lo tanto, partiendo de una función $h_m \in \mathcal{C}_c$, vemos que al realizar una composición con alguna función $f \in \mathcal{C}_c$ obtenemos una función $h_{m+1} \in \mathcal{C}_c$ (pues $\mathcal{C}_c$ es cerrado por composición) que mantiene la propiedad enunciada.

Para la vuelta ($\Leftarrow$) mostramos que podemos construir cualquier $f(x_1, \ldots, x_n) = k$ o $f(x_1, \ldots, x_n) = x_i + k$ a partir de composición de las funciones iniciales, y por lo tanto $f \in \mathcal{C}_c$.

- $f(x_1, \ldots, x_n) = k = s^k(n(x_1, \ldots, x_n))$
- $f(x_1, \ldots, x_n) = x_i + k = s^k(u_i^n(x_1, \ldots, x_n))$

b)

En el ejercicio 2 vimos que la función $suma(x, y) = x + y$ es P.R. pero $suma \notin \mathcal{C}_c$ pues no cumple con la propiedad.

Ejercicio 4

Pendiente

Ejercicio 5

Pendiente

Ejercicio 6

Pendiente

Ejercicio 7

Pendiente

Ejercicio 8

Pendiente

Ejercicio 9

Pendiente

Ejercicio 10

Pendiente

Ejercicio 11

Pendiente

Ejercicio 12

Pendiente

Ejercicio 13

Pendiente

Ejercicio 14

Pendiente

Ejercicio 15

Pendiente