Funciones primitivas recursivas y clases PRC (Parte 1)

Lógica y Computabilidad - Curso de verano 2018

Franco Frizzo

(Basado en una clase de Pablo Ariel Heiber)

31 de enero de 2018

1. Funciones iniciales y composición

Repaso

- Llamamos **funciones iniciales** a las siguientes funciones:
  - $\mathrm{n} : \mathbb{N} \to \mathbb{N}, \quad \mathrm{n}(x) = 0.$
  - $\mathrm{s} : \mathbb{N} \to \mathbb{N}, \quad \mathrm{s}(x) = x + 1.$
  - $\mathrm{u}_i^n : \mathbb{N}^n \to \mathbb{N}, \quad \mathrm{u}_i^n(x_1, \ldots, x_n) = x_i, \qquad \text{con } n \geqslant 1,\ i \in \{1, \ldots, n\}.$
- Una función $h : \mathbb{N}^n \to \mathbb{N}$ se obtiene por **composición** a partir de $f : \mathbb{N}^k \to \mathbb{N}$ y de $g_1, \ldots, g_k : \mathbb{N}^n \to \mathbb{N}$ si se puede escribir como

$$h(x_1, \ldots, x_n) = f\big(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)\big).$$

- Una clase de funciones $\mathcal{C}$ está **cerrada por composición** si cada vez que una función f se puede obtener por composición a partir de funciones de $\mathcal{C}$, f pertenece también a $\mathcal{C}$.

Ejercicio 1

Sea $\mathcal{C}_c$ la mínima clase que contiene a las funciones iniciales y está cerrada por composición. Determinar si están en $\mathcal{C}_c$ las siguientes funciones.

a. $\mathrm{id} : \mathbb{N} \to \mathbb{N}, \quad \mathrm{id}(x) = x.$

b. $\mathrm{uno} : \mathbb{N} \to \mathbb{N}, \quad \mathrm{uno}(x) = 1.$

c. $\mathrm{s}_1^2 : \mathbb{N}^2 \to \mathbb{N}, \quad \mathrm{s}_1^2(x, y) = x + 1.$

d. $\mathrm{flip}_h : \mathbb{N}^2 \to \mathbb{N}, \quad \mathrm{flip}_h(x, y) = h(y, x), \quad$ sabiendo que $h : \mathbb{N}^2 \to \mathbb{N} \in \mathcal{C}_c$.

e. $h_i^n : \mathbb{N}^n \to \mathbb{N}, \quad h_i^n(x_1, \ldots, x_n) = h(x_i), \quad$ con $i \in \{1, \ldots, n\}$, sabiendo que $h : \mathbb{N} \to \mathbb{N} \in \mathcal{C}_c$.

f. $\mathrm{diag}_h : \mathbb{N} \to \mathbb{N}, \quad \mathrm{diag}_h(x) = h(x, \ldots, x), \quad$ sabiendo que $h : \mathbb{N}^k \to \mathbb{N} \in \mathcal{C}_c$.

g. $\mathrm{p} : \mathbb{N} \to \mathbb{N}, \quad \mathrm{p}(x) = x \dot{-} 1.$

Resolución

a. $\mathrm{id} \in \mathcal{C}_c$, pues $\mathrm{id} = \mathrm{u}_1^1$.

b. $\mathrm{uno}(x) = \mathrm{s}\big(\mathrm{n}(x)\big)$. Luego, $\mathrm{uno} \in \mathcal{C}_c$, pues sigue el esquema de composición con $f = \mathrm{s}$, $g_1 = \mathrm{n}$.

c. $\mathrm{s}_1^2(x, y) = \mathrm{s}\big(\mathrm{u}_1^2(x, y)\big)$. Sigue el esquema de composición con $f = \mathrm{s}$, $g_1 = \mathrm{u}_1^2$.

d. $\mathrm{flip}_h(x, y) = h\big(\mathrm{u}_2^2(x, y), \mathrm{u}_2^1(y, x)\big)$. Sigue el esquema de composición con $f = h$, $g_1 = \mathrm{u}_2^2$, $g_2 = \mathrm{u}_1^2$.

e. $h_i^n(x_1, \dots, x_n) = h\big(\mathrm{u}_i^n(x_1, \dots, x_n)\big)$. Sigue el esquema de composición con $f = h$, $g_1 = \mathrm{u}_i^n$.

f. $\mathrm{diag}_h = h\big(\mathrm{id}(x), \dots, \mathrm{id}(x)\big)$. Sigue el esquema de composición con $f = h$, $g_1 = \dots = g_k = \mathrm{id}$.

g. $\mathrm{p} \notin \mathcal{C}_c$. La demostración queda como ejercicio.[1]

Ejercicio 2

a. Considerando la clase $\mathcal{C}_c$ del ejercicio anterior, sea $h : \mathbb{N}^n \to \mathbb{N} \in \mathcal{C}_c$. Demostrar que existen $i \in \mathbb{N}$ y $\tilde{h} : \mathbb{N} \to \mathbb{N}$ tales que $h = \tilde{h} \circ \mathrm{u}_i^n$.

b. Dar un ejemplo de una función $h : \mathbb{N}^n \to \mathbb{N}$ que no pertenece a $\mathcal{C}_c$, diferente de la función p del ejercicio anterior.

Resolución

a. Para este ejercicio, utilizaremos por primera vez una herramienta que nos será muy útil en diferentes momentos de la materia. Se trata de la **inducción estructural**, una generalización del principio de inducción, que permite demostrar propiedades de los elementos de conjuntos que están definidos de *manera inductiva.*

   Un conjunto está definido de manera inductiva cuando todos sus elementos están especificados de alguna de las siguientes dos formas:

   - como parte de un conjunto de *elementos base* o *iniciales*, o
   - como el resultado de aplicar una *regla de formación* a elementos que ya se sabía que estaban en el conjunto.

   Para demostrar que cierta propiedad se cumple utilizando inducción estructural, verificamos que los elementos base cumplen la propiedad y que esta se preserva a través de las reglas de formación.

   En el caso de este ejercicio, la clase de funciones $\mathcal{C}_c$ está definida de manera inductiva; los elementos base son las funciones iniciales, y la única regla de formación posible es el esquema de composición. Es importante notar que no hay funciones en $\mathcal{C}_c$ que no entren en uno de estos dos casos, ya que es la clase *más chica* que contiene a las funciones iniciales y está cerrada por composición.

   Sea $h \in \mathcal{C}_c$; utilizando el principio de inducción estructural, separamos la demostración en casos.

   I. h es una función inicial. En este caso es trivial verificar que h cumple la propiedad:
      - si $h = \mathrm{n}$ o $h = \mathrm{s}$, tomamos $\tilde{h} = h$ e $i = 1$.
      - si $h = \mathrm{u}_j^n$, tomamos $\tilde{h} = \mathrm{id}$ e $i = j$.

   II. h se obtiene por composición a partir de $f, g_1, \dots, g_k \in \mathcal{C}_c$. Por hipótesis inductiva, asumimos que todas estas funciones cumplen la propiedad. Es decir:
      - existen $\tilde{f}$ y j tales que
      $$f = \tilde{f} \circ \mathrm{u}_j^n .$$

[1] Se puede usar la propiedad que se pide demostrar en el ejercicio 3 de la práctica 1.

- existen $\tilde{g}_1,\ldots,\tilde{g}_k$ e $i_1,\ldots,i_k$ tales que, para cada $r \in \{1,\ldots,k\}$, se cumple

$$g_r = \tilde{g}_r \circ \mathrm{u}^n_{i_r}.$$

En tal caso,

$$\begin{aligned}
h(x_1,\ldots,x_n) &= f\Big(g_1(x_1,\ldots,x_n),\ldots,g_k(x_1,\ldots,x_n)\Big)\\
&= \Big(\tilde{f}\circ \mathrm{u}^n_j\Big)\Big(g_1(x_1,\ldots,x_n),\ldots,g_k(x_1,\ldots,x_n)\Big)\\
&= \tilde{f}\Big(\mathrm{u}^n_j\big(g_1(x_1,\ldots,x_n),\ldots,g_k(x_1,\ldots,x_n)\big)\Big)\\
&= \tilde{f}\Big(g_j(x_1,\ldots,x_n)\Big)\\
&= \tilde{f}\Big(\big(\tilde{g}_j\circ \mathrm{u}^n_{i_j}\big)(x_1,\ldots,x_n)\Big)\\
&= \Big(\tilde{f}\circ\big(\tilde{g}_j\circ \mathrm{u}^n_{i_j}\big)\Big)(x_1,\ldots,x_n)\\
&= \Big(\big(\tilde{f}\circ\tilde{g}_j\big)\circ \mathrm{u}^n_{i_j}\Big)(x_1,\ldots,x_n)
\end{aligned}$$

Por lo tanto, tomando $\tilde{h} = \tilde{f}\circ\tilde{g}_j$ e $i = i_j$, verificamos que h también cumple con la propiedad.

b. La propiedad que demostramos en el inciso anterior quiere decir, intuitivamente, que las funciones de $\mathcal{C}_c$ solo dependen realmente de uno solo de sus parámetros, y "descartan" la información de todos los demás. Podemos pensar, entonces, en cualquier función que dependa realmente de sus dos parámetros, y demostrar que no pertenece a $\mathcal{C}_c$ verificando que no cumple la propiedad anterior.

   Tomemos, por ejemplo, la suma: $h(x,y) = x + y$. Veamos que no existen $\tilde{h}$ e i tales que $h = \tilde{h}\circ \mathrm{u}^2_i$. Lo demostraremos por el absurdo, suponiendo que sí existen y viendo que eso nos lleva a una contradicción.

   De existir tales $\tilde{h}$ e i, solo hay dos posibilidades: $i = 1$ o $i = 2$. Veamos los dos casos:

   I. Si $i = 1$, $h(x,y) = \tilde{h}\big(\mathrm{u}^2_1(x,y)\big) = \tilde{h}(x)$. Como $h(0,0) = 0$, debe ser que $\tilde{h}(0) = 0$. Pero como $h(0,1) = 1$, debe ser que $\tilde{h}(0) = 1$, lo cual es absurdo.

   II. El caso $i = 2$, análogo al anterior, queda como ejercicio.

   Como por ambos caminos se llega a una contradicción, no es correcto suponer que existen tales $\tilde{h}$ e i. Así, la suma no cumple la propiedad demostrada y, por lo tanto, no pertenece a la clase $\mathcal{C}_c$.

2. Recursión primitiva

Repaso

- Una función $h : \mathbb{N}^{n+1} \to \mathbb{N}$ se obtiene por **recursión primitiva** a partir de $f : \mathbb{N}^n \to \mathbb{N}$ y de $g : \mathbb{N}^{n+2} \to \mathbb{N}$ si se puede escribir como

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

  Nota: Si $n = 0$, el caso base se define por $h(0) = k \in \mathbb{N}$.

- Una función es **primitiva recursiva** (p.r.) si se puede obtener a partir de las funciones iniciales aplicando finitas veces composición y recursión primitiva.

- Una clase de funciones $\mathcal{C}$ es **PRC** si contiene a las funciones iniciales y está cerrada por composición y por recursión primitiva.

- Proposición. La clase de las funciones primitivas recursivas es la clase PRC más chica. Es decir, es en sí misma una clase PRC, y además es la intersección de todas las clases PRC.
  - Corolario. Una función es primitiva recursiva si y solo si está en toda clase PRC.

Ejercicio 3

Analizar el comportamiento de las siguientes funciones obtenidas por recursión primitiva.

a. $h_1 : \mathbb{N}^2 \to \mathbb{N}$, a partir de $f_1 = \mathrm{s}$, $g_1(x_1, x_2, x_3) = x_1 + x_2 + x_3$.

b. $h_2 : \mathbb{N}^3 \to \mathbb{N}$, a partir de $f_2(x_1, x_2) = x_1 + x_2$, $g_2(x_1, x_2, x_3, x_4) = x_1 \cdot x_2 \cdot x_3$.

c. $h_3 : \mathbb{N} \to \mathbb{N}$, a partir de $0 \in \mathbb{N}$, $g_3(x_1, x_2) = x_2$.

Resolución

a. Siguiendo el esquema de recursión primitiva,

$$\begin{aligned} h_1(x, 0) &= f_1(x) = \mathrm{s}(x) = x + 1, \\ h_1(x, t+1) &= g_1\big(h_1(x, t), x, t\big) = h_1(x, t) + x + t. \end{aligned}$$

Analizando los primeros casos:

$$\begin{aligned} h_1(x, 0) &= x + 1, \\ h_1(x, 1) &= h_1(x, 0) + x + 0 = 2x + 1, \\ h_1(x, 2) &= h_1(x, 1) + x + 1 = 3x + 1 + 1, \\ h_1(x, 3) &= h_1(x, 2) + x + 2 = 4x + 1 + 1 + 2, \\ &\vdots \end{aligned}$$

llegamos a la conclusión de que

$$h_1(x, y) = (y + 1)x + 1 + \sum_{i=1}^{y-1} i.$$

b. Análogamente al caso anterior,

$$\begin{aligned} h_2(x, y, 0) &= f_2(x, y) = x + y, \\ h_2(x, y, t+1) &= g_2\big(h_2(x, y, t), x, y, t\big) = h_2(x, t) \cdot x \cdot y. \end{aligned}$$

Analizando los primeros casos:

$$\begin{aligned} h_2(x, y, 0) &= x + y, \\ h_2(x, y, 1) &= h_2(x, y, 0) \cdot x \cdot y = (x + y) \cdot x \cdot y, \\ h_2(x, y, 2) &= h_2(x, y, 1) \cdot x \cdot y = (x + y) \cdot x^2 \cdot y^2, \\ h_2(x, y, 3) &= h_2(x, y, 2) \cdot x \cdot y = (x + y) \cdot x^3 \cdot y^3, \\ &\vdots \end{aligned}$$

llegamos a la conclusión de que

$$h_2(x, y, z) = (x + y) \cdot x^z \cdot y^z.$$

c. Esta vez,

$$\begin{aligned} h_3(0) &= 0, \\ h_3(t+1) &= g_3\big(h_3(t)\big) = t. \end{aligned}$$

Analizando con atención esta función, concluimos que se trata de la función p que no habíamos podido escribir cuando no contábamos con recursión primitiva:

$$h_3(x) = \mathrm{p}(x) = x \dot{-} 1.$$

Ejercicio 4

Sea $\mathcal{C}$ una clase PRC cualquiera. Demostrar que las siguientes funciones están en $\mathcal{C}$.

a. $h_1 : \mathbb{N}^2 \to \mathbb{N}, \quad h_1(x, y) = x \dot{-} y.$

b. $h_2 : \mathbb{N} \to \mathbb{N}, \quad h_2(x) = x!$

c. $h^{(\bullet)} : \mathbb{N}^2 \to \mathbb{N}, \quad h^{(n)}(x) = (\underbrace{h \circ \cdots \circ h}_{n \text{ veces}})(x),$ sabiendo que $h : \mathbb{N} \to \mathbb{N} \in \mathcal{C}$.

¿Cuáles de estas funciones son primitivas recursivas?

Resolución

a. Intentaremos escribir h siguiendo el esquema de recursión primitiva.

- El caso base es sencillo; queremos que se cumpla

  $$h_1(x, 0) = x \dot{-} 0 = x = \mathrm{id}(x).$$

  Como sabemos que $\mathrm{id} \in \mathcal{C}$, ya que los proyectores están en toda clase PRC, podemos tomar $f_1 = \mathrm{id}$ como caso base del esquema de recursión primitiva.

- Para el caso recursivo, queremos que se cumpla

  $$h_1(x, t+1) = x \dot{-} (t+1).$$

  Podemos reescribir esto utilizando el valor de $h_1(x, t) = x \dot{-} t$, que asumimos conocido; nos queda:

  $$\begin{aligned} h_1(x, t+1) &= (x \dot{-} t) \dot{-} 1 \\ &= \mathrm{p}(x \dot{-} t) \\ &= \mathrm{p}\big(h_1(x, t)\big) \end{aligned}$$

  Debemos escribir esta última expresión como una función $g_1 : \mathbb{N}^3 \to \mathbb{N}$, que tome como argumentos $h_1(x, t)$, x y t, y que podamos verificar que está en $\mathcal{C}$. En esta definición podemos usar la función p, ya que del ejercicio anterior se sigue trivialmente que está en cualquier clase PRC.[2]
  Como queremos que se cumpla

  $$g_1(h_1(x, t), x, t) = \mathrm{p}\big(h_1(x, t)\big),$$

  podemos definir:

  $$\begin{aligned} g_1(x_1, x_2, x_3) &= \mathrm{p}(x_1) \\ &= \mathrm{p}\big(u_1^3(x_1, x_2, x_3)\big), \end{aligned}$$

  que claramente pertenece a $\mathcal{C}$ por ser una composición de funciones de dicha clase.

[2] Convencerse de esto queda como ejercicio.

Como h_1 se obtiene aplicando el esquema de recursión primitiva sobre las funciones f_1 y g_1 recién definidas, y ambas pertenecen a $\mathcal{C}$, podemos afirmar que h_1 pertenece a $\mathcal{C}$.

b. Escribimos h_2 según el esquema de recursión primitiva a partir de las funciones f_2 y $g_2 \in \mathcal{C}$ que definimos a continuación.

- Caso base: como h_1 solo tiene un parámetro, el caso base se define como una constante natural. De acuerdo con la definición de la función factorial, tomamos $h_2(0) = 1$.
- Caso recursivo: queremos que
$$\begin{aligned} h_2(t+1) &= (t+1)! \\ &= (t+1)\cdot t! \\ &= (t+1)\cdot h_2(t), \end{aligned}$$
es decir,
$$g_2(h_2(t), t) = (t+1)\cdot h_2(t).$$
Por lo tanto, definimos
$$\begin{aligned} g_2(x_1, x_2) &= (x_2 + 1)\cdot x_1 \\ &= \mathrm{s}(x_2)\cdot x_1 \\ &= \mathrm{s}\Big(\mathrm{u}_2^2(x_1, x_2)\Big)\cdot \mathrm{u}_1^2(x_1, x_2). \end{aligned}$$
Asumiendo que el producto está en $\mathcal{C}$ (al resolver el ejercicio 2 de la práctica 1, se demuestra que está en toda clase PRC), podemos afirmar que $g_2 \in \mathcal{C}$.

c. Es muy importante no caer en el error de afirmar que
$$h^{(n)}(x) = (\underbrace{h \circ \cdots \circ h}_{n \text{ veces}})(x)$$
es una escritura correcta de $h^{(\bullet)}$ según el esquema de composición. Esto solo sería válido si la cantidad de apariciones de n estuviera fija, pero no es así: se trata de uno de los parámetros de la función.[3]

Lo que sí podemos hacer es escribir $h^{(\bullet)}$ siguiendo el esquema de recursión primitiva, a partir de dos funciones f_h y $g_h \in \mathcal{C}$ que definiremos a continuación. Para ajustarnos mejor al esquema, escribiremos
$$h'(x, n) = h^{(n)}(x),$$
con lo cual estará claro que la recursión es sobre el último parámetro.[4]

- Caso base: queremos que $h'(x, 0) = x = \mathrm{id}(x)$; por lo tanto, podemos tomar $f_h = \mathrm{id} \in \mathcal{C}$.
- Caso recursivo: queremos que
$$\begin{aligned} h'(x, t+1) &= (\underbrace{h \circ \cdots \circ h}_{t+1 \text{ veces}})(x) \\ &= h\Big((\underbrace{h \circ \cdots \circ h}_{t \text{ veces}})(x)\Big) \\ &= h\big(h'(x, t)\big), \end{aligned}$$
es decir,
$$g_h(h'(x, t), x, t) = h(h'(x, t)).$$

[3]¡Entender por qué esto es cierto es fundamental para poder avanzar en la materia! Si no estás del todo convencido o convencida, te recomiendo que lo consultes y vuelvas a razonarlo hasta que ya no queden dudas.

[4]En realidad, el orden de los parámetros no es relevante para determinar la pertenencia de una función a una clase PRC, ya que, como se vio en el primer ejercicio, se puede alterar fácilmente el orden de los parámetros componiendo la función con los proyectores adecuados.

Por lo tanto, definimos

$$\begin{aligned} g_h(x_1, x_2, x_3) &= h(x_1) \\ &= h\big(\mathrm{u}_1^3(x_1, x_2, x_3)\big), \end{aligned}$$

que claramente pertenece a $\in \mathcal{C}$, por ser la composición de dos funciones que están en dicha clase.

Por último, como respuesta a la pregunta final, todas las funciones de este ejercicio son primitivas recursivas: como hemos demostrado que pertenecen a una clase PRC arbitraria, podemos afirmar que pertenecen a toda clase PRC; por lo tanto, pertenecen a la intersección de todas las clases PRC, que es la clase de las funciones primitivas recursivas.