Utilización de listas y pares en funciones PRC

Herman Schinca (con modificaciones menores de Rodrigo Castaño)

19 de agosto de 2016

Ejercicio 1. *Mostrar que fib, la función de Fibonacci, está en toda clase PRC, donde:*

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

Resolución. Luego de razonar un rato concluimos que la recursión que utiliza Fibonacci, basada en 2 pasos recursivos, no matchea con el esquema de recursión primitiva que podemos utilizar. Recordamos entonces la existencia de los pares, su codificación y que la misma resulta biyectiva. Además, sabemos que tanto los constructores como los observadores de pares son p.r.

La idea es construir una función fib' definida como:

$$fib'(n) = \langle fib(n),\ fib(n+1) \rangle$$

Veamos si fib' es primitiva recursiva. Analicemos primero el valor de la función en 0.

$$fib'(0) = \langle fib(0),\ fib(1) \rangle = \langle 0, 1 \rangle$$

Es decir que $fib'(0)$ está definida como una constante, lo cual va bien con la definición de recursión primitiva.

Veamos ahora cómo luce fib' para el caso $t+1$.

$$\begin{aligned} fib'(t+1) &= \langle fib(t+1),\ fib(t+2) \rangle \\ &= \langle fib(t+1),\ fib(t+1) + fib(t) \rangle \\ &= \langle r(fib'(t)),\ r(fib'(t)) + l(fib'(t)) \rangle \end{aligned}$$

Por lo tanto el caso recursivo depende únicamente del valor de la función en el punto anterior, haciendo que fib' use recursión primitiva.

Observar que demostramos que fib' hace uso de recursión primitiva a partir de analizar casos particulares usando su definición. Si hubiésemos dado la definición por casos de manera directa (es decir, sin derivarla de su definición) habría que demostrar por inducción que la misma calcula lo que decimos que calcula.

Resta ver cómo obtener fib a partir de fib', pero esto es sencillo: simplemente proyectamos la primer componente del par.

$$fib(n) = l(fib'(n))$$

Como fib' y la proyección l de pares están en toda clase PRC, entonces fib también lo está.

□

Ejercicio 2. *Mostrar que $f : \mathbb{N} \to \mathbb{N}$ es primitiva recursiva, donde f está definida de la siguiente manera:*

$f(0) = 0$

Para todo $x \neq 0$, $f(x)$ es el único número que cumple los siguiente (utilizando la codificación de listas[1] vista en la teórica y las funciones $|\cdot|$, $\cdot[\cdot]$ asociadas a la misma):

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

1. $|f(x)| = |x|$

2. $\forall i \in \{1, \ldots, |x|\}\ .\ par(i) \Rightarrow f(x)[i] = x[i]^2$

3. $\forall i \in \{1, \ldots, |x|\}\ .\ impar(i) \Rightarrow f(x)[i] = x[i]$

Ejemplo: dado $x = [1, 2, 3, 4, 5, 6]$*,* $f(x) = [1, 4, 3, 16, 5, 36]$*.*

Resolución. Analizando la descripción de $f(x)$, vemos que lo que hace la función es, dado un natural x, si es 0 devuelve 0 y si no lo interpreta como una lista sin ceros al final (hay una única forma de hacer esto por ej 13a) y la modifica elevando al cuadrado las posiciones pares, luego calcula su codificación y devuelve ese número.

Entonces es razonable pensar que f es:

$$f(x) = \begin{cases} \prod_{i=1}^{|x|} nprimo(i)^{x[i]^{2^{\alpha(impar(i))}}} & \text{si } x > 0 \\ 0 & \text{si no} \end{cases}$$

Donde $\alpha(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{si no} \end{cases}$

Luego, como x^y (ej 2), $impar(y)$ (par: ej 6), $nprimo(x)$ (ej 8), $\cdot[\cdot]$ (ej 13), $|\cdot|$ (ej 13), α (teórica) y $\prod_{i=1}^{n} f'$ con f' en una clase PRC (teórica) son todas p. r., $f(x)$, por composición, es p. r.

Siendo que $f(x)$ es p.r., resta ver que cumpla las propiedades del enunciado.

El caso $f(0)$ es trivialmente correcto.

Consideremos el otro caso y las propiedades correspondientes:

1. $|\prod_{i=1}^{|x|} nprimo(i)^{x[i]^{2^{\alpha(impar(i))}}}|$ no puede ser mayor que $|x|$ por definición de $|x|$.

   Por otra parte, para ser menor debería pasar que $x[|x|]^{2^{\alpha(impar(i))}} = 0$, pero $x[|x|]$ justamente no es 0 por definición de longitud.

2,3 $\prod_{i=1}^{|x|} nprimo(i)^{x[i]^{2^{\alpha(impar(i))}}}[i] = x[i]^{2^{\alpha(impar(i))}} = \begin{cases} x[i]^2 & par(i) \\ x[i] & impar(i) \end{cases}$

□

Ejercicio 3. *Sea $\mathcal{C}$ la mínima clase de funciones que contiene a la función nula $n(x) = 0$, todas las proyecciones y es cerrada por composición y por recursión primitiva y sea $k \in \mathbb{N}$. Demostrar que para toda función $f : \mathbb{N}^n \to \mathbb{N}$ de $\mathcal{C}$ se cumple que $\forall\ x_1, \ldots, x_n \in \mathbb{N}$, si $x_i \leq k\ \forall\ i = 1, \ldots, n$, entonces $f(x_1, \ldots, x_n) \leq k$.*

Resolución. Analicemos qué se pide que demostremos. Bajo la hipótesis enunciada, que los parámetros de f son siempre menores o iguales que k, y siendo que no incluímos explícitamente en $\mathcal{C}$ a la funcion $s(x) = x + 1$, no será posible que el valor de f sea mayor que k.

Vamor a usar inducción estructural para ver que $f(x_1, \ldots, x_n) \leq k$ siempre que se cumpla la hipótesis, que $x_i \leq k\ \forall i = 1, \ldots, n$.

Caso $f(x) = n(x) = 0 \leq k$ para cualquier x. Queda trivialmente demostrado.

Caso $f(\vec{x}) = u_i^n(\vec{x}) = x_i \leq k$ por hipótesis.

Caso $f(x_1, \ldots, x_n) = f'(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ donde tanto f' como $g_1, \ldots, g_m$ son de $\mathcal{C}$. Por hipótesis inductiva, si $x_i \leq k\ \forall\ i = 1, \ldots, n$, entonces $g_j(x_1, \ldots, x_n) \leq k$ para todo j en $1, \ldots, m$.

Nuevamente, por hipótesis inductiva, $f'(x'_1, \ldots, x'_m) \leq k$ siempre que $x'_i \leq k\ \forall i = 1, \ldots, m$. Esto último se cumple, ya que $x'_i = g_i(x_1, \ldots, x_n) \leq k$.

Caso h (para no colisionar el nombre con el usado en recursión primitiva) se obtiene por recursión primitiva a partir de f y g, ambas de $\mathcal{C}$ (y cumplen la propiedad a demostrar), de la siguiente manera:

$h(x_1, \ldots, x_{n-1}, 0) = f(x_1, \ldots, x_{n-1})$
$h(x_1, \ldots, x_{n-1}, t+1) = g(h(x_1, \ldots, x_{n-1}, t), x_1, \ldots, x_{n-1}, t)$

Para este caso vamos a hacer inducción en t (adicional a la inducción estructural).

Si $t = 0$, estamos en el caso de la primera ecuación, como para f vale la hipótesis inductiva (de inducción estructural), se cumple la propiedad a demostrar.

En la segunda ecuación, de nuevo, g cumple la propiedad a demostrar, por lo que, si vale que $x_i \leq k \ \forall i = 1, \ldots, n-1$, $t \leq k$, y $h(x_1, \ldots, x_{n-1}, t) \leq k$, entonces necesariamente $g(h(x_1, \ldots, x_{n-1}, t), x_1, \ldots, x_{n-1}, t) \leq k$.

De no valer lo primero o lo segundo ($x_i \leq k \ \forall i = 1, \ldots, n-1$ y $t \leq k$)[2], no se cumple el antecedente de la implicación que queremos probar, por lo que la propiedad se cumple.

Lo último ($h(x_1, \ldots, x_{n-1}, t) \leq k$) se cumple por hipótesis inductiva (esta vez inducción en t).

$\square$

Ejercicio 4. *Sean $k > 0 \in \mathbb{N}$, $f_0, \ldots, f_{k-1} : \mathbb{N} \to \mathbb{N}$, $g : \mathbb{N}^{k+2} \to \mathbb{N}$ y $h : \mathbb{N}^2 \to \mathbb{N}$ definida como*

$$h(x,t) = \begin{cases} f_0(x) & si\ t = 0 \\ \vdots & \\ f_{k-1}(x) & si\ t = k-1 \\ g(t, h(x, t-1), \ldots, h(x, t-k), x) & si\ t \geq k \end{cases}$$

Demostrar que si $f_0 \ldots f_{k-1}$ y g pertenecen a una clase PRC $\mathcal{C}$, entonces también h pertence a $\mathcal{C}$.

Resolución. Intuitivamente, la recursión parece similar al caso de Fibonacci. Vamos a buscar formular una función h' que devuelva una k-tupla. Con h' vamos a probar que la misma se adapta al esquema de recursión primitiva.

Tomemos $h'(x,t) = \langle h(x, t+k-1), \ldots, h(x,t) \rangle$.

Vamos a usar las funciones p_i tales que $p_i(\langle x_1, \ldots, x_n \rangle) = x_i \ \forall i = 1..n$ (por ej 12 existen tanto los observadores p_i como las funciones codificadoras $\langle \cdot \rangle_n$, y las mismas son PRC).

En este caso $h'(x, 0) = \langle f_{k-1}(x), \ldots, f_0(x) \rangle$ por definición de h para $t < k$, que es de $\mathcal{C}$ ya que se obtiene por composicion de $\langle \cdot \rangle_n$ y de $f_0, \ldots, f_{k-1}$. Por el enunciado, $f_0, \ldots, f_{k-1}$ son de $\mathcal{C}$ y $\langle \cdot \rangle_n$ es de $\mathcal{C}$ por ser p.r..

Retomando el otro caso:

$$\begin{aligned} h'(x, t+1) &= \langle h(x, t+k), h(x, t+k-1) \ldots, h(x, t+1) \rangle \\ &= \langle g(t, h(x, t+k-1), \ldots, h(x,t), x), h(x, t+k-1), \ldots, h(x, t+1) \rangle \\ &= \langle g(t, h(x, t+k-1), \ldots, h(x,t), x), p_1(h'(x,t)), \ldots, p_{k-1}(h'(x,t)) \rangle \\ &= \langle g(t, p_1(h'(x,t)), \ldots, p_{k-1}(h'(x,t)), x), p_1(h'(x,t)), \ldots, p_{k-1}(h'(x,t)) \rangle \end{aligned}$$

Como el caso recursivo de h' se obtiene por composición a partir de funciones p.r. utilizando únicamente $h'(x,t)$, x y t, este caso es una función p.r. y se adapta al esquema de recursión primitiva.

Habiendo demostrado que h' es de $\mathcal{C}$, el último paso es demostrar que h es también de $\mathcal{C}$.

Usamos el resultado demostrado en el ejercicio 5:

Las hipótesis del mismo con respecto a los predicados se cumplen, ya que los mismos son disjuntos y cada uno de ellos es de $\mathcal{C}$ (falta demostrarlo). Con respecto a las funciones, para todos los casos excepto el último, por el enunciado sabemos que son de $\mathcal{C}$.

Como demostramos que h' también está en $\mathcal{C}$, entonces reformulamos el último caso como $p_1(h'(x, t \dot{-} k))$ que se obtiene por composición a partir de h', $\dot{-}$ y p_1, ambas en $\mathcal{C}$ (para ser rigurosos deberíamos mostrar que encaje en el esquema de composición).

[2] Por hipótesis de nuestra propiedad tenemos que $t+1 \leq k$, pero $t < t+1$, por lo que $t \leq k$.

Lema 1. *(ejercicio 5 de la práctica) 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 $(a_1, \ldots, a_n) \in \mathbb{N}^n$). También está en $\mathcal{C}$ cualquier función h que cumpla:*

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

Observar que h queda completamente determinada por este esquema.

□

Ejercicio 5. *Demostrar que la función mcd(x, y) que calcula el máximo común divisor está en cualquier clase $\mathcal{C}$ PRC.*

Resolución. Sea $d = mcd(x, y)$, se cumple todo lo siguiente:

- $divide(d, x)$
- $divide(d, y)$
- $\forall i \in \mathbb{N}\ (divide(i, x) \cdot divide(i, y)) \Rightarrow i \leq d$

Podemos hacer una maximización acotada, ya que sabemos que el mcd va a estar entre 1 y el mínimo de los dos números. ¿Podemos también acotar el cuantificador universal? Sí, en particular podemos usar la misma cota, ya que cualquier número mayor que x o y necesariamente no podrá dividir a ambos. Entonces maximizamos de la siguiente manera:

$mcd(x, y) = \text{máx}_{d \leq x}\, divide(d, x) \cdot divide(d, y) \cdot (\forall i)_{\leq x}(divide(i, x) \cdot divide(i, y)) \Rightarrow i \leq d$ □