Resueltos Lógica y Computabilidad

Ignacio E. Losiggio

February 18, 2019

1 Práctica 3 — Funciones no-computables y conjuntos c.e.

1.1 Probar, usando una diagonalización, que las siguientes funciones no son computables:

$$f_1(x,y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y)\downarrow \\ 0 & \text{en otro caso} \end{cases} \qquad f_3(x,y,z) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y)\downarrow \text{ y } \Phi_x^{(1)}(y) > z \\ 0 & \text{en otro caso} \end{cases}$$

$$f_2(x,y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y) = 0 \\ 0 & \text{en otro caso} \end{cases} \qquad f_4(x) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(x)\downarrow \text{ y } \Phi_x^{(1)}(x) \neq x \\ 0 & \text{en otro caso} \end{cases}$$

1.1.1 $f_1(x,y)$

Si f_1 es computable entonces puedo construir el siguiente programa que lo la use a cuyo número llamaremos e:

$$[A] \text{ IF } f_1(X_1, X_1) \neq 0 \text{ GOTO } A$$

Luego intentemos determinar el valor de $f_1(e,e)$:

$$f_1(e,e) = 1 \iff \Phi_e^{(1)}(e)\downarrow \iff f_1(e,e) = 0$$

1.1.2 $f_2(x,y)$

Armemos de vuelta una función que sea molesta:

$$\begin{aligned} &[A] \text{ IF } f_2(X_1, X_1) = 0 \text{ GOTO } A \\ &\quad Y \leftarrow Y + 1 \end{aligned}$$

E intentemos determinar el valor de $f_2(e,e)$ otra vez:

$$f_2(e,e) = 1 \iff \Phi_e^{(1)}(e) = 0$$
$$f_2(e,e) = 0 \iff \Phi_e^{(1)}(e) \neq 0 \vee \Phi_e^{(1)}(e) \uparrow \iff f_2(e,e) \neq 0 \vee f_2(e,e) \uparrow$$

1.1.3 $f_3(x,y,z)$

Cómo venimos haciendo empezamos por el "programa rebelde":

$[A]$ IF $f_3(X_1, X_1, X_1) \neq 0$ GOTO A
$Y \leftarrow X_1 + 1$

Y intentamos ver que determinar el valor de $f_3(e,e,e)$ no tiene sentido:

$$\begin{aligned} f_3(e,e,e) = 1 &\iff \Phi_e^{(1)}(e) \downarrow \wedge\ \Phi_e^{(1)}(e) > e \\ &\iff f_3(e,e,e) = 0 \ \wedge\ f_3(e,e,e) = 0 \\ &\iff f_3(e,e,e) = 0 \end{aligned}$$

1.1.4 $f_x(x)$

$[A]$ IF $f_4(X_1) \neq 0$ GOTO A
$Y \leftarrow X_1 + 1$

$$\begin{aligned} f_4(e) = 1 &\iff \Phi_e^{(1)}(e) \downarrow \wedge\ \Phi_e^{(1)}(e) \neq e \\ &\iff f_4(e) = 0 \ \wedge\ f_4(e) = 0 \iff f_4(e) = 0 \end{aligned}$$

1.2 Probar, reduciendo a cualquier función del ejercicio 1, que las siguiente funciones no son computables:

$$g_1(x,y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y) \uparrow \\ 0 & \text{en otro caso} \end{cases}$$

$$g_2(x,y,z,w) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(z) \downarrow \text{ y } \Phi_y^{(1)}(w) \downarrow \text{ y } \Phi_x^{(1)}(z) > \Phi_y^{(1)}(w) \\ 0 & \text{en otro caso} \end{cases}$$

$$g_3(x,y,z) = \begin{cases} z+1 & \text{si } \Phi_x^{(1)}(y) \downarrow \text{ y } \Phi_x^{(1)}(y) \neq z \\ 0 & \text{en otro caso} \end{cases}$$

$$g_4(x,y,z) = \begin{cases} (\Phi_x^{(1)} \circ \Phi_y^{(1)})(z) & \text{si } \Phi_y^{(1)}(z) \downarrow \text{ y } (\Phi_x^{(1)} \circ \Phi_y^{(1)})(z) \downarrow \\ 0 & \text{en otro caso} \end{cases}$$

1.2.1 $g_1(x, y)$

$$\alpha \circ g_1 = f_1$$

1.2.2 $g_2(x, y, z, w)$

Sea e el número de un programa que computa la función identidad podemos decir:

$$g_2(x, e, y, z) = f_2(x, y, z)$$

1.2.3 $g_3(x, y, z)$

$$\beta(g_3(x, x, x)) = f_4(x)$$

1.2.4 $g_4(x, y, z)$

Sea (otra vez) e el número de un program que computa la función identidad podemos decir:

$$g_4(x, e, y) = f_1(x, y)$$

1.3 Probar que la siguiente función no es computable reduciendo la función f_4 del primer ejercicio

$$g_3'(x, y, z) = \begin{cases} z & \text{si } \Phi_x^{(1)}(y) \downarrow \text{ y } \Phi_x^{(1)}(y) \neq z \\ 0 & \text{en otro caso} \end{cases}$$

Sugerencia: Revisar que la reducción maneje correctamente el caso $f_4(0)$.

No sé si esto está bien pero yo supondría que:

$$\beta(g_3'(x, x, x)) = f_4(x)$$

Dado que 0 es el programia vacío (una implementación de $n(x)$) entonces que $g_4(0, 0, 0) = 0$ se corresponde con el valor de $f_4(0)$. En el resto de los valores de x es evidente que ambas funciones hacen lo mismo.

1.4 Decimos que una función parcial computable f es *extensible* si existe g computable tal que $g(x) = f(x)$ para todo $x \in \text{Dom } f$. Probar que existe una función parcial computable que no es extensible (*Sugerencia*: considerar una función tal que con su extensión se podría computar alguna variante del halting problem).

Vamos a tomar la siguiente f que suponemos extensible:

$$f(x) = \begin{cases} \Phi_x^{(1)}(x) & \text{si } \Phi_x^{(1)}(x)\downarrow \\ \uparrow & \text{sino} \end{cases}$$

Cómo es extensible entonces existe un g que puede calcular resultados para todo su dominio y devuelve "basura" para cualquier otro valor. Dado que g es computable podemos definir el siguiente programa y tomar su número e:

$$\begin{aligned} &[A] \text{ IF } g(X_1) \neq X_1 \text{ GOTO } A \\ &\quad\ Y \leftarrow X_1 \end{aligned}$$

Ahora, ¿Cuál es el valor de $g(e)$?

$$g(e) = e \implies g(e) \neq e$$
$$g(e) \neq e \implies g(e) = e$$