Práctica 8: Funciones no computables y diagonalización

Lenguajes Formales, Autómatas y Computabilidad, FCEN-UBA

Primer Cuatrimestre 2025

Ejercicio 1. Demostrar que $Halt : \mathbb{N} \to \mathbb{N}$ no es computable.

Ejercicio 2. Sea $f : \mathbb{N}^3 \to \mathbb{N}$ tal que

$$f(x, y, z) = min_{t \leq z}(Halt(t, y) \wedge t > x)$$

Decidir si la función es computable o no y demostrar. Analizar qué sucede si se cambia el mínimo por un para todo o por un existencial (en ambos casos con la misma cota que el mínimo).

Ejercicio 3. Sea $f : \mathbb{N}^3 \to \mathbb{N}$ tal que

$$f(x, y, z) = \forall_{t \leq z+1}(Halt(x + y, x + y) \wedge t > z)$$

Decidir si la función es computable o no y demostrar. Analizar qué sucede si se cambia el para todo por un mínimo o por un existencial (en ambos casos con la misma cota que el para todo).

Ejercicio 4. Demostrar que existen más funciones $f : \mathbb{N} \to \mathbb{N}$ que números naturales (es decir, que no existe ninguna función $g : \mathbb{N} \to \{f : \mathbb{N} \to \mathbb{N}\}$ sobreyectiva).

Ejercicio 5. Probar usando diagonalización, que las siguientes funciones no son computables:

$$f_1(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y)\downarrow \\ 0 & \text{en otro caso} \end{cases} \qquad f_2(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y) = 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$f_3(x, y, z) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y)\downarrow \text{ y } \Phi_x^{(1)}(y) > z \\ 0 & \text{en otro caso} \end{cases} \qquad f_4(x) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(x)\downarrow \text{ y } \Phi_x^{(1)}(x) \neq x \\ 0 & \text{en otro caso} \end{cases}$$

Ejercicio 6. Probar, reduciendo cualquier función del ejercicio anterior, que las siguientes funciones no son computables:

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

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

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

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

Ejercicio 7. Probar que la siguiente función no es computable reduciendo la función f_4 del Ej. 5 .

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

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

Ejercicio 8. Decimos que una función parcial computable f es *extensible* si existe g computable tal que $g(x) = f(x)$ para todo $x \in \operatorname{Dom} f$. 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).

Ejercicio 9. Considerar la función $BB : \mathbb{N} \to \mathbb{N}$ que cumple que $BB(n)$ devuelve la cantidad de pasos de ejecución que realiza el programa de tamaño n que más pasos realiza antes de terminar, empezando con input 0. Es decir, dado n, BB devuelve, de entre todos los programas de tamaño n **que terminan con input 0**, cuantos pasos realiza aquel que más pasos realiza antes de terminar con input 0. Demostrar que BB no es computable (*Sugerencia:* entender bien primero qué devuelve la función, y luego pensar como podría usarse BB para computar $HALT$ si fuera computable).

Ejercicio 10. Para cada $n \in \mathbb{N}$ definimos una función total $g_n : \mathbb{N} \to \mathbb{N}$ de la siguiente manera:

$$g_n(x) = \begin{cases} t+1 & \text{si } |x+1| = n \text{ y } \Phi_x(x)\downarrow \text{ en exactamente } t \text{ pasos} \\ 0 & \text{si no} \end{cases}$$

a) Siendo $g(n,x) = g_n(x)$, ¿es $g : \mathbb{N}^2 \to \mathbb{N}$ una función computable?

b) ¿Existe algún $n \in \mathbb{N}$ tal que g_n resulta computable?

Ejercicio 11. Sea $f_0, f_1, f_2, f_3, \ldots$ un orden de las funciones totales computables unarias.

a) ¿Existe una función total computable binaria g tal que $g(n,x) = f_n(x)$ para todo $n, x \in \mathbb{N}$?

b) Sea $k \in \mathbb{N}$ fijo. Demuestre que existe una función total computable binaria g tal que $g(n,x) = f_n(x)$ para todo $0 \leq n \leq k$ y $x \in \mathbb{N}$.

Ejercicio 12. Sea $a : \mathbb{N} \to \mathbb{N}$ una función total tal que, para toda función $f : \mathbb{N} \to \mathbb{N}$ total computable., existe un $e_f \in \mathbb{N}$ tal que $a(y) \geq f(y)$ para todo $y > e_f$.

a) Decidir si a es computable o no, y demostrarlo.

b) Decidir y demostrar si la siguiente afirmación es verdadera o no:
'Si $g : \mathbb{N} \to \mathbb{N}$ es una función total que cumple que existe un n_0 y una función total computable $f : \mathbb{N} \to \mathbb{N}$ tal que $g(y) \leq f(y)$ para todo $n_0 \geq e$, entonces g es computable.

Ejercicio 13. Decidir si las siguientes afirmaciones son verdaderas o falsas, y demostrar.

a. Si $f : \mathbb{N}^2 \to \mathbb{N}$ es una función que cumple que, para todo $n \in \mathbb{N}$, la función $g_n(x) = f(n,x)$ es total y computable, entonces f es computable (total).

b. Sea e un número fijo, la siguiente función no es computable:

$$f(x) = \begin{cases} x+1 & \text{si } \mathrm{Halt}(e,e) \\ 0 & \text{en otro caso} \end{cases}$$