1. Diagonalización y reducciones

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

$$Halt(x) = \begin{cases} 1 & si\ \Phi_x(x)\downarrow \\ 0 & sino \end{cases}$$

Demo: Supongamos que es Computable

Sea P tal que $\Psi_P^{(1)}(x) = Halt(x)$. (Computa Halt)

Construimos una funcion $f(x) = \begin{cases} 0 & si\ Halt(x) = 0 \\ \uparrow & si\ no \end{cases}$ ("Niega su Comportamiento")

Sea Q el Programa que Computa $f(x)$ While $\Psi_P^{(1)}(x_1) \neq 0$ do{} ∴ F es Computable xq halt lo es.

Sea e / $\Psi_Q^{(1)} = \Phi_e^{(1)}$.

$\Phi_e(e)\downarrow \leftrightarrow \Psi_P(e) = 0 \leftrightarrow \Phi_e(e)\uparrow$ ABS, vino de Suponer Halt es Computable

Ejercicio 2. 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).

Supongamos que existe $g : \mathbb{N} \to A$, $A = \{f / f : \mathbb{N} \to \mathbb{N}\}$.

Entonces podemos $g(0) = f_0$, $g(1) = f_1$, $g(2) = f_2$, $\vdots$, $g(n) = f(n)$

y Defino $D : \mathbb{N} \to \mathbb{N}$ / $D(n) = f_n(n) + 1$. Entonces $D \in A$ xque es $\mathbb{N} \to \mathbb{N}$

Como g es Sobreyectiva tiene que existir ∴ $\exists\, k \in \mathbb{N}$ / $D = f_k$

∴ $g(k) = f_k = D$ Entonces $f_k(k) = g(k) =$

$D(k) = f_k(k) + 1$

Ejercicio 3. Probar usando diagonalización, que las siguientes funciones no son computables:

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

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

Halt(x,y)

1) Supongamos que f_1 es Computable, entonces puedo definir Un programa P / $\Psi_P^{(1)}(x) = g(x)$ que computa la función con $g(x) = f(x,x) = \begin{cases} 1 & si\ \Phi_x^1(x)\downarrow \\ 0 & sino \end{cases}$ (Ej 1)

Con g, que Es parcial computable xq f lo es, armo $h(x) = \begin{cases} 0 & si\ g(x) = 0 \\ 1 & si\ g(x) = 1 \end{cases}$

h es parcial Computable xq g lo es. Sea e el numero de Programa que Computa $h(x)$

$\Phi_a(a) = 0 \leftrightarrow \Psi_P^{(1)}(x) = 0 \leftrightarrow \Phi_a(a)\uparrow$ ABS! Vino de Suponer f computable

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

Suponemos f computable, podemos Armar $g(x) = f(x,x) = \begin{cases} 1 & si\ \Phi_x(x) = 0 \\ 0 & otro\ caso \end{cases}$, que es Computable si f lo es.

```
if( Φx(x) = 0 )then{
    y := 1 }
Else{ y := 0 }
```

Sea $g'(x) = \begin{cases} 0 & si\ g(x) = 1 \\ 1 & si\ g(x) = 0 \end{cases}$, computable porque g es computable

Sea e / $\Phi_e(x) = g'(x)$, $\Phi_e(e) = 1 \leftrightarrow g(e) = 1 \leftrightarrow \Phi_e(e) = 0$ ABS! de Suponer f computable

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

Asumo f computable

Sea $g(x) = f(x,x,0) = \begin{cases} 1 & \text{si } \Phi_x(x)\downarrow \wedge \Phi_x(x) > 0 \\ 0 & \text{otro caso} \end{cases}$, que si f es computable, g lo es

Sea $g'(x) = \begin{cases} 1 & \text{si } g(x) = 0 \\ 0 & \text{si } g(x) = 1 \end{cases}$, que si g es computable, g' lo es también.

Sea e/ $\Phi_e(x) = g'(x)$.

$\Phi_e(e) = 0 \leftrightarrow g(e) = 1 \leftrightarrow \Phi_e(e)\downarrow \wedge \Phi_e(e) > 0$ ABS

ó $\Phi_e(e) = 1 \leftrightarrow g(e) = 0 \leftrightarrow \neg(\Phi_e(e)\downarrow \wedge \Phi_e(e) > 0)$

$\neg\Phi_e(e)\downarrow \vee \neg\Phi_e(e) > 0$ ABS

$\Phi_e(e)\uparrow \vee \Phi_e(e) \leq 0$

x cualquiera de las 2 viene de suponer f computable

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

Asumo f computable

Sea $f'(x) = \begin{cases} x+1 & \text{si } f(x) = 0 \\ x & \text{si } f(x) = 1 \end{cases}$, que si f es computable, f' lo es.

Sea e/ $\Phi_e^{(1)}(x) = f'(x)$

$\Phi_e(e) = e \leftrightarrow f(x) = 1 \leftrightarrow \Phi_e(e)\downarrow \wedge \Phi_e(e) \neq e$

ó

$[\Phi_e(e) = e+1 \leftrightarrow f(x) = 0 \leftrightarrow \Phi_e(e)\uparrow \vee \Phi_e(e) = e]$ ABS

Ejercicio 4. Probar, reduciendo cualquier función del ejercicio anterior, que las siguientes funciones no son computables:

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

Supongo que g es computable

Entonces $\exists P / \Psi_P^{(2)}(x,y) = g(x,y)$

Sea PF1 el programa / $\Psi_{PF1} = f_1(x,y)$

PF1: P

if Y = 0 then

Y := 1

Else

Y := 0

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

Construimos un PF1 / $\Psi^{(2)}_{PF1}(x,y) = f_1(x,y)$

Pero $f_1(x,y)$ no es computable, si dimos

. Un programa que lo es. ¿¿ABS?

Ejercicio 1 - otra forma

Otra forma de demostrar que no es computable es mediante una **reducción**, que ya vimos la clase pasada. La idea es aprovechar que la función es muy parecida a $Halt(x,x)$, y que ya sabemos que HALT no es computable, para demostrar que f_2 tampoco lo es. Primero, ¿qué relación exactamente tienen f_2 y HALT? $HALT(x,x) = \alpha(f_2(x)) = 1 - f_2(x)$.

Suponiendo que f_2 es computable, debe existir algún programa P_2 que la computa. Podemos definir entonces un nuevo programa P':

$$\Psi_{P_2}^{(1)}(X_1)$$
$$Y := 1 - Y$$

y por inspección es claro que

$$\Psi_{P'}(x) = 1 - \Psi_{P_2}(x) = 1 - f_2(x) = HALT(x,x)$$

Y esto resulta un absurdo, porque ya sabemos que HALT no es computable. Por lo tanto, f_2 tampoco puede serlo.

$F = 1 - G$

$g = 1 - F \qquad g + F = 1$

$F = 1 - g$

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

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

Asumo g computable $\rightarrow \exists P_G / \Psi_P^{(4)} = g$.

Como $g_2(x,e,x,w) = \begin{cases} 1 & \text{si } \Phi_x(x)\downarrow \wedge \underbrace{\Phi_e(w)\downarrow}_{\text{True}} \wedge \Phi_x(x) > 0 \\ 0 & \text{en otro caso} \end{cases} = f(x,$

siendo e el programa $Q(x)=0$

$\Phi_e(z) = id(z)$

$$F(x,y,z) = g(x,e,y,z) = \begin{cases} 1 & \text{si } \Phi_x(y) \wedge \underbrace{\Phi_e(z)}_{\text{True}} \wedge \Phi_x(y) > z \\ 0 & \text{otro caso} \end{cases}$$

Con e $\Phi_e = h(x)$

$h(x) = x$ (id)

Como escribí F no computable con g, puedo dar el programa y es ABS $\therefore$ g no era computable

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

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

Asumo G computable

$$f(x) = \begin{cases} 1 & \text{si } (g_3(x,x,x) = x+1) \\ 0 & \text{otro caso} \end{cases}$$

Pero puedo computar F! ABS. G no es computable

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

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

Asumo g computable

$F(x,y) = g(e,x,y)$ siendo e, programa que computa h / $h(x)=1$

$$g(e,x,y) = \begin{cases} 1 & \text{si } \Phi_x(y)\downarrow \wedge \underbrace{(\Phi_e \circ \Phi_x(y)\downarrow)}_{} \\ 0 & \text{otro caso} \end{cases}$$

True y $\Phi_x(y)\downarrow$ y por lógica de cortocircuito, lo es.

ABS! G no es computable

Ejercicio 5. 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 $f_4(0)$.

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

$$F_4(x) = \begin{cases} h(g_3'(x,x,x)) & \text{si } x \neq 0 \\ 1 & \text{si } \Phi_0(0)\downarrow \wedge \Phi_0(0) \neq 0 \\ 0 & \text{otro caso} \end{cases}$$

Siendo $h(x)$ la funcion $\begin{cases} 0 & \text{si } x = 0 \\ 1 & \text{otro caso} \end{cases}$

En este caso 0 es Posi ∞ TMB envoldo

$$F(x) = \begin{cases} 1 & \text{si } g'(x,x,x) = x \wedge x \neq 0 \\ 0 & \text{otro caso} \end{cases}$$

Ejercicio 6. Decimos que una función parcial computable f es *extensible* si existe g computable tal que $g(x) = f(x)$ para todo $x \in \mathrm{Dom}\, f$. 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).

Sea $H(x) = \begin{cases} 1 & \text{si } \Phi_x(x)\downarrow \\ \uparrow & \text{otro caso} \end{cases}$, que es parcial computable (se puede escribir con st y Fcod).

Sea $g(x) /\ g(x) = f(x)\ \forall x \in \mathrm{Dom}\, h$, es decir siempre que da 1. Pero si $g(x) \underbrace{\begin{cases} H(x) & \text{si } x \in \mathrm{Dom}(h) \\ 0 & \text{si no} \end{cases}}$

Ver si $\in \mathrm{Dom}(h)$ implica correr h si se $\uparrow$ no puede dar 0.

g, es halt (No computable), entonces No es Extensible

Ejercicio 7. Para cada $n \in \mathbb{N}$ definimos una función total $g_n : \mathbb{N} \to \mathbb{N}$ de la siguiente manera:

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

t no es fijo

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?

A) Supongamos g computable. Entonces $Halt(x) = \begin{cases} 1 & \text{si } \Phi_x(x)\downarrow = g(x+1,x) \\ 0 & \text{sino} \end{cases}$

Como Halt no es Computable ABS, viene de suponer g computable

B) $\forall$

Ejercicio 8. 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 $x \in \mathbb{N}$.

A)

Supongamos que $g(n,x) = F_n(x)$ existe y es total computable.

Entonces existe un $\underbrace{n' / g(n',x)}_{\text{Esto es lo bruto}} = h(x) \;/\; h(x) = g(x,x) + 1$ que es computable xq g lo es.

$$g(n',n') = h(n') = g(n',n') + 1$$ ABS, de suponer g computable.

B) Si, como son funtor uno a uno o todo uno.

El argumento de antes mostraria que h no esta en k menor.

Ejercicio 9. 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.

A) $a : \mathbb{N} \to \mathbb{N}$ total $/ \forall f : \mathbb{N} \to \mathbb{N}$ total computable $\exists e_f \in \mathbb{N} / a(y) \geq f(y) \; \forall y > e_f$

idea: No! ya que si $\not\exists$ es la funcion mas grande, y existe a, o es una f ∴ debe existir una mas grande?

Supongamos que a existe y es computable. Si a es total computable, creo $g(x) = a(x) + 1$. que es computable porque a lo es. a deberia superar a h desde algun e_f a g ya que g es una $F : \mathbb{N} \to \mathbb{N}$ total computable.

$\exists e_f / \forall y > e_f, a(y) \geq g(y) \equiv a(y) \geq a(x) + 1$. ABS! vino de suponer a computable.

b) $G : \mathbb{N} \to \mathbb{N} / (\exists n_0 \wedge \exists F : \mathbb{N} \to \mathbb{N} / g(y) \leq f(y) \; \forall n_0 \geq e$, entonces g es computable

Sea $F(y) = 1$. Es computable total.

Sea $g(y) = \begin{cases} 1 & \text{si } \Phi_y(y)\downarrow \\ 0 & \text{sino} \end{cases}$, que total. $\forall n_0 \geq 0 \quad g(y) \geq 1$. ∴ g es computable ABS!

g es Halt que no es computable

Falso por contra ej.

Ejercicio 10. Dada una función total $f : \mathbb{N} \to \mathbb{N}$, un *aproximador* de f es una función total $g : \mathbb{N}^2 \to \mathbb{N}$ tal que para todo x, $g(x,t) = f(x)$ para todo t salvo finitos valores. Dicho de otra manera, $\lim_{t \to \infty} g(x,t) = f(x)$. Decidir si son verdaderas o falsas las siguientes afirmaciones. Justificar la respuesta.

a. Si f es computable entonces tiene un aproximador computable.

b. Si f tiene un aproximador computable entonces f es computable.

Dado $F : \mathbb{N} \to \mathbb{N}$. Un aproximador de f es una func total $g : \mathbb{N}^2 \to \mathbb{N} / \forall x, g(x,t) = f(x) \; \forall t - \mathcal{C}$

conj finito de valores

A) Si, ya que si $\# \mathcal{C} = 0$ $g(x,t) = f(x)$, luego $\forall \underbrace{t \in \mathcal{C}}_{\text{Finito}}$, defino un if con $f(x) + 1$.

b) Supongamos que si:

Tomemos $F(x) = \begin{cases} 1 & \text{si } \Phi_x(x)\downarrow \\ 0 & \text{sino} \end{cases}$ que es No computable. Sea g su aproximador.

$$g(x,t) = \begin{cases} 1 & \text{si } Step(x,x,t) = 1 \\ 0 & \text{sino} \end{cases}$$

(tiempo en $\leq t$ pasos. Step es computable y en el infinito, los errores son finitos por ende.)

Ejercicio 11. 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:

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

A) $f : \mathbb{N}^2 \to \mathbb{N}$ $\forall n \in \mathbb{N}$ $g_n(x) = f(n, x)$ es total computable, entonces f también.

Sea $f(n, x) = \begin{cases} 1 & \text{si } \Phi_n(n) \downarrow \\ 0 & \text{sino} \end{cases}$ (no computable) (ignoro x)

Sea g_n, Entonces si n está definido, como es fijo es 1 constante g, sino 0. Eso Es computable.

$\therefore$ Es Falso

B) Como es constante, si es computable, el programa existe, yo no lo conozco. Es constante

Segun dice, Halt(e, e) es True o False. Pero tomar con Pinzas. Puede ser