Ciencias de la Computación (2021-22)

El teorema de recursión

Ejercicio 1.– Probar que existe $\mathtt{p} \in \mathtt{GOTO_P}$ tal que para cualesquiera $x_1, x_2 \in \mathbb{N}$

$$[\![\mathtt{p}]\!]^{(2)}(x_1, x_2) = [\![\mathtt{p}]\!]^{(2)}([\![\mathtt{p}]\!]^{(1)}(x_1), [\![\mathtt{p}]\!]^{(1)}(x_2)) + 1$$

Ejercicio 2.– Probar que no existe una transformación automática de programas $\mathtt{T} : \mathtt{GOTO_P} \to \mathtt{GOTO_P}$ tal que cambie el resultado en 0 (incluido $\uparrow$) a todo programa.

Ejercicio 3.– Sea $\sigma : \mathbb{N} \to \mathbb{N}$ una permutación recursiva (es decir, una biyección recursiva). La permutación σ induce una permutación sobre las instrucciones de GOTO

$$I \mapsto I^\sigma$$

donde I^σ es la instrucción de código $\sigma(\#(I))$. Así mismo, σ induce una transformación $\mathtt{p} \mapsto \mathtt{p}^\sigma$ donde $\mathtt{p}^\sigma$ es el programa resultante de sustituir cada instrucción I de $\mathtt{p}$ por su transformada I^σ (eliminando, si es necesario, la última instrucción I^σ si es $\mathtt{Y} \leftarrow \mathtt{Y}$). Se pide:

1. Prueba que si σ es primitiva recursiva, entonces también lo es la función $f(\#(\mathtt{p})) = \#(\mathtt{p}^\sigma)$
2. Demostrar que existe un programa $\mathtt{p}$ tal que $[\![\mathtt{p}]\!] = [\![\mathtt{p}^\sigma]\!]$
3. ¿Existe un programa $\mathtt{p}$ (distinto del programa vacío) tal que (en el dominio de $[\![\mathtt{p}]\!]$) se verifique que $[\![\mathtt{p}]\!] = [\![\mathtt{p}^\sigma]\!] = [\![(\mathtt{p}^\sigma)^\sigma]\!]$?

Ejercicio 4.– Dado $\mathtt{p} \in \mathtt{GOTO_P}$, denotaremos por $\mathtt{p}^*$ el programa que resulta de eliminar de $\mathtt{p}$ todas las etiquetas de las instrucciones etiquetadas. Decidir razonadamente si los siguientes conjuntos son recursivos:

- $A = \{\#(\mathtt{p}) \ : \ [\![\mathtt{p}]\!] = [\![\mathtt{p}^*]\!]\}$
- $B = \{\#(\mathtt{p}) \ : \ [\![\mathtt{p}]\!] \neq [\![\mathtt{p}^*]\!]\}$
- $C = \{\#(\mathtt{p}) \ : \ \mathtt{p} = \mathtt{p}^*\}$

Ejercicio 5.– Probar que existe $e \in \mathbb{N}$ tal que $e \in \mathcal{K}$ y $W_e^{(1)} = \mathcal{K}$.

Indicación: Considérese la función $f : \mathbb{N}^2 - \to \mathbb{N}$ definida como:

$$f(x,y) = \begin{cases} 0 & \text{si } x = y \\ \varphi_y^{(1)}(y) & \text{si } x \neq y \wedge y \in \mathcal{K} \\ \uparrow & \text{e.c.o.c.} \end{cases}$$

probar que es recursiva y aplicar el teorema de recursión.

Ejercicio 6.– Usando el t. de recursión, probar que existe $\mathtt{p} \in \mathtt{GOTO_P}$ que sólo para sobre su propio código.

Ejercicio 7.– Sea $f : \mathbb{N} \to \mathbb{N}$ recursiva. Probar que:

1. El conjunto $\{e \ : \ \exists k(\forall n \in \mathbb{N})[\varphi_e(n) = f(n)^k]\}$ no es recursivo

2. Existe $e \in \mathbb{N}$ tal que para todo n, $\varphi_e(n) = f(n)^e$.
3. Probar que **no** es extensional el siguiente predicado $\mathtt{P}(\mathtt{p}) \equiv [\![\mathtt{p}]\!]^{(1)}(x) = x^{\#(\mathtt{p})}$
4. Aplicando diagonalización, probar que no es recursivo el conjunto

$$A = \{e \ : \ (\forall n \in \mathbb{N})[\varphi_e(n) = f(n)^e]\}$$

Ejercicio 8.– Para cada $p \in \mathbb{N}$, notaremos por $\mathbb{N}_p$ el conjunto de los múltiplos de p.

1. ¿Es recursivo el conjunto $A = \{e \in \mathbb{N} : (\exists p \in \mathbb{N})[W_e = \mathbb{N}_p]\}$?
2. Probar que existe e tal que $W_e = \mathbb{N}_e$.
3. Probar que el predicado $P(e) \equiv W_e = \mathbb{N}_e$ **no** es extensional.
4. Probar que P no es recursivo (*Indicación*: aplicar diagonalización).

Ejercicio 9.– En cada uno de los siguientes casos, probar que existe una única función computable h, que además es total, verificando las condiciones exigidas:

1. $h(x) = \begin{cases} 1 & \text{si } x = 0 \\ h(\lfloor \frac{x}{2} \rfloor)^{10} & \text{si } x > 0 \end{cases}$
2. Sea $k \in \mathbb{N}$ un número fijo y $f, g : \mathbb{N} \to \mathbb{N}$ tales que $\forall x \in \mathbb{N},\ f(x+1) < x+1$, siendo $f, g \in \mathcal{R}$, y definamos $h : \mathbb{N} \to \mathbb{N}$ por

$$\begin{aligned} h(0) &= k \\ h(x+1) &= g(h(f(x+1))) \end{aligned}$$

(Indicación: Probar la existencia usando el teorema de recursión y la unicidad por inducción en x).

Ejercicio 10.– Probar que existe $f \in \mathcal{R}^{(2)}$ tal que

$$f(x,y) = \begin{cases} 1 & \text{si } x = 0 \\ f(x-1, f(x \dot{-} y, y)) & \text{e.o.c.} \end{cases}$$

Indicación: Probar la existencia utilizando el teorema de recursión y la unicidad probando que por inducción en x que la función $f_x(y) = f(x,y)$ es única.

Ejercicio 11.– Sea $g : \mathbb{N} \to \mathbb{N}$ una función recursiva. Probar que existe $f \in \mathcal{P}^{(1)}$ tal que para todo $x \in \mathbb{N}$

$$f(x) = \begin{cases} f(g(x)) & \text{si } x \text{ es par} \\ x & \text{si } x \text{ es impar} \end{cases}$$

Indicación: Considérese la función $h \in \mathcal{P}^2$ definida por

$$h(u,x) = \begin{cases} \varphi_u(g(x)) & \text{si } x \text{ par} \\ x & \text{e.o.c.} \end{cases}$$

Pruébese que es recursiva y apliquése el teorema de recursión.

Ejercicio 12.– Probar que existen dos programas $\mathtt{p}$ y $\mathtt{q}$, con códigos consecutivos, tal que

$$\forall x \in \mathbb{N}\,[[\![\mathtt{p}]\!](x)\downarrow \implies [\![\mathtt{p}]\!](x) = [\![\mathtt{q}]\!](x)]$$

Indicación: Considérese $h(u,x) = \varphi_u(x) + |\varphi_u(x) - \varphi_{u+1}(x)|$, probar que es recursiva y aplicar el t. de recursión.