Lógica y computabilidad
Primer cuatrimestre — 2017

Práctica 5: Programas en $\mathcal{S}$. Funciones y predicados computables

Programas en $\mathcal{S}$

A lo largo de esta práctica es posible reutilizar como *macro* cualquier función que haya sido definida en ejercicios o items anteriores.

Un *predicado* P es una función $P : \mathbb{N} \to \mathbb{N}$ tal que su imagen está contenida en $\{0, 1\}$.

1. Escribir en lenguaje $\mathcal{S}$ algoritmos para calcular las siguientes funciones

(*a*) La suma

$$suma(x_1, x_2) = x_1 + x_2$$

(*b*) El producto

$$producto(x_1, x_2) = x_1 \cdot x_2.$$

(*c*) Exponenciación

$$exp(x_1, x_2) = x_1^{x_2}$$

2. Mostrar que la función vacía $\emptyset : \mathbb{N}^k \to \mathbb{N}$ definida por

$$\emptyset(x_1, \ldots, x_k) = \uparrow,$$

es computable en $\mathcal{S}$.

3. Escribir programas en lenguaje $\mathcal{S}$ que calculan los siguientes predicados:

(*a*) La igualdad

$$igual(x_1, x_2) = \begin{cases} 1 & \text{si } x_1 = x_2 \\ 0 & \text{si } x_1 \neq x_2. \end{cases}$$

(*b*) La distinción

$$distinto(x_1, x_2) = \begin{cases} 1 & \text{si } x_1 \neq x_2 \\ 0 & \text{si } x_1 = x_2. \end{cases}$$

(*c*) La relación mayor

$$mayor(x_1, x_2) = "x_1 > x_2" = \begin{cases} 1 & \text{si } x_1 > x_2 \\ 0 & \text{si } x_1 \leq x_2. \end{cases}$$

(*d*) La relación menor

$$menor(x_1, x_2) = "x_1 < x_2" = \begin{cases} 1 & \text{si } x_1 < x_2 \\ 0 & \text{si } x_1 \geq x_2. \end{cases}$$

(*e*) La negación

$$neg(x_1) = \text{''}\neg x_1\text{''} = \begin{cases} 1 & \text{si } x_1 = 0 \\ 0 & \text{sino.} \end{cases}$$

4. Sea $f(x)$ el mayor número natural k con la propiedad que $2 \cdot k \leq x$. Es decir,

$$f(x) = \max\{k \in \mathbb{N} : 2 \cdot k \leq x\}.$$

(*a*) Escribir un programa en $\mathcal{S}$ que compute a f.
(*b*) Probar que $f(x) = \lfloor x/2 \rfloor$.
(*c*) Dar un programa en $\mathcal{S}$ que compute el indicador de paridad

$$par(x_1) = \begin{cases} 1 & \text{si } x_1 \text{ es par} \\ 0 & \text{sino.} \end{cases}$$

(*d*) Repetir los items anteriores para otros naturales en lugar de 2.

5. Escribir programas en $\mathcal{S}$ que computen la funciones

(a) $f(x) = \lfloor \sqrt{x} \rfloor$ (b) $f(x) = \begin{cases} \lfloor \log_2 x \rfloor & \text{si } x \neq 0 \\ \uparrow & \text{si } x = 0 \end{cases}$

6. Dar un programa en $\mathcal{S}$ que dado un número natural devuelva su último dígito. Análogamente, dar otro que devuelva su primer dígito.

7. Dado un número natural x se definen la sucesiones a_n y b_n de la siguiente manera:

(*a*) $a_1 = x, \quad a_{n+1} = \begin{cases} a_n + 1 & \text{si } a_n \text{ es impar} \\ a_n/2 & \text{si } a_n \text{ es par.} \end{cases}$

(*b*) $b_1 = x, \quad b_{n+1} = \begin{cases} 3b_n + 1 & \text{si } b_n \text{ es impar} \\ b_n/2 & \text{si } b_n \text{ es par.} \end{cases}$

Escribir un programas en $\mathcal{S}$ que computen la funciones $f(n)$ y $g(n)$ que determinan el primer valor de n en el que $a_n = 1$ y $b_n = 1$ respectivamente. ¿Es verdad que $f(n)$ o $g(n)$ son menores o iguales que n?

8. Escribir un programa $\mathcal{S}$ que compute la siguiente función f de dos variables. Dados números x_1, x_2, el número $f(x_1, x_2)$ es el menor natural n tal que las últimas tres cifras de x_1^n son x_2, en el caso que x_2 tenga a lo sumo tres cifras. Si x_2 tiene más de tres cifras o no hay ninguna potencia como la pedida, entonces el programa no termina. ¿Es posible analizar cuáles son los valores de x_1, x_2 que hacen que el programa no termine?

Random Fact. La canción "Come together" del grupo musical inglés *The Beatles* habla sobre sus integrantes; cada verso hace referencia a uno de los músicos Ringo Starr, George Harrison, John Lennon y Paul McCartney en ese orden.

Funciones computables

Decimos que una función f es *computable* (o "$\mathcal{S}$-computable" o "Turing computable") si existe un programa en $\mathcal{S}$ que computa f.

En los siguientes ejercicios, siempre que se pida escribir un programa en $\mathcal{S}$ que compute cierta función o predicado a partir de otras funciones o predicados, utilizar los programas de estos últimos como macros.

9. Sea *Comp* la clase de funciones computables. Probar que:

(*a*) (*i*) La función sucesor $suc : \mathbb{N} \to \mathbb{N}$

$$suc(x) = x + 1$$

está en $Comp$.

(*ii*) Las proyecciones $P_{in} : \mathbb{N}^n \to \mathbb{N}$

$$P_{in}(x_1, \ldots, x_n) = x_i$$

está en $Comp$.

(*iii*) Las constantes $C_{kn} : \mathbb{N}^n \to \mathbb{N}$

$$C_{kn}(x_1, \ldots, x_n) = k$$

está en $Comp$.

(*b*) Si $f : \mathbb{N}^k \to \mathbb{N}$ y $g_i : \mathbb{N}^r \to \mathbb{N}, 1 \leq i \leq k$ están en $Comp$ entonces la composición:

$$h(x_1, \ldots, x_r) = f(g_1(x_1, \ldots, x_r), \ldots, g_k(x_1, \ldots, x_r))$$

está en $Comp$.

(*c*) Si $g : \mathbb{N}^2 \to \mathbb{N}$ es está en $Comp$ y $k \in \mathbb{N}$, entonces la función $h : \mathbb{N} \to \mathbb{N}$ definida por

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

está en $Comp$.

(*d*) Similarmente si $g : \mathbb{N}^{n+2} \to \mathbb{N}$ y $f : \mathbb{N}^n \to \mathbb{N}$ están en $Comp$ entonces $h : \mathbb{N}^{n+1} \to \mathbb{N}$ definida por

$$\begin{aligned} h(x_1, \ldots, x_n, 0) &= f(x_1, \ldots, x_n) \\ h(x_1, \ldots, x_n, t+1) &= g(t, x_1, \ldots, x_n, h(x_1, \ldots, x_n, t)) \end{aligned}$$

está en $Comp$.

(*e*) La función de decisión $d : \mathbb{N}^4 \to \mathbb{N}$

$$d(x, y, s, t) = \begin{cases} s & \text{si } x = y \\ t & \text{si } x \neq y \end{cases}$$

está en $Comp$.

10. Sea $P : \mathbb{N} \to \{0, 1\}$ un predicado computable. Dadas una variable V y una etiqueta L, dar un programa en $\mathcal{S}$ que realice la instrucción

$$IF\ P(V) = 1 \quad GOTO\ TO\ [L]$$

11. Sea $f : \mathbb{N} \to \mathbb{N}$ una función parcial computable. Escribir un programa en $\mathcal{S}$ que compute la función

$$g(x) = \begin{cases} 1 & \text{si } f(x) = 1 \\ \uparrow & sino. \end{cases}$$

12. Sea $g : \mathbb{N} \to \mathbb{N}$ una función parcial computable. Dar un programa en $\mathcal{S}$ que compute a la función

$$h(x) = \begin{cases} 1 & \text{si } g(x) \downarrow \\ \uparrow & \text{si } g(x) \uparrow. \end{cases}$$

13. Sean P y Q predicados computables. Dar programas en $\mathcal{S}$ de $\neg P$, $P \vee Q$, $P \wedge Q$ y $P \to Q$.

14. Sean $P_1, \ldots, P_k : \mathbb{N} \to \{0,1\}$ predicados totales computables mutuamente excluyentes y exhaustivos (es decir para un número natural, exactamente uno de ellos toma el valor 1) y $g_1, \ldots, g_k$ funciones computables. Dar un programa en $\mathcal{S}$ que compute

$$f(n) = \begin{cases} g_1(n) & \text{si } P_1(n) = 1 \\ & \vdots \\ g_k(n) & \text{si } P_k(n) = 1 \end{cases}$$

¿Qué sucede si no todos los predicados son totales? ¿Y si no son mutuamente excluyentes y/o exhaustivos?

15. Sea f una función computable. Probar que existen infinitos programas que computan la misma función.

16. Sea $f : \mathbb{N} \to \mathbb{N}$ una biyección computable (en particular f es total). Dar un programa en $\mathcal{S}$ que compute la inversa de f.

17. Sean $f : \mathbb{N} \to \mathbb{N}$ y $g : \mathbb{N} \to \mathbb{N}$ funciones parciales computables tales que existe un predicado *computable* $P : \mathbb{N} \to \{0,1\}$ que cumple

$$P(n) = 1 \Leftrightarrow f(n) \downarrow .$$

Dar un programa en $\mathcal{S}$ que compute

$$h(n) = \begin{cases} f(n) & \text{si } f(n) \downarrow \\ g(n) & \text{si } f(n) \uparrow \end{cases}$$