Parcial de computabilidad

Lógica y computabilidad

Verano 2018

El examen es a libro abierto y se puede suponer demostrado lo dado en las clases y los ejercicios de las guías colocando referencias claras. Entregar cada ejercicio en hojas separadas. En cada hoja debe figurar nombre, apellido y número de orden.

Ejercicio 1. Considere la función listaOrdenada$(x) : \mathbb{N} \to \mathbb{N}$ que dada una lista x devuelve otra lista con todos los elementos de la lista x que son mayores que 0, ordenados de mayor a menor y sin repetir. Por ejemplo:

$$\text{listaOrdenada}([3,\ 1,\ 2]) = [3,\ 2,\ 1]$$
$$\text{listaOrdenada}([3,\ 1,\ 2,\ 3,\ 4,\ 1]) = [4,\ 3,\ 2,\ 1]$$
$$\text{listaOrdenada}([3,\ 1,\ 2,\ 0,\ 4,\ 1]) = [4,\ 3,\ 2,\ 1]$$

Demuestre que la función listaOrdenada(x) es primitiva recursiva.

Ejercicio 2. Decida y demuestre si la siguiente función es computable.

$$g(x,y,z) = \begin{cases} 1 & \text{si } \exists w \text{ tal que } \Phi_x^{(1)}(w) \cdot \Phi_y^{(1)}(w) = z \\ \uparrow & \text{otro caso.} \end{cases}$$

Ejercicio 3. Decida y demuestre si la siguiente función es computable:

$$f(x) = \begin{cases} 1 & \text{si } \forall y \Phi_x^{(1)}(y) = y^2 + x \\ 0 & \text{otro caso.} \end{cases}$$

Ejercicio 4. Decida y justifique si los siguientes conjuntos son p. r., c. e., co-c. e. o computables:

a. $C_1 = \left\{x : \forall y \Phi_x^{(1)}(2y) \uparrow\right\}$

b. $C_2 = \left\{x : \forall y \Phi_x^{(1)}(2y) \downarrow\right\}$

LU: Lógica y Computabilidad – 1° Parcial

DNI: 38.992.934 N°Orden:

1	2	3	4	N
A	A	A	A	10

B Corrigió Tomás 22 2 18

1) Sea la función listaOrdenada(x): $\mathbb{N} \to \mathbb{N}$ que devuelve una lista con los elementos de x mayores a cero, sin repetir y ordenados de mayor a menor, vemos que podemos definir las funciones:

* $seRepite(i,x) = \begin{cases} 1 & \text{si } x[i] \text{ se repite en } x \text{ al final} \\ 0 & \text{si no} \end{cases} = \begin{cases} 1 & \text{si } (\exists t) \leq |x|\ x[t] = x[i] \wedge t > i \\ 0 & \text{si no} \end{cases} = \propto(\propto(\min_{t \leq |x|} x[t] = x[i] \wedge t > i \wedge t \neq 0))$ (puesto que si $\exists$ tal $t \Rightarrow x[t] = x[i]$ con $t > i$ y $x[i]$ se repite con índice mayor, y $t \in \{i+1, \dots, |x|\}$, $t > 0 \Rightarrow \propto(\propto(t)) = 1$)

Vemos que seRepite(i,x) indica si $x[i]$ se repite en x luego de i. Dicha función es composición de la minimización acotada, $\propto$, operadores de listas y operadores lógicos. Todos ellos son p.r. $\Rightarrow$ seRepite es p.r.

* sinRepes(x) = lista x sin elementos repetidos $= \min_{z \leq maxLista(x)} \underbrace{(\forall i) \leq |x|\ (\exists t) \leq |z|\ z[t] = x[i]}_{\text{todo elemento de } x \text{ está en } z} \wedge \underbrace{(\forall t) \leq |z|\ \propto(seRepite(t,z))}_{z \text{ no contiene repetidos}}$

* $maxLista(x) = nprimo(|x|+1)^{\times[\min_{t \leq |x|} (\forall t) \leq |x|\ x[t] \leq x[i]]^{|x|}}$

sinRepes nos devuelve una lista con los elementos de x sin repetir (si alguno estuviese repetido, para la aparición de menor índice de la repetición $\propto(seRepite(t,z)) = 0$. Por otro lado, maxLista nos da una cota superior para la minimización acotada de sinRepes, tomando el siguiente primo al último de la codificación de x y elevándolo al elemento máximo de x, que a su vez está elevado a la cantidad de elementos de x (la idea es tener una cota lo suficientemente grande para abarcar todas las listas necesarias).

Ambas funciones son composición de funciones p.r. por lo que maxLista es p.r. y luego sinRepes también.

En base a todas estas funciones podemos definir listaOrdenada de x como:

* listaOrdenada(x) = "lista de los elementos de $x > 0$, sin repetidos y ordenados de mayor a menor" = listaDec(sinRepes(Mayores Cero(x)))

donde Mayores Cero $(x) = \min_{z \leq maxLista(x)} \underbrace{(\forall t) \leq |x|\ (x[t] = 0 \vee \sum_{j=0}^{|x|} x[j] = z[j]) = \sum_{j=0}^{|x|} x[j] = x[j]) \wedge |z| \leq |x| \wedge (\forall i) \leq |z|}_{\text{todo elemento de } x \text{ aparece en } z \text{ igual cantidad de veces si el mayor no cero}}$

donde * Mayores Cero $(x) = \min_{z \leq maxLista(x)} (\forall i) \leq |x|\ \underbrace{(x[i] > 0 \to (\sum_{j=1}^{|x|} x[i] = z[j]) = (\sum_{j=1}^{|x|} x[i] = x[j]))}_{\text{todo elemento de } x > 0 \text{ aparece en } z \text{ igual cantidad de veces}} \wedge \underbrace{|z| \leq |x|}_{z \text{ no supera a } x \text{ en longitud}} \wedge \underbrace{\propto(\sum_{j=1}^{|z|} z[j] = 0)}_{\text{no hay elementos en } z = 0}$

y * Lista Dec $(x) = \min_{z \leq maxLista(x)} \underbrace{(\forall i) \leq |x|\ (\exists t) \leq |z|\ z[t] = x[i]}_{\text{todo elemento de } x \text{ está en } z} \wedge \underbrace{|x| = |z|}_{\text{mismo longitud}} \wedge \underbrace{(\forall j) < |z|\ z[j] \geq z[j+1]}_{\text{orden decreciente}}$

Es decir, Mayores Cero devuelve una lista con los elementos de x mayores a cero (sobreescribiendo de repetirse su cantidad de apariciones en x) y Lista Dec nos devuelve los elementos de la lista x ordenados de forma decreciente, teniendo en cuenta que x es una lista sin elementos repetidos (de no ser así la lista resultante podría contener elementos que no están en x). Ambas funciones son composición de funciones p.r. (la sumatoria es p.r.) por lo que ambas son p.r.

En base a su definición, listaOrdenada(x) nos devuelve la lista pedida y al ser composición de funciones p.r., esta también es p.r.

LU: | LÓGICA Y COMPUTABILIDAD - 1° PARCIAL | 22 | 2 | 18

DNI: 38.992.934 N° ORDEN:

(B)

2) Sea $g(x,y,z) = \begin{cases} 1 & \text{si } \exists w \text{ tal que } \phi_x^{(1)}(w) \cdot \phi_y^{(1)}(w) = z \\ \uparrow & \text{si no} \end{cases}$ demostremos que es parcial-computable al exhibir el programa P:

P: $Y \leftarrow Y+1$ (terminamos $Z_1 \leftarrow \langle w,t \rangle$ donde w será la variable definida en ϕ_x y ϕ_y y t la cantidad de pasos de STP y SNAP)

[B] $Z_2 \leftarrow STP^{(1)}(L(Z_1), X_1, r(Z_1))$ ($Z_2 \leftarrow STP^{(1)}(w,x,t)$)

IF $Z_2 = 0$ GOTO A (como $Z_2 = 0$, el programa x no termina en t pasos para w, por lo que cambio la tupla)

$Z_3 \leftarrow STP^{(1)}(L(Z_1), X_2, r(Z_1))$ ($Z_3 \leftarrow STP^{(1)}(w,y,t)$)

IF $Z_3 = 0$ GOTO A (como $Z_3 = 0$, el programa y no terminó en t pasos para w y cambiamos la tupla)

$Z_2 \leftarrow SNAP^{(1)}(L(Z_1), X_1, r(Z_1))$ ($Z_2 \leftarrow \phi_x^{(1)}(w)$)

$Z_3 \leftarrow SNAP^{(1)}(L(Z_1), X_2, r(Z_1))$ ($Z_3 \leftarrow \phi_y^{(1)}(w)$)

$Z_4 \leftarrow Z_2 \cdot Z_3$ ($Z_4 \leftarrow \phi_x^{(1)}(w) \cdot \phi_y^{(1)}(w)$)

IF $Z_4 = X_3$ GOTO E (si $\phi_x^{(1)}(w) \cdot \phi_y^{(1)}(w) = z \Rightarrow$ el programa termina devolviendo 1)

[A] $Z_1 \leftarrow Z_1 + 1$ (Z_1 pasa a tener el siguiente valor de $\langle w,t \rangle$)

GOTO B (siguiente iteración del ciclo)

Como se puede ver, $\Psi_P(x,y,z) = \begin{cases} 1 & \text{si } \exists w \text{ tal que } \phi_x^{(1)}(w) \cdot \phi_y^{(1)}(w) = z = g(x,y,z) \\ \uparrow & \text{si no} \end{cases} \Rightarrow$ P computa a g y g es parcial computable.

Aclaración: los macros utilizados en P son los de las funciones computables llamadas a ... fueron definidos en las guías prácticas.

podías hacerlo en una líneas con un Z no acotado.

LU: | LÓGICA Y COMPUTABILIDAD – 1º PARCIAL | 22 | 2 | 18

DNI: 38.997.934 Nº ORDEN: | Corrigió: Franco (A)

3) $f(x) = \begin{cases} 1 & \text{si } \forall y\, \phi_x^{(1)}(y) = y^2 + x \\ 0 & \text{si no} \end{cases}$ Probaremos que esta función no es computable

Para ello, supondremos que lo es y tomaremos la función $g(x,y) = x + y^2$ la cual es p.r. por ser composición de funciones p.r. $(+, x, a^b)$ y por ende es computable. Con ella aplicamos el teorema de la recursión y vemos que existe un programa de número e tal que:

$\phi_e^{(1)}(y) = g(e,y) = e + y^2 \Rightarrow f(e) = 1$. Usaremos esta idea para lo siguiente: Pero... ¿cuándo usás este e?

Al definir $h(x,y) = \begin{cases} x + y^2 & \text{si } F(x) = 0 \\ \uparrow & \text{si no} \end{cases}$ vemos que $h(x)$ es parcial computable pues F lo es (computable) y luego puedo aplicarle el teorema de la recursión para ver que $\exists e' \in \mathbb{N} / \phi_{e'}^{(1)}(y) = h(e,y) = \begin{cases} y^2 + e' & \text{si } F(e) = 0 \\ \uparrow & \text{si no} \end{cases}$

Con esto, vemos que:

i) $F(e') = 1 \iff \forall y\, \phi_{e'}^{(1)}(y)(e') = y^2 + e'$ (definición de F) $\iff \forall y\, h(e',y) = y^2 + e'$ ($\phi_{e'}^{(1)}(y) = h(e',y)$) $\iff F(e') = 0$ (definición de h)

ii) $F(e') = 0 \iff \exists y\, \phi_{e'}^{(1)}(y)\uparrow \vee \phi_{e'}^{(1)}(y) \neq y^2 + e'$ (definición de F) $\iff \exists y\, h(e',y)\uparrow \vee h(e',y) \neq y^2 + e'$ ($\phi_e^{(1)}(y) = h(e',y)$) $\iff F(e') = 1$

(definición de h) En i) y ii) llegamos a un ABSURDO $\therefore$ F no es computable

(Alcanzaba solo con i) o con ii)

LU: Lógica y Computabilidad – 1° Parcial — 22 | 2 | 18

DNI: 38.992.934 N°Orden:

4) a) $\mathcal{C}_1 = \{x: \forall y\ \phi_x^{(1)}(2y)\uparrow\}$

co-ce: Si tomamos $\overline{\mathcal{C}_1} = \{x: \exists y\ \phi_x^{(1)}(2y)\downarrow\}$ luego vemos que la función característica de dicho conjunto es:

$g(x) = \begin{cases} 1 & \text{si } (\exists y)\phi_x^{(1)}(2y)\downarrow \\ \uparrow & \text{si no} \end{cases} = \alpha\left(\alpha\left(\min_{\langle t,w\rangle} STP^{(1)}(2*w, x, t)\right)\right)$ (por minimización no acotada, si $(\exists \langle t,w\rangle)\ STP^{(1)}(2*w,x,t) \Rightarrow$ $\alpha(\alpha(\langle t,w\rangle)) = 1$ y sino $g(x)\uparrow$) donde vemos que por ser composición de funciones parciales computables, g también lo es.

En consecuencia $\overline{\mathcal{C}_1}$ es c.e. pues $\overline{\mathcal{C}_1} = \{x: g(x)\downarrow\}$ y luego $\mathcal{C}_1$ es co-c.e. ✓

computable: Probaremos que $\mathcal{C}_1$ no es computable suponiendo que lo es y tomando su función característica: $F_{\mathcal{C}_1}(x) = \begin{cases} 1 & \text{si } \forall y\ \phi_x^{(1)}(2y)\uparrow \\ 0 & \text{si no} \end{cases}$

En base a eso, vemos que $\mathcal{C}_1 = \{x: F(x)\}$ y $\mathcal{C}_1$ es un conjunto de índices, puesto que si $x \in \mathcal{C}_1$ y $\phi_{x'}^{(1)} = \phi_x^{(1)} \Rightarrow \forall y\ \phi_{x'}^{(1)}(2y)\uparrow \Rightarrow$ $\Rightarrow x' \in \mathcal{C}_1$. Q.q.v. Vemos que dicho conjunto no es trivial, puesto que para los programas P y Q siguientes:

P: $Y \leftarrow Y+1$

Q: $Y \leftarrow Y+2$
[A] IF $Y \neq 0$ GOTO A

vemos que, si $\#P = e_1$ y $\#Q = e_2$, $e_1 \notin \mathcal{C}_1$ $(\forall y\ \phi_{e_1}^{(1)}(y)\downarrow)$ y $e_2 \in \mathcal{C}_1$ $(\forall y\ \phi_{e_1}^{(1)}(y)\uparrow \Rightarrow$ $\Rightarrow \forall y\ \phi_{e_2}^{(1)}(2y)\uparrow)$

En base a esto vemos que $\mathcal{C}_1$ es un conjunto de índices no trivial, y aplicando el teorema de Rice, vemos que $\mathcal{C}_1$ no es computable. ✓

c.e.: Si $\mathcal{C}_1$ fuese c.e., como es co-c.e. debería ser computable. Sin embargo, no lo es $\Rightarrow$ $\mathcal{C}_1$ no es c.e. ✓

p.r.: Como $\mathcal{C}_1$ no es computable, no es p.r. ✓

b) $\mathcal{C}_2 = \{x: \forall y\ \phi_x^{(1)}(2y)\downarrow\}$ Sabemos que el conjunto $Tot = \{x: \forall y\ \phi_x^{(1)}(y)\downarrow\}$ no es computable, p.r., c.e. ni co-c.e. Veamos si podemos reducir Tot a $\mathcal{C}_2$. Queremos hallar $F: \mathbb{N} \to \mathbb{N}$ / $f(x) \in \mathcal{C}_2 \iff x \in Tot$

$\left[f(x) \in \mathcal{C}_2 \iff \forall y\ \phi_{f(x)}^{(1)}(2y)\downarrow \iff \forall y\ \phi_x^{(1)}(y)\downarrow \iff x \in Tot\right]$ (paso creativo)

Para ello, tomamos $f(x) = s_1^1(e,x)$ y vemos que $\forall y\ \Phi_{f(x)}^{(1)}(2y)\downarrow \iff \forall y\ \phi_{s_1^1(e,x)}^{(1)}(2y)\downarrow \iff \phi_e^{(2)}(2y,x)\downarrow \forall y \iff \phi_x^{(1)}(y)\downarrow \forall y$

y vemos que para $\phi_e^{(2)}(y,x) = \begin{cases} 1 & \text{si } (\exists t)\left(2*t = y \wedge \phi_x^{(1)}(t)\downarrow\right) \\ \uparrow & \text{si no} \end{cases}$

* Si $\forall y\ \phi_e^{(2)}(2y,x)\downarrow \Rightarrow \forall y\ (\exists t)\ 2*t = 2y \wedge \phi_x^{(1)}(t)\downarrow \Rightarrow \forall y\ \phi_x^{(1)}(y)\downarrow\ (2*y = 2y) \Rightarrow x \in Tot$

* Si $\forall y\ \phi_x^{(1)}(y)\downarrow \Rightarrow \forall y\ \phi_e^{(2)}(2y,x)\ (2y = 2*y \wedge \phi_x^{(1)}(y) \Rightarrow$ por definición de $\phi_e^{(2)}$: $\phi_e^{(2)}(2y,x)\downarrow) \Rightarrow \forall y\ \phi_{s_1^1(e,x)}^{(1)}(2y)\downarrow$ (teorema del parámetro: $\phi_{s_1^1(e,x)}^{(1)}(y) = \phi_e^{(2)}(y,x)$) $\Rightarrow \forall y\ \phi_{f(x)}^{(1)}(2y)\downarrow$ $[f(x) = s_1^1(e,x)] \Rightarrow f(x) \in \mathcal{C}_2$

$\therefore$ $f(x) \in \mathcal{C}_2 \iff x \in Tot \Rightarrow Tot \leq \mathcal{C}_2$ y como Tot no es c.e, co-c.e., computable ni p.r., $\mathcal{C}_2$ tampoco. ✓

Resultados:

	c.e.	co-c.e.	computable	p.r.
C_1	×	✓	×	×
C_2	×	×	×	×

a	b
B	B