Guía 3 - Funciones no-computables y conjuntos c.e.

Solución de un alumno

Verano 2021

Ejercicio 1

a) realizado en práctica

b)

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

Tomamos el caso $y = x$, tenemos:

$$f_2'(x) = \begin{cases} 1 & si\ \Phi_x^{(1)}(x) = 0 \\ 0 & c.c. \end{cases}$$

Tenemos que mostrar que f_2' no es computable y que la reducción es computable.

- Primero mostremos que f_2' no es computable:

Tratemos de armar una función que genere una inconsistencia con f_2' y que sea computable:

$$g(x) = \begin{cases} 7 & si\ f_2'(x) = 1 \text{ (es decir } \Phi_x(x) = 0) \\ 0 & si\ f_2'(x) = 0 \text{ (es decir } \Phi_x(x) \neq 0 \text{ ó } \uparrow) \end{cases}$$

Notar que 7 podría ser cualquier valor distinto de 0.

En particular si $x = e$ siendo e el programa de $g(x)$ entonces:

$$g(e) \underset{\Phi_e(e)=0}{=} 7 \underset{\Phi_e(e)=0}{\neq} 0\ ABS!$$

- Nos falta mostrar que la reducción es computable: es directo con una proyección

---

c)

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

Definimos f_3' como:

$$f_3'(x) = f_3(x,x,0) = \begin{cases} 1 & si\ \Phi_x^{(1)}(x) \downarrow \text{ y } \Phi_x^{(1)}(x) > 0 \\ 0 & c.c. \end{cases}$$

Veamos que f_3' no es computable. Armamos una función $g(x)$ que sea computable y que genere una inconsistencia con f_3':

$$g(x) = \begin{cases} \uparrow & si\ f_3'(x) = 1 \text{ (es decir } \Phi_x^{(1)}(x) \downarrow \text{ y } \Phi_x^{(1)}(x) > 0) \\ 1 & si\ f_3'(x) = 0 \text{ (es decir } \Phi_x^{(1)}(x) \uparrow \text{ o } \Phi_x^{(1)}(x) \leq 0) \end{cases}$$

En particular si e donde e es el número del programa que computa g, entonces:

$$\Phi_e^{(1)}(e) \downarrow \text{ y } \Phi_e^{(1)}(e) > 0 \Leftrightarrow \Phi_e^{(1)}(e) \uparrow \text{ ó} \Phi_e^{(1)}(e) \uparrow \text{ ó } \Phi_e^{(1)}(e) \leq 0 \Leftrightarrow \Phi_e^{(1)}(e) = 1$$

La primera es falsa y la segunda también. Entonces ¡Absurdo!

Ejercicio 2

$$g_1(x,y) = \begin{cases} 1 & si\ \Phi_x(y) \uparrow \\ 0 & cc \end{cases}$$

$$\alpha(g_1(x,y)) = f_1(x,y)$$

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

Sea e el número de un programa que computa la función identidad:

$$g_2(x,e,z,w) = f_3(x,z,w)$$

$$g_3(x,y,z) = \begin{cases} z+1 & si\ \Phi_x(y) \downarrow\ y\ \Phi_x(y) \neq z \\ 0 & cc \end{cases}$$

$$g_3(x,x,x) \dot{-} x = f_4(x)$$

$$g_4(x,y,z) = \begin{cases} (\Phi_x \circ \Phi_y)(z) & si\ \Phi_y(z) \downarrow\ y\ (\Phi_x \circ \Phi_y)(z) \downarrow \\ 0 & cc \end{cases}$$

Tomo e como el número de un programa que computa la función $f(x) = 1$, entonces:

$$g_4(e, y, z) = \begin{cases} 1 & si\ \Phi_y(z) \downarrow \\ 0 & cc \end{cases} = f_1(y, z)$$

Ejercicio 3

$$g_3'(x, y, z) = \begin{cases} z & si\ \Phi_x(y) \downarrow\ \ y\ \Phi_x(y) \neq z \\ 0 & cc \end{cases}$$

$$g_3''(x, y, z) = \begin{cases} z & si\ \Phi_x(y) \downarrow\ \ y\ \Phi_x(y) \neq z \\ 0 & si\ x = 0 \\ 0 & cc \end{cases}$$

$$g_3'''(x) = g_3''(x, x, x) \dot{-} (x \dot{-} 1) = \begin{cases} 1 & si\ \Phi_x(y) \downarrow\ \ y\ \Phi_x(y) \neq x \\ 0 & si\ x = 0 \\ 0 & cc \end{cases}$$

$$g_3''' = f_4$$

Ejercicio 4

Podemos tomar la función:

$$HaltComp(x) = \begin{cases} \Phi_x(x) & si\ \Phi_x(x) \downarrow \\ \uparrow & cc \end{cases}$$

Si la anterior función fuese extensible entonces:

$$ExtHaltComp(x) = \begin{cases} \Phi_x(x) & si\ \Phi_x(x) \downarrow \\ f(x) & cc \end{cases}$$

existiría y sería computable, con $f(x)$ computable. Pero si esto sucede entonces también podemos computar:

$$Halt(x) =? \begin{cases} 1 & si\ ExtHaltComp(x) \neq f(x) \\ 0 & cc \end{cases}$$

MAL, ExtHaltComp(x) puede terminar y ser igual a f(x). **REVISAR**

Ejercicio 5

$$g_1(x) = \begin{cases} 1 & si\ Halt(1337, x) \equiv \Phi_x(1337) \downarrow \\ 0 & cc \end{cases}$$

Tomamos el número del siguiente programa:

```
Z1 <- Phi(X2,X2)
```

Y por teo. parámetro existe $S_1^1(x,e)$ tal que esconde la variable x en X2 dentro del programa. Entonces si componemos esta función con la anterior tenemos que:

$$h(x) = g_1(S(x,n)) = \begin{cases} 1 & si\ \Phi_{S(x,n)}(1337) \downarrow \equiv \Phi_x(x) \downarrow \\ 0 & cc \end{cases} = Halt(x)$$

$$g_2(x,y,z) = \begin{cases} 1 & si\ \Phi_x(z) \downarrow y\ \Phi_y(z) \downarrow y\ \Phi_x(z) > \Phi_y(z) \\ 0 & cc \end{cases}$$

Sea e el número de un programa que computa la identidad:

$$g_2'(x,y) = g_2(x,y,e) = \begin{cases} 1 & si\ \Phi_x(z) \downarrow y\ \Phi_x(z) > z \\ 0 & cc \end{cases}$$

Dado el programa con número e' el cual es:

```
Z1 <- Phi(X1,X2) //Programa X2 con entrada X1
Y <- X1
Y <- Y + 1
```

Sea la función $S_1^1(x,e')$ la función que esconde x en la variable X2, entonces:

$$\Phi_{S(x,e')}(y) = \Phi_{e'}(x,y)$$

$$g_2''(x,z) = g_2'(S(x,e'),z) = \begin{cases} 1 & si\ \Phi_{S(x,e')}(z) \downarrow y\ \Phi_{S(x,e')}(z) > z \\ 0 & cc \end{cases}$$

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

$$g_3(x) = \begin{cases} 13 & si\ \Phi_x\ es\ la\ constante\ 7 \\ 0 & cc \end{cases}$$

$$g_3'(x) = g_3(x) - 12 = \begin{cases} 1 & si\ \Phi_x\ es\ la\ constante\ 7 \\ 0 & cc \end{cases}$$

Sea el siguiente programa con número e:

```
Z1 <- Phi(X1,X1)
Y <- 7
```

$$\Phi_{S(x,e')}(y) = \Phi_{e'}(x,y)$$

$$g_3''(x) = g_3'(S(x,e)) = \begin{cases} 1 & si\ \Phi_{S(x,e)}\ es\ la\ constante\ 7 \\ 0 & cc \end{cases}$$

$$= \begin{cases} 1 & si\ \Phi_{S(x,e)}\ es\ la\ constante\ 7 \equiv \Phi_x(x)\downarrow \\ 0 & cc \end{cases}$$

$$= Halt(x)$$

$$g_2(x,y) = \begin{cases} 1 & si\ \Phi_x(y)\downarrow y\ \Phi_y(x)\downarrow y\ \Phi_x(y) \neq \Phi_y(x) \\ 0 & cc \end{cases}$$

Sea e el número de un programa que calcula la función nula. Entonces:

$$g_2(x,e) = \begin{cases} 1 & si\ \Phi_x(e)\downarrow y\ \Phi_x(e) \neq 0 \\ 0 & cc \end{cases}$$

El programa:

```
Z1 <- Phi(X1,X1)
Y <- 1
```

$$g_2(S(x,e'),e) = \begin{cases} 1 & si\ \Phi_{S(x,e')}(e)\downarrow y\ \Phi_{S(x,e')}(e) \neq 0 \equiv \Phi_x(x)\downarrow \\ 0 & cc \end{cases}$$

$$= Halt(x)$$

Ejercicio 6

$$f(x,y) = \begin{cases} 1 & si\ x = y \\ \uparrow & cc \end{cases}$$

Por teo. parámetro existe e número de programa tal que:

$$\Phi_e = f(x,e) = \begin{cases} 1 & si\ x = e \\ \uparrow & cc \end{cases}$$

Notar que e cumple con enunciado.

Ejercicio 7

Usando teo. de recursión ver que no son computable:

a)

$$h_1(x) = \begin{cases} 1 & si\ x \in Im(\Phi_x^{(1)}) \\ 0 & cc \end{cases}$$

$$g(x, y) = \begin{cases} y + 1 & si\ h_1(x) \equiv x \in Im(\Phi_x^{(1)}) \\ y & cc \equiv x \notin Im(\Phi_x^{(1)}) \end{cases}$$

Sea **e** el número de programa que compute $g'(x)$ (existe por teo. recursión) tal que:

$$g'(x) = g(x, e) = \begin{cases} e + 1 & si\ h_1(x) \equiv x \in Im(\Phi_x^{(1)}) \\ e & cc \equiv x \notin Im(\Phi_x^{(1)}) \end{cases}$$

En particular, si tomamos $g'(e)$:

$$g'(e) = e + 1 \Leftrightarrow e \in Im(\Phi_e^{(1)})$$

FALSO

$$g'(e) = e \Leftrightarrow e \notin Im(\Phi_e^{(1)})$$

FALSO

---

b)

$$h_2(x, y) = \begin{cases} 1 & si\ \Phi_x^{(1)}(y) \downarrow \wedge\ \Phi_x^{(1)}(y) > x \\ 0 & cc \end{cases}$$

$$h_2'(x, y) = \alpha(h_2(x, y)) \cdot (x + 1) = \begin{cases} 0 & si\ \Phi_x^{(1)}(y) \downarrow \wedge\ \Phi_x^{(1)}(y) > x \\ x + 1 & cc \end{cases}$$

$$\Phi_e(y) = h_2'(e, y) = \begin{cases} 0 & si\ \Phi_e^{(1)}(y) \downarrow \wedge\ \Phi_e^{(1)}(y) > e \\ e + 1 & cc \end{cases}$$

$$\Phi_e(e) \downarrow \wedge \Phi_e(e) > e \Leftrightarrow \Phi_e = 0$$

FALSO

$$\Phi_e(e) \uparrow \vee\ \Phi_e(e) \leq e \Leftrightarrow \Phi_e = e + 1$$

FALSO

---

c)

$$h_3(x) = \begin{cases} 1 & si\ Im(\Phi_x^{(1)})\ es\ infinita \\ 0 & cc \end{cases}$$

Sea **e** el número de programa que computa:

```
Z <- Phi(X2,X2)
Y <- X1
```

Sea $S(x, e)$ la función p.r. que esconde el valor de x en X2 dentro del programa y devuelve el número de tal programa.

Entonces:

$$g(x) = h_3(S(x, e)) = \begin{cases} 1 & si\ Im(\Phi_{S(x,e)}^{(1)})\ es\ inf \equiv \Phi_x^{(1)} \downarrow \\ 0 & cc \end{cases}$$

$$g(x) = Halt(x)$$

---

d)

$$h_4(x) = \begin{cases} 1 & si\ |Dom(\Phi_x^{(1)})| = x \\ 0 & cc \end{cases}$$

Sea **e** el número de programa que computa:

```
    X1 <- x
    X2 <- y

    Z <- Phi(X1,X1)
    IF X3 < X2 GOTO A
[R] GOTO R
[A] Y <- 1

//En X1 nos guardamos el programa a ejecutar (para transformar en halt)
//En X2 nos guardamos el número de programa
//En X3 es el parámetro que nos pasa
```

Sea $S(x_1, x_2, e)$ la función p.r. que esconde el valor de x_1 en X1 y x_2 en X2 dentro del programa (lo de color verde en el código), es decir:

$$e \underset{S(x_1,x_2,e)}{\longrightarrow} e_{x_1,x_2}$$

Entonces

$$g(x_1, x_2) = h_3(S(x_1, x_2, e)) = \begin{cases} 1 & si\ |Dom(\Phi_{S(x_1,x_2,e)}^{(1)})| = S(x_1, x_2, e) \\ 0 & cc \end{cases}$$

Por teo. de recursión, existe ***e'*** número del programa que computa la función:

$$g'(x) = g(x, e') = \begin{cases} 1 & si \ |Dom(\Phi_{e'}^{(1)})| = e' \equiv \Phi_x^{(1)} \downarrow \\ 0 & cc \end{cases}$$

$$g'(x) = Halt(x)$$

$$h_4(x) = \begin{cases} 1 & si \ |Dom(\Phi_x^{(1)})| = x \\ 0 & cc \end{cases}$$

$$g(x, y) = \begin{cases} \uparrow & si \ h_4(x) \\ 0 & si \ \neg h_4(x) \ \wedge \ x < y \\ \uparrow & cc \end{cases}$$

Sea **e** el número del programa que computa la función tal que:

$$\Phi_e(x) = g(x, e) = \begin{cases} \uparrow & si \ h_4(x) \\ 0 & si \ \neg h_4(x) \ \wedge \ x < e \\ \uparrow & cc \end{cases}$$

Entonces, en particular si tomamos $\Phi_e(e)$:

$$|Dom(\Phi_e^{(1)})| = e \Leftrightarrow h_4(e) \Leftrightarrow \Phi_e(e) \uparrow$$

FALSO

$$|Dom(\Phi_e^{(1)})| \neq e \Leftrightarrow \neg h_4(e) \Leftrightarrow \Phi_e(e) = 0 (\forall x < e) \text{ ó } \Phi_e(e) \uparrow \ (\forall x \geq e) \Leftrightarrow |Dom(\Phi_e^{(1)})| = e$$

FALSO

Ejercicio 8

Sean $C_1, ..., C_k$ conjuntos de índices de programas y sea $C = C_1 \cap ... \cap C_k$.

a) Demostrar que C es un conjunto de índices de programas (i.e., $C = x : \Phi_x \in \mathcal{C}$, con $\mathcal{C}$ una clase de funciones).

$$C = \{x : \Phi_x \in \mathcal{C}_1 \cap ... \cap \mathcal{C}_k\} = \{x : \Phi_x \in \mathcal{C}\}$$

b) Proponer un conjunto que no sea un conjunto de índices de programas y no sea computable.

$$A = \{x : \Phi_x \text{tiene al menos 5 instrucciones}\}$$

Ejercicio 9

(⇒)

$$\#p \in D \Leftrightarrow \Phi_p \in C_D$$

$$\Psi_p = \Psi_q \Rightarrow \Phi_p = \Phi_q \Rightarrow \Phi_q \in C_D \Rightarrow \#q \in D$$

(<=) **TERMINAR**

Ejercicio 10

$$g_1(x) = \begin{cases} 1 & si\ Dom(\Phi_x^{(1)}) = \\ 0 & cc \end{cases}$$

Podemos ver g_1 como una función característica del conjunto:

$$A = \{x : Dom(\Phi_x^{(1)}) = \}$$

Veamos que no es trivial; El siguiente programa no pertenece:

```
Y <- 0
```

El siguiente programa pertenece:

```
[A] GOTO A
```

Ahora veamos que es un conjunto de índices:

$$C = \{g(x) : g(x) \uparrow \forall x\}$$

$$g_3(x,y) = \begin{cases} 1 & si\ Dom(\Phi_x^{(1)}) \cup Dom(\Phi_y^{(1)}) = \mathbb{N} \\ 0 & cc \end{cases}$$

Sea e el número de un programa que se cuelga para cualquier entrada (trivial de hacer):

$$g_3'(x) = g_3(x,e) = \begin{cases} 1 & si\ Dom(\Phi_x^{(1)}) = \mathbb{N} \\ 0 & cc \end{cases}$$

$$A = \{x : Dom(\Phi_x) = \mathbb{N}\}$$

Sea C la clase de funciones:

$$C = \{g(x) : Dom(g) = \mathbb{N}\}$$

Entonces podemos escribir al conjunto A como:

$$A = \{x : \Phi_x \in C\}$$

Veamos que no es un conjunto trivial:

El programa vacío pertenece.

El siguiente programa no pertenece:

```
[A] GOTO A
```

Por Teo. Rice entonces no es computable.

$$g_4(x,y) = \begin{cases} \Phi_x^{(1)}(\Phi_y^{(1)}(72)) & si\ \Phi_x^{(1)} \circ \Phi_y^{(1)} es\ total \\ 73 & cc \end{cases}$$

Tomamos e como el número del programa que computa la función nula (total):

$$g_4'(y) = g_4(e,y) = \begin{cases} 0 & si\ \Phi_y\ es\ total \\ 73 & cc \end{cases}$$

$$g_4''(x) = \alpha(g_4'(x)) = \begin{cases} 1 & si\ \Phi_x\ es\ total \\ 0 & cc \end{cases}$$

La función $g_4''(x)$ es la función característica de Tot (que no es comp). Entonces no es comp. Entonces $g_4(x,y)$ tampoco es computable, porque $g_4''(x)$ es composición de funciones p.r. sobre $g_4(x,y)$.

Ejercicio 11

a) VERDADERO

Como B es computable, quiere decir que su función característica es computable. Que B sea c.e. implica que existe $g(x)$ parcial computable tal que:

$$B = Dom(g)$$

Entonces tomamos $g(x)$ como la función:

$$g(x) = \begin{cases} 1 & B(x) \\ \uparrow & cc \end{cases}$$

$g(x)$ es computable porque $B(x)$ es computable.

b) FALSO

A no computable $\Rightarrow \bar{A}$ no computable

Buscamos un conjunto que sea c.e. y no computable: el conjunto K.

c) FALSO

A comp sii A y $\bar{A}$ c.e.

Entonces K es c.e. pero $\bar{K}$ no es c.e.

Ejercicio 12

Decidir cuáles de los siguientes conjuntos son p.r., cuáles son computables, cuáles son c.e., cuáles son co-c.e. y demostrar en cada caso:

$$C_1 = \{x : \Phi_x^{(1)}(x) = 2 \cdot x\}$$

Es un conjunto de índices de la clase de funciones

$$C = \{g(x) : g(x) = 2 \cdot x\}$$

No es trivial (trivial ver cual está y cual no). Por teo de Rice no es comp.

Como no es comp entonces no puede ser co-ce y ce.

Veamos ce. Buscamos $f(x)$ parcial comp tq $f(x) \downarrow \Leftrightarrow x \in C_1$:

$$f(x) = (\exists t)(STP(x, x, t) = 1 \wedge r(SNAP(x, x, t))[1] = 2 \cdot x)$$

Entonces no es co-ce.

$$C_2 = \{x : 1 \in Dom(\Phi_x^{(1)})\}$$

Es un conjunto de índices de la clase de funciones

$$C = \{g(x) : g(1) \downarrow\}$$

No es trivial (trivial ver cual está y cual no). Por teo de Rice no es comp.

Como no es comp entonces no puede ser co-ce y ce.

Veamos si es ce:

$$f(x) = (\exists t)(STP(1, x, t) = 1)$$

Entonces no es co-ce.

$$C_3 = \{x : Dom(\Phi_x^{(1)}) \subseteq \{0, ..., x\}\}$$

Veamos que es co-ce:

$$f(x) = (\exists < t, p >)(STP(p, x, t) = 1 \wedge \ p > x)$$

Veamos la función caract. de C_3:

$$g_3(x) = \begin{cases} 1 & Dom(\Phi_x) \subseteq \{0, ..., x\} \\ 0 & cc \end{cases}$$

$$g_3'(x) = \begin{cases} 1 & g_3(x) \\ \uparrow & \neg g_3(x) \end{cases}$$

Sea e el número de programa de $g_3'(x)$:

$$g_3'(e) \uparrow \Leftrightarrow \neg g_3(x) \Leftrightarrow Dom(\Phi_e) \nsubseteq \{0, ..., x\}$$

FALSO: $Dom(\Phi_e) = \subseteq \{0, ..., x\}$.

$$g_3'(e) = 1 \Leftrightarrow g_3(x) \Leftrightarrow Dom(\Phi_e) \subseteq \{0, ..., x\}$$

FALSO: $Dom(\Phi_e) = \mathbb{N}$.

Entonces no es computable. Entonces no es c.e. (porque es co-ce y no computable).

$$C_4 = \{< x, y >: \forall z \in (Dom(\Phi_x) \cap Dom(\Phi_y))\Phi_x(z) < \Phi_y(z)\}$$

Sospecho que no es un conjunto de índices (es difícil demostrar que no es un conj de índices). Veamos si encontramos un conjunto que lo podamos reducir a este y el cual podamos demostrar fácilmente que es un conjunto de índices.

$$C_4(< x, y >) = \begin{cases} 1 & si\ \forall z \in (Dom(\Phi_x) \cap Dom(\Phi_y))\Phi_x(z) < \Phi_y(z) \\ 0 & cc \end{cases}$$

Si $C_4' \leq C_4$ ("es reducible a"), entonces buscamos una $f(x)$ total tal que:

$$x \in C_4' \Rightarrow f(x) \in C_4$$

Sea e el número de un programa que compute la identidad, entonces tomamos $f(x)$ como:

$$f(< x, y >) =< x, e >$$

Luego

$$C_4'(< x, y >) = C_4(f(< x, y >)) = C_4(< x, e >)$$

$$C_4(< x, e >) = \begin{cases} 1 & si\ \forall z \in Dom(\Phi_x)\ \Phi_x(z) < z \\ 0 & cc \end{cases}$$

Es el conjunto de índices de la clase de funciones:

$$C = \{g(x) : (\forall y)(g(y)\downarrow \Rightarrow \ g(y) < y)\}$$

P1:

```
Y <- 0
```

P2:

```
Y <- X1 + 1
```

$P1 \notin C'_4$ y $P2 \in C'_4$ entonces no es un conjuntos de índices trivial, entonces no es computable.

$$C_5 = \{< x, y, t, i >: (\exists\sigma)SNAP(x, y, t) =< i, \sigma >\}$$

Es computable!

$$C_5(< x, y, t, i >) = \begin{cases} 1 & si\ (\exists\sigma)SNAP(x, y, t) =< i, \sigma > \\ 0 & cc \end{cases}$$

Como siempre va a existir un σ tal que $r(SNAP(x, y, t)) = \sigma$ entonces:

$$C_5(< x, y, t, i >) = \begin{cases} 1 & si\ l(SNAP(x, y, t)) = i \\ 0 & cc \end{cases}$$

$l(x), SNAP(x, y, z)$ y $=$ son funciones pr, entonces C_5 es pr, entonces es computable, entonces es ce y co-ce.

$$C_6 = \{< x, y, i >: (\exists\sigma, t)SNAP(x, y, t) =< i, \sigma >\}$$

Vamos a reducirlo al conjunto de Halt.

$$f(< x, y, i >) =< x, x, |x + 1| >$$

$$C'_6(< x, y, i >) = C_6(f(< x, y, i >) = C_6(< x, x, |x{+}1| >) = \begin{cases} 1 & si\ (\exists\sigma, t)SNAP(x, x, t) =< |x + 1|, \sigma >\equiv (\exists t)SNAP(x, \\ 0 & cc \end{cases}$$

$$= \begin{cases} 1 & si\ Halt(x, x) \\ 0 & cc \end{cases}$$

No es computable, ni co-ce, pero es ce (por ser el conjunto K, visto en teórica).

Ejercicio 13

a) VERDADERO. Solo hace falta ver para la unión entre dos.

Sean A_1 y A_2 dos conjuntos c.e., entonces $A_1 \cup A_2$ es c.e. porque podemos hacer el programa cuya dominio son los valores del conjunto. El programa es:

```
[R] T <- 0
    IF STP(X1, G1, T) = 1 GOTO E
    IF STP(X1, G2, T) = 1 GOTO E
    T <- T + 1
    GOTO R
```

Siendo G1 y G2 las funciones cuyo dominio son los valores de los conjuntos A_1 y A_2 respectivamente.

b) ¿qué es familia de conjuntos c.e.?

c) VERDADERO. Igual que a) pero con el programa:

```
[R] T <- 0
    IF STP(X1, G1, T) != 1 GOTO N
    IF STP(X1, G2, T) = 1 GOTO E
[N] T <- T + 1
    GOTO R
```

d) ¿qué es familia de conjuntos c.e.?

Ejercicio 14

Sea B un conjunto infinito.

a)

Como B es c.e. entonces:

$$B = dom\ g = \{x : (\exists t)(R(x,t)\}$$

Siendo R un pred pr. Podemos escribir entonces el siguiente programa:

```
[R] T <- T+1
    IF R(l(T),r(T)) != 1 GOTO R
    U <- U+1
    IF U != X1 GOTO R
```

Notar que la función no es creciente.

b)

```
[I] B(T) != 1 GOTO R
    C <- C + 1
    IF C != X1 GOTO R
    Y <- T
    GOTO E
[R] T <- T + 1
```

c) Generamos un conjunto computable infinito agarrando la secuencia creciente que genera $f(0), f(1), f(2), ...$ Es decir solo agarro los elementos en las posiciones i cuando

$f(i) \geq f(j)$ $(\forall j)(j < i)$. Este conjunto es infinito, ya que la imagen de f es infinita, y los elementos menores de cualquier $f(x)$ son finitos.

Entonces la función característica va a ser simplemente:

$$B(x) = \begin{cases} 1 & si\ min\ arg(f(i) \geq x) = x \\ 0 & cc \end{cases}$$

Ejercicio 15

Demostrado en teórica.

Ejercicio 16

$$ID = \{x : \Phi_x\ es\ la\ función\ identidad\}$$

Reduzcamos a TOT. Buscamos $f(x)$ tal que:

$$x \in TOT \Rightarrow f(x) \in ID$$

Sea e el número del siguiente programa:

```
Phi(X1,X2) //Programa X2 con entrada X1
Y <- X1
```

Y sea $S_1^1(x, e)$ la función que agarra el programa con número e anterior y esconde el valor de x en X2 (existe por teo. parámetro), entonces:

$$ID(S_1^1(x,e)) = \begin{cases} 1 & si\ \Phi_{S_1^1(x,e)}\ es\ la\ identidad \\ 0 & cc \end{cases}$$

$$= \begin{cases} 1 & si\ \Phi_x\ es\ total \\ 0 & cc \end{cases} = TOT(x)$$

Entonces:

$$f(x) = S_1^1(x,e)$$

$f(x)$ es total y computable. Entonces el conjunto ID no es ce ni co-ce.

$$S = \{x : Im(\Phi_x^{(1)}) = \mathbb{N}\}$$

Sea e el número de programa igual que antes:

$$g(x) = S(S_1^1(x,e)) = \begin{cases} 1 & si\ Im(\Phi^{(1)}_{S_1^1(x,e)}) = \mathbb{N} \\ 0 & cc \end{cases}$$

$$g(x) = \begin{cases} 1 & si\ \Phi_x\ es\ total \\ 0 & cc \end{cases} = TOT(x)$$

Ejercicio 17

$$C = \{f(x) : f \text{ es la función de Ackerman}\}$$

Entonces no existe función $g(x)$ pr tal que:

$$C = Im(g)$$

Entonces siempre existe k tal que:

$$k \in (C \cup Im(f)) \wedge k \notin (C \cap Im(f))$$

Ejercicio 18

a) VERDADERO. Podemos tomar:

$$g(x) = 0$$

Entonces:

$$(f \circ g) = f(0) = k$$

$$(g \circ f) = 0$$

Ambas son pr.

---

b) VERDADERO. Si para todo $x \in C$ vale $\neg Halt(x, x)$ entonces C es el conjunto $\bar{K}$ el cual no es computable.

---

c) VERDADERO. Si C y D son conjuntos c.e. infinitos, entonces $C \cup D$ es c.e. (dem vista en teórica). Entonces por ejercicio 14a, existe $f(x)$ inyectiva y computable tal que $Im(f) = C \cup D$.

---

d) VERDADERO. Como $f(n, x)$ es total y computable para todos los valores de n y x entonces el programa que computa f simplemente se le van a dar un n y un x y los va a computar. El programa de $f(x, y)$ sería:

```
y <- f(X1, X2) //Es computable porque g_x1(x2) es computable
```

---

e) VERDADERO. $f(x)$ es la constante 1 porque la $Im(\Phi_x)$ siempre es un conjunto c.e. La contaste 1 es una función pr.

La demostración de

$$f(x) \text{ parcial computable} \wedge S = \{f(x) : f(x)\downarrow\} \Rightarrow S \text{ es c.e.}$$

se encuentra en el libro "Computability Complexity and Languages Fundamentals of Theoretical Computer Science" página 102. La idea general es ir yendo por todos los valores de $f(x)$ aumentando x buscando el valor que queremos determinar si está en S.

---

f) FALSO. Como e es fijo, entonces $f(x) = 0$ o $f(x) = x + 1$ lo cual es computable.

---

g) FALSO. No existe tal función.

Si existiera, por teo. del punto fijo, tenemos que existe un e tal que:

$$\Phi_e = \Phi_{f(e)}$$