COMPUTABLILIDAD DE CONJUNTOS (II)

Conjuntos computablemente enumerables y reducibilidad de conjuntos: una teoría relajada.

Repasito: definiciones, propiedades y preguntas

Definiciones (todas equivalentes):

A es un conjunto c.e. cuando:

A es el dominio de una función p.c.. Es decir,

$$x \in A \Leftrightarrow f(x)\downarrow$$

(para alguna f p.c.)

Repasito: definiciones, propiedades y preguntas

Definiciones (todas equivalentes):

A es un conjunto c.e. cuando:

(o el rango)

A es la imagen de una función p.c. Es decir,

primitiva recursiva

total computable

$$A = \{f(0), f(1), f(2), \ldots\}$$

(para alguna f p.c.)

primitiva recursiva

total computable

Repasito: definiciones, propiedades y preguntas

A es un conjunto co-c.e. cuando:

su complemento es un conjunto c.e.. Es decir,

$$\bar{A} \text{ es c.e.}$$

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

$$A \text{ computable} \Rightarrow A \text{ c.e.}$$

Idea de la dem:
como puedo decidir computablemente si $x \in A$, pregunto: ¿$x \in A$?. Si me dice que sí, devuelvo algo. Si me dice que no, me cuelgo a propósito. Entonces, tengo una función p.c. definida exactamente para los $x \in A$.

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

$$A \text{ computable} \Rightarrow A \text{ co-c.e.}$$

Idea de la dem: igual que antes.

Sin ensuciarse las manos: recordar que A comp. $\Rightarrow$ $\bar{A}$ comp.

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

A computable $\Leftrightarrow$ A c.e.+co-c.e.

Idea de la dem:

$\Rightarrow$) lo que ya vimos.

$\Leftarrow$) si A y $\bar{A}$ son c.e., para c/u tengo una función p.c. que está definida en sus elementos. Para saber si $x \in A$ o si $x \in \bar{A}$, voy probando de a un paso a ver cuál termina (y alguna de las dos tiene que terminar).

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

$$K = \{x : \Phi(x, x)\downarrow\} \text{ es c.e.}$$

Idea de la dem:

$f(t) = \Phi(t, t)$ es una función p.c. y

$f(x)\downarrow$ sii $x \in K$

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

A c.e. $\Rightarrow$ A computable

Idea de la dem:

(sospecha) ya sé que comp⇒ce, si ce⇒comp, serían equivalentes y no entiendo para qué se molestaron en inventar la definición.

(formal) buscar un contraejemplo, es decir, un conjunto ce que no sea comp.

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

K es co-c.e.

Idea de la dem:
Ya vimos que es ce. Si además fuera co-ce, sería computable (y ya sabemos que no lo es).

Repasito: definiciones, propiedades y preguntas

¿Verdadero o falso?

$$TOT = \{x : \phi_x \text{ es total}\} \text{ es c.e.}$$

computable

co-c.e.

Idea de la dem:

Que no es comp es fácil (Rice)

No ce: ?? y No co-ce: ??????? (ver la teórica)

Ejercicito

p.r.

p.c. total

Dado A un conjunto c.e., y f una función p.c., probar que $f(A) = \{f(x) : x \in A\}$ es c.e..

Idea de la dem (ver la dem. completa y como debe ser en la clase)

p.c. total

Como A es c.e., es la imagen de una función p.c.. Es decir,

p.r.

p.r.

$A = \{g(0), g(1), g(2), \ldots\}$ con g p.c.. Luego, $f(A) = \{f(g(0)), f(g(1)), \ldots\}$, es decir,

p.c. total

p.r. p.r.

$f(A) = \{h(0), h(1), h(2), \ldots\}$ con $h(x) = f(g(x))$. Como f y g son p.c., h es p.c. y

p.c. total p.c. total

p.r.

entonces f(A) es c.e. porque es la imagen de una función p.c..

p.c. total

Ejercicito

Dado A un conjunto c.e., y f una función p.c., probar que $f(A) = \{f(x) : x \in A\}$ es c.e..

Si pedimos un poco más: A computable y f total computable ¿podremos afirmar que f(A) sea computable?

Idea de la dem (ver la dem. completa y como debe ser en la clase)
Vimos que todos los conjuntos c.e. son la imagen de una función p.c., y también de una p.c. total. Es decir, para cualquier cjto B c.e., tenemos que $B = f(N)$ siendo f p.c. total (y N todos los naturales).
Como N es computable, si la afirmación fuera cierta, B sería computable y otra vez no habría diferencia entre c.e. y comp. (y ya sabemos que no es así).

Ejercicito

Dado A un conjunto c.e., y f una función p.c., probar que $f(A) = \{f(x) : x \in A\}$ es c.e..

¿Y si A es co-c.e.? ¿Podemos afirmar que f(A) va a ser co-c.e.?

Idea de la dem (ver la dem. completa y como debe ser en la clase)
Un argumento parecido al anterior: tomamos $A = \mathbb{N}$, que como es comp., es co-ce. Sea B un conjunto ce cualquiera. Entonces, $B = f(\mathbb{N})$ para alguna f p.c.. Si la afirmación fuera cierta, como $\mathbb{N}$ es co-ce y f es p.c., sería B co-ce y, por lo tanto, B sería computable y no existiría la diferencia entre ce y co-ce.

EJERCICIO

Decidir si es c.e., co-c.e. o computable el conjunto:

$$VAC = \{i : W_i = Dom(\Phi_i) \text{ es vacío}\}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

Vamos de lo más fácil de probar a lo más difícil.

1) No es **computable.** Esto es rápido por el **T de Rice.** VAC es el cjto de índices de la clase de funciones que están indefinidas siempre (que no es trivial). (Justificación de tarea)

2) Es **co-ce**: si i no pertenece a VAC, existe un x t.q. $\Phi(x, i)\downarrow$, entonces podemos definir una función p.c. que lo busque: $v(i) = \exists_{\langle x, t\rangle} stp(i, x, t)$. Como v se cuelga si no lo encuentra, tenemos **$v(x)\downarrow$ sii x no está en VAC.** (Justificación de por qué v es p.c. y por qué funciona, de tarea)

3) No puede ser **ce**: ya **vimos que no es comp. y que es co-ce.** Si fuera ce, sería computable.

Ejercicio

Decidir si es c.e., co-c.e. o computable el conjunto:

$$M = \{\min(W_i) : i \in \mathbb{N} \text{ y } W_i \text{ no vacío}\}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)
¡Momento! ¡M son todos los naturales! Para probar que M es ce, co-ce y computable, probemos que $M = \mathbb{N}$.
Ya sabemos que M está incluido en $\mathbb{N}$, veamos $\mathbb{N}$ incluido en M.
Dado un n, veamos que n está en M, es decir, que existe un i, tal que $\min(Dom(\Phi_i)) = n$. Como no nos importa quién es i, basta con mostrar una función p.c. tq el mínimo de su dominio sea n. Por ejemplo

$f_n(x) = \uparrow$ (si $x < n$)
666 (si no).

(Justificación de por qué f_n es p.c. y cumple con las condiciones, de tarea)

Otro ejercicio

Decidir si es computable el conjunto:

$$A = \{ \langle \#P,\ k \rangle : \Psi_P(k)+1 = \Psi_P(k+1) \text{ (y ambos están definidos)} \}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

Sospecha: A no puede ser computable porque si lo fuera, lo sería la función A'(p, k) que decide si P evaluado en k+1 es p evaluado en k, +1. Esto parece mucho, de hecho, si A' fuera comp., entonces sería comp A''(p) = A'(p, 35) que decide si el programa p evaluado en 36 vale 1 más que evaluado en 35. Y, en definitiva, esto define una clase de funciones p.c. no trivial: las funciones que están definidas en 35 y en 36, y evaluadas en 36 valen uno más que evaluadas en 35. Por el T de Rice no es computable decidir esto, y por lo tanto A'', y por lo tanto A' y por lo tanto A no son comps.

SARASA ALERT

Definición: Reducciones entre conjuntos

Dados A y B conjuntos de naturales, decimos que A es reducible a B, y lo notamos $A \leq B$, cuando:

Existe una f computable (total) tal que:

$$x \in A \iff f(x) \in B$$

Es decir, podemos usar una maquinita que decide si las cosas pertenecen a B, para decidir si algo pertenece a A (si previamente lo adaptamos con f).

Reducciones entre conjuntos: Propiedades

Dados A y B conjuntos t.q. $A \leq B$, entonces:

Si B es computable, entonces A es computable.
c.e. c.e.
co-c.e. co-c.e.

Dem de tarea
(háganla, es linda!)

Corolario: Si $A \leq B$, entonces:

Si A **no** es computable, entonces B **no** es computable.
c.e. c.e.
co-c.e. co-c.e.

Otro ejercicio (el mismo de antes)

Decidir si es computable el conjunto:

$$A = \{ \langle \#P,\ k \rangle : \Psi_P(k)+1 = \Psi_P(k+1) \text{ (y ambos están definidos)} \}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

Sin sarasa. Reduzcamos un cjto no computable a A.

O sea, busquemos un A'' no comp. tal que A'' ≤ A.

Proponemos $A'' = \{\#P: \Psi_P(35)+1 = \Psi_P(36)\}$. Por el T de Rice, A'' no es computable (tarea!)

Probemos A'' ≤ A:

Vemos: $\#P \in A''$ sii $<\#P, 35> \in A$.

Podemos tomar $f(\#P) = <\#P, 35>$, que es comp. y total (tarea!).

Otro ejercicio

Decidir si es c.e. o co-c.e. el conjunto:

$$A = \{ \langle \#P, k \rangle : \Psi_P(k)+1 = \Psi_P(k+1) \text{ (y ambos están definidos)} \}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

Por cómo está expresada la propiedad, parece más factible probar que sea **ce**: es decir, dar una función p.c. / programa que, para un programa p y un k, esté definida si se cumple la condición del conjunto e indefinida si no. Sobre todo, porque si p se indefine en k, está bien que la función/prog que armemos se indefina, porque entonces el par <#p, k> no está en el conjunto. (Tarea: mostrar la función o el prog).

Si es ce, no puede ser **co-ce** porque ya vimos que no es comp..

Otro ejercicio

Decidir si es computable, c.e. o co-c.e. el conjunto:

$$B = \{\#P : \Psi_P \text{ es total y estrictamente creciente}\}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

El T de Rice no nos va a dejar que B sea comp.

Si fuera ce, deberíamos mostrar una función que se detenga después de asegurar que P es total y creciente.

Si fuera co-ce, deberíamos mostrar una función que se detenga si detecta que P no es total, o sea, que se indefine en alguna de las entradas.

Nada de esto parece muy razonable. Entonces, parece que tenemos algo que no es ce ni co-ce. Esto tal vez tenga que ver con el hecho de que decidir si algo es total es "complicado".

Otro ejercicio

Decidir si es computable, c.e. o co-c.e. el conjunto:

$$B = \{\#P: \Psi_P \text{ es total y estrictamente creciente}\}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)

Vamos a reducir TOT a B. Luego, como TOT no es ce ni co-ce, B no puede ser ce ni co-ce.

Probemos, entonces, TOT ≤ B. Queremos una f p.c. total tal que

$x \in TOT$ sii Φ_x es total sii $\Phi_{f(x)}$ es total y estr. creciente sii $\Phi_{f(x)} \in B$.

O sea, necesitamos f comp que nos convierta nros de programa, tq:

1) Si el prog original es total, el resultado sea total y creciente.
2) Si el prog original no es total, el resultado sea no total o no crec.

Aprovechamos que podemos elegir y para 2) elegimos que no sea total.

Otro ejercicio

Decidir si es computable, c.e. o co-c.e. el conjunto:

$$B = \{\#P: \Psi_P \text{ es total y estrictamente creciente}\}$$

Idea de la dem (ver la dem. completa y como debe ser en la clase)
Tenemos que convertir nros de programa así que vamos a usar a nuestro amigo: **el T del Parámetro.** Recordamos que queremos que:

1) si el programa es total, el resultado sea total y creciente
2) si el programa no es total, el resultado no sea total.

Idea: Agarrar el programa original, enchufarle al final una función computable, total y creciente, y devolver el resultado de evaluar esta función. (Convencerse de que esto cumple 1 y 2.)

Otro ejercicio

Decidir si es computable, c.e. o co-c.e. el conjunto:

$$B = \{\#P: \Psi_P \text{ es total y estrictamente creciente}\}$$

Sea C p.c., total y estr. creciente.

Convertimos un prog total en uno total y estr. creciente.	Convertimos una función p.c. total en otra total y estr. creciente.
Z1 <- $\Phi_p(x)$ Z1 <- $\Phi(X1, X2)$ Y <- C(x) Y <- C(X1)	$h(x, p) = C(x)$ (si $\Phi_p(x)\downarrow$) $\uparrow$ (si no)

Sea e el nro del programa que mostramos / de un prog que computa h.
$f(p) = S(p, e)$ es la función que nos construye, a partir de p, el programa que queremos. (ver que p está en TOT sii f(p) está en B).

¿Verdadero o Falso?

Sea $f: N \to N$. Si Dom(f) e Im(f) son ambos computables, entonces f es computable.

Idea de la dem:

Si f es total, ya tenemos Dom(f) computable (porque Dom(f) = N).

Si f es acotada, tenemos Im(f) computable (porque Im(f) finito).

Tarea: pensar una tal f que sirva de contraejemplo.

¿Verdadero o Falso?

Si A es un conjunto infinito y no computable, todos sus subconjuntos infinitos son no computables.

Idea de la dem:

Tomamos A un cjto de índices no trivial. Ej: A = los números de todos los programas que computan funciones totales y constantes.

Tomamos un cjto S de nros de programa que dependa de la forma que tienen los programas y que cumplan la propiedad que define a A. Ej: S = los nros de los programas compuestos solo por instrucciones de la forma Y <- Y+1.

Con Rice podemos probar que A no es comp.. En gral decidir sobre la forma de los programas es p.r., entonces S es comp.. Queda probar $S \subseteq A$.