Lógica y Computabilidad

2do cuatrimestre 2020 - **A DISTANCIA**

Departamento de Computación - FCEyN - UBA

Computabilidad - clase 7

Conjuntos c.e., teorema de la enumeración, teorema de Rice y aplicaciones

Conjuntos en teoría de la computabilidad

Cuando hablamos de un conjunto de naturales A pensamos siempre en la función característica de ese conjunto.

$$A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si no} \end{cases}$$

Así, un conjunto puede ser:

- computable
- primitivo recursivo

Teorema

Sean A, B conjuntos de una clase PRC $\mathcal{C}$. Entonces $A \cup B$, $A \cap B$ y $\overline{A}$ están en $\mathcal{C}$.

Conjuntos computablemente enumerables

Igual que con las funciones

- hay conjuntos computables, por ejemplo

$$\emptyset \quad , \quad \mathbb{N} \quad , \quad \{p : p \text{ es primo}\}$$

- hay conjuntos no computables, por ejemplo

$$\{\langle x, y\rangle : \mathrm{HALT}(x, y)\} \quad , \quad \{\langle x, \langle y, z\rangle\rangle : \Phi_x(y) = z\}$$

Definición

Un conjunto A es computablemente enumerable (c.e.) cuando existe una función parcial computable $g : \mathbb{N} \to \mathbb{N}$ tal que

$$A = \{x : g(x) \downarrow\} = \mathrm{dom}\ g$$

- podemos decidir algoritmicamente si un elemento sí pertenece a A, pero para elementos que no pertenecen a A, el algoritmo se indefine
- se llaman algoritmos de semi-decisión: resuelven una aproximación al problema de decidir la pertenencia de un elemento al conjunto A

Propiedades de los conjuntos c.e.

Un conjunto A es co-c.e. si $\overline{A}$ es c.e.

Teorema

Si A es computable entonces A es c.e.

Demostración.

Sea P_A un programa para [la función característica de] A.
Consideremos el siguiente programa P:

$$[C] \qquad \text{IF } P_A(X) = 0 \text{ GOTO } C$$

Tenemos

$$\Psi_P(x) = \begin{cases} 0 & \text{si } x \in A \\ \uparrow & \text{si no} \end{cases}$$

y por lo tanto

$$A = \{x : \Psi_P(x)\downarrow\}$$

$\square$

Propiedades de los conjuntos c.e.

Teorema

Si A y B son c.e. entonces $A \cup B$ y $A \cap B$ también son c.e.

Demostración.

Sean $\quad A = \{x : \Phi_p(x)\downarrow\} \quad , \quad B = \{x : \Phi_q(x)\downarrow\}$

$(A \cap B)$ El siguiente programa R tiene como dominio a $A \cap B$:

$$Y \leftarrow \Phi_p(x)$$
$$Y \leftarrow \Phi_q(x)$$

En efecto, $\Psi_R(x)\downarrow$ sii $\Phi_p(x)\downarrow$ y $\Phi_q(x)\downarrow$.

$(A \cup B)$ El siguiente programa R' tiene como dominio a $A \cup B$:

$$\begin{array}{ll} [C] & \text{IF } \mathrm{STP}^{(1)}(X, p, T) = 1 \text{ GOTO } E \\ & \text{IF } \mathrm{STP}^{(1)}(X, q, T) = 1 \text{ GOTO } E \\ & T \leftarrow T + 1 \\ & \text{GOTO } C \end{array}$$

En efecto, $\Psi_{R'}(x)\downarrow$ sii $\Phi_p(x)\downarrow$ o $\Phi_q(x)\downarrow$.

Propiedades de los conjuntos c.e.

Teorema

A es computable sii A y $\overline{A}$ son c.e.

Demostración.

($\Rightarrow$) si A es computable entonces $\overline{A}$ es computable

($\Leftarrow$) supongamos que A y $\overline{A}$ son c.e.

$$A = \{x : \Phi_p(x)\downarrow\} \qquad , \qquad \overline{A} = \{x : \Phi_q(x)\downarrow\}$$

Consideremos P:

$$\begin{array}{ll}
[C] & \text{IF STP}^{(1)}(X, p, T) = 1 \text{ GOTO } F \\
 & \text{IF STP}^{(1)}(X, q, T) = 1 \text{ GOTO } E \\
 & T \leftarrow T + 1 \\
 & \text{GOTO } C \\
[F] & Y \leftarrow 1
\end{array}$$

Para cada x, $x \in A$ o bien $x \in \overline{A}$. Entonces Ψ_P computa A.$\square$

Teorema de la enumeración

Definimos

$$W_n = \{x : \Phi_n(x)\downarrow\} = \text{dominio del } n\text{-ésimo programa}$$

Teorema

Un conjunto A es c.e. sii existe un n tal que $A = W_n$.

Existe una enumeración de todos los conjuntos c.e.

$$W_0, W_1, W_2, \ldots$$

Problema de la detención (visto como conjunto)

Recordar que

$$W_n = \{x : \Phi_n(x) \downarrow\}$$

Definimos

$$K = \{n : n \in W_n\}$$

Observar que

$$n \in W_n \qquad \text{sii} \qquad \Phi_n(n) \downarrow \qquad \text{sii} \qquad \mathrm{HALT}(n, n)$$

Teorema

K es c.e. pero no computable.

Demostración.

- la función $n \mapsto \Phi(n, n)$ es parcial computable, de modo que K es c.e.
- supongamos que K fuera computable. Entonces $\overline{K}$ también lo sería. Luego existe un e tal que $\overline{K} = W_e$. Por lo tanto

$$e \in K \qquad \text{sii} \qquad e \in W_e \qquad \text{sii} \qquad e \in \overline{K}$$

$\square$

Más propiedades de los conjuntos c.e.

Teorema

Si A es c.e., existe un predicado p.r. $R : \mathbb{N}^2 \to \mathbb{N}$ tal que

$$A = \{x : (\exists t)\ R(x,t)\}$$

Demostración.

Sea $A = W_e$. Es decir,

$$A = \{x : \Phi_e(x) \downarrow\}.$$

Entonces $x \in A$ cuando en algún tiempo t, el programa e con entrada x termina, i.e.

$$A = \{x : (\exists t)\ \underbrace{\mathrm{STP}^{(1)}(x,e,t)}_{R(x,t)}\}$$

$\square$

Más propiedades de los conjuntos c.e.

Teorema

Si $A \neq \emptyset$ es c.e., existe una función p.r. $f : \mathbb{N} \to \mathbb{N}$ tal que

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

Demostración.

Por el teorema anterior, existe P p.r. tal que

$$A = \{x : (\exists t)\ P(x, t)\}.$$

Sea $a \in A$ y definamos

$$f(u) = \begin{cases} l(u) & \text{si } P(l(u), r(u)) \\ a & \text{si no} \end{cases}$$

- $x \in A \Rightarrow$ existe t tal que $P(x, t) \Rightarrow f(\langle x, t\rangle) = x$
- sea x tal que $f(u) = x$ para algún u. Entonces $x = a$ o bien u es de la forma $u = \langle x, t\rangle$, con $P(x, t)$. Luego $x \in A$. $\square$

Más propiedades de los conjuntos c.e.

Teorema

Si $f : \mathbb{N} \to \mathbb{N}$ *es parcial computable,* $A = \{f(x) : f(x) \downarrow\}$ *es c.e.*

Demostración.

Sea $\Phi_p = f$. Definamos el programa Q

$$
\begin{array}{ll}
[A] & \text{IF } \mathrm{STP}^{(1)}(Z, p, T) = 0 \text{ GOTO } B \\
 & \text{IF } \Phi_p(Z) = X \text{ GOTO } E \\
[B] & Z \leftarrow Z + 1 \\
 & \text{IF } Z \leq T \text{ GOTO } A \\
 & T \leftarrow T + 1 \\
 & Z \leftarrow 0 \\
 & \text{GOTO } A
\end{array}
$$

Notar que $\Psi_Q(X) \downarrow$ si existen Z, T tal que

- $Z \leq T$
- $\mathrm{STP}^{(1)}(Z, p, T)$ es verdadero (i.e. el programa para f termina en T o menos pasos con entrada Z)
- $X = f(Z)$

$$
\Psi_Q(x) = \begin{cases} 0 & \text{si } x \in A \\ \uparrow & \text{si no} \end{cases}
$$

Luego A es c.e. $\square$

Caracterizaciones de los conjuntos c.e.

Teorema

Si $A \neq \emptyset$, son equivalentes:

1. *A es c.e.*
2. *A es el rango de una función primitiva recursiva*
3. *A es el rango de una función computable*
4. *A es el rango de una función parcial computable*

Demostración.

$(1 \Rightarrow 2)$ *Teorema de hoja 10*

$(2 \Rightarrow 3)$ *Trivial*

$(3 \Rightarrow 4)$ *Trivial*

$(4 \Rightarrow 1)$ *Teorema de hoja 11*

$\square$

Teorema de Rice

$A \subseteq \mathbb{N}$ es un conjunto de índices si existe una clase de funciones $\mathbb{N} \to \mathbb{N}$ parciales computables $\mathcal{C}$ tal que $A = \{x : \Phi_x \in \mathcal{C}\}$

Teorema

Si A es un conjunto de índices tal que $\emptyset \neq A \neq \mathbb{N}$, A no es computable.

Demostración.

Supongamos $\mathcal{C}$ tal que $A = \{x : \Phi_x \in \mathcal{C}\}$ computable. Sean $f \in \mathcal{C}$ y $g \notin \mathcal{C}$ funciones parciales computables.
Sea $h : \mathbb{N}^2 \to \mathbb{N}$ la siguiente función parcial computable:

$$h(t, x) = \begin{cases} g(x) & \text{si } t \in A \\ f(x) & \text{si no} \end{cases}$$

Por el Teorema de la Recursión, existe e tal que $\Phi_e(x) = h(e, x)$.

- $e \in A \Rightarrow \Phi_e = g \Rightarrow \Phi_e \notin \mathcal{C} \Rightarrow e \notin A$
- $e \notin A \Rightarrow \Phi_e = f \Rightarrow \Phi_e \in \mathcal{C} \Rightarrow e \in A$

□

Aplicaciones del Teorema de Rice

El teorema da una fuente de conjuntos no computables:

- $\{x : \Phi_x \text{ es total}\}$
- $\{x : \Phi_x \text{ es creciente}\}$
- $\{x : \Phi_x \text{ tiene dominio infinito}\}$
- $\{x : \Phi_x \text{ es primitiva recursiva}\}$

¡Todos son no computables porque todos son conjuntos de índices no triviales!

Ejemplos de conjuntos no c.e.

- $\overline{K} = \{x : \Phi_x(x)\uparrow\}$ no es c.e.
  - K es c.e. de modo que si $\overline{K}$ lo fuera, K sería computable
- $Tot = \{x : \Phi_x \text{ es total}\}$ no es c.e.:
  - es una diagonalización simple. Supongamos que Tot es c.e.
  - existe f computable tal que $Tot = \{f(0), f(1), f(2), \dots\}$
  - entonces existe e tal que $\Phi_e(x) = \Phi_{f(x)}(x) + 1$
  - como Φ_e es total, $e \in Tot$. De modo que existe u tal que $f(u) = e$
  - $\Phi_{f(u)}(x) = \Phi_{f(x)}(x) + 1$. Absurdo para $x = u$.
- $\overline{Tot} = \{x : \Phi_x \text{ no es total}\}$ no es c.e.
  - parecido a lo que hicimos la clase pasada. Supongamos $\overline{Tot}$ c.e.
  - existe d tal que $\overline{Tot} = \text{dom } \Phi_d$
  - definimos el siguiente programa P:

$$
\begin{array}{ll}
[C] & \text{IF STP}^{(1)}(X, d, T) = 1 \text{ GOTO } E \\
 & T \leftarrow T + 1 \\
 & \text{GOTO } C
\end{array}
$$

  - sigue igual a lo que vimos la clase pasada

$$\Psi_P^{(2)}(x, y) = g(x, y) = \begin{cases} \uparrow & \Phi_x \text{ es total} \\ 0 & \text{si no} \end{cases}$$

Conjuntos más difíciles que el halting problem

- $K = \{x : \Phi_x(x) \downarrow\}$ no es computable
  - pero K es c.e.
- $Tot = \{x : \Phi_x \text{ es total}\}$ no es computable
  - Tot no es c.e.
  - $\overline{Tot}$ no es c.e.
- de alguna forma, Tot es más difícil que K
  - esto se formaliza dentro de la teoría
  - no hay tiempo para verlo en esta materia
  - pero lo estudiamos en Teoría de la Computabilidad
    - se suele dar los primeros cuatrimestres
    - da 3 puntos como optativa para la licenciatura y se cursa 1 vez por semana
    - http://www.glyc.dc.uba.ar/teocomp/