Lenguajes Formales, Autómatas y Computabilidad

Clase Teórica
Otras funciones no computables

Segundo Cuatrimestre 2025

Funciones de Kolmogorov

Definimos $C : \mathbb{N} \to \mathbb{N}$,

$$C(x) = \min_{\#P}(\Psi_P^{(0)} = x)$$

Por lo tanto $C(x)$ es el menor código de programa que computa x sin usar input.

Teorema

La función C es no es total computable.

Demostración de que C no es total computable

Supongamos C es una función total computabley S es un programa que la computa. Consideremos el siguiente programa Q, que no usa entrada:

$\alpha \leftarrow$ constante igual al codigo de este programa (o mayor si preferis)
$Z \leftarrow 0$
while $(S(Z) \leq \alpha)$
    $Z++$
}
$Y \leftarrow Z$

Tenemos que $\#Q$ es el índice de Q, y quirurgicamente elegimos α de manera tal que

$$\alpha = \#Q.$$

Por un lado, el programa Q sin entrada imprime Y garantizando

$$C(Y) > \alpha = \#Q.$$

Pero por otro lado Y es la salida de Q entonces

$$C(Y) \leq \#Q$$

Uniendo las desigualdades $\#Q < C(Y) \leq \#Q$ Absurdo.
Provino de suponer que C es total computable.
Concluimos que la función C no es total computable. $\square$

La equivalencia entre programas no es computable

Definimos $f : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ como

$$f(x,y) = 1 \text{ si y solo si } \phi_x^{(1)} = \phi_y^{(1)}$$

donde $\phi_x^{(1)} = \phi_y^{(1)}$ significa igualdad extensional:
$\forall n \left(\phi_x^{(1)}(n)\uparrow \wedge \phi_y^{(1)}(n)\uparrow\right)$ ó $\left(\phi_x^{(1)}(n) = \phi_y^{(1)}(n)\right)$.
Es decir,

$$\forall n \left(\forall t\ STP^{(1)}(n,x,t) \text{ falso } \wedge STP^{(1)}(n,y,t) \text{ falso })\right) \text{ ó}$$
$$\left(\exists t\ STP^{(1)}(n,x,t)\text{verdadero } STP^{(1)}(n,y,t) \text{ verdadero y dan igual valor}\right)$$

Teorema

La funcíon $f : \mathbb{N} \times \mathbb{N} \to \{0,1\}$, $f(x,y) = 1$ si y solo si $\phi_x^{(1)} = \phi_y^{(1)}$, no es total computable.

Demostración

Sea z el codigo tal que $\forall n$, $\phi_z^{(1)}(n) = 0$,
Sea u el codigo tal que $\forall n$, $\phi_u^{(1)}(n) \uparrow$.
Sea función total computable $r : \mathbb{N} \to \mathbb{N}$ tal que dado x genera el codigo $r(x)$ que satisface

si $\phi_x^{(1)}(0) \downarrow$, entonces $\phi_{r(x)}^{(1)} = \phi_z^{(1)}$, la funcion constante 0

si $\phi_x^{(1)}(0) \uparrow$, entonces $\phi_{r(x)}^{(1)} = \phi_u^{(1)}$, la función siempre diverge.

Es decir, dado x, $r(x)$ es el código de un programa que ignora su entrada, simula $\phi_x^{(1)}(0)$, y si termina devuelve 0 para todo n; si no, diverge para todo n.

Supongamos, por contradicción, que f es computable.
Pero
$f(r(x), z) = 1$ si y solo si $\phi_{r(x)}^{(1)} = \phi_z^{(1)}$, lo que ocurre si y solo si $\phi_x^{(1)}(0) \downarrow$.
Por lo tanto, podríamos decidir si $\phi_x^{(1)}(0) \downarrow$, lo que contradice la indecidibilidad del problema de la detención de todos los programas para entrada 0.
Luego, f no es computable.

Lenguajes Formales, Autómatas y Computabilidad

Clase Teórica
Conjuntos computablemente enumerables

Segundo Cuatrimestre 2025

Bibliografía

Introduction to Automata Theory, Languages and Computation, J. Hopcroft, R. Motwani, J. Ullman, Second Edition, Addison Wesley, 2001. Capítulo 8 y 9.

Computability, Complexity, and Languages: Fundamentals of Theoretical ... By Martin Davis, Ron Sigal, Elaine J. Weyuker, Second Ediiton, Morgan Kaufmann, 1994

Tres formas de usar una MT

1. Aceptadoras de conjuntos en $(\Sigma \setminus \{\mathbf{B}\})^*$
El lenguaje $L \subseteq (\Sigma \setminus \{\mathbf{B}\})^*$ aceptado por una MT $\mathcal{M}$ es el conjunto de palabras de entrada $w \in (\Sigma \setminus \{\mathbf{B}\})^*$ que causan que $\mathcal{M}$ llegue al estado final,

$$L = \mathcal{L}(\mathcal{M}) = \{w \in (\Sigma \setminus \{\mathbf{B}\})^* : q_0 w \overset{*}{\underset{\mathcal{M}}{\vdash}} \alpha q_f \beta, \text{ donde } \alpha\beta \in \Sigma^*\}$$

2. Calculadoras de funciones parciales computables

Luego, $L \subseteq (\Sigma \setminus \{\mathbf{B}\})^*$ es c.e. si representa el dominio de $f : \mathbb{N} \to \mathbb{N}$ parcial computable, es decir, hay MT $\mathcal{M}$ que computa f y

$$L \text{ codifica el conjunto } \{x \in \mathbb{N} : f(x) \downarrow\}.$$

3. Enumeradoras de conjuntos en $(\Sigma \setminus \{\mathbf{B}\})^*$ **no vacíos**
L representa la imagen de una funcion $f : \mathbb{N} \to \mathbb{N}$ totalmente computable, es decir hay una MT $\mathcal{M}$ que computa f y

$$L \text{ codifica el conjunto } \{y \in \mathbb{N} : \exists x \in \mathbb{N}, f(x) = y\}.$$

Conjuntos computablemente enumerables

Dado un alfabeto Σ hay una correspondencia uno a uno entre $\mathbb{N}$ y Σ^*.
Las definciones que siguen las podemos dar en $\mathbb{N}$ o en Σ^*.

Definición (Conjunto c.e.)

Un conjunto $A \subseteq \mathbb{N}$ es computablemente enumerable, que abreviamos c.e., si hay una MT $\mathcal{M}$ tal que $\mathcal{L}(M) = L$.

Ejemplos:
$\{\langle x, y\rangle : \mathsf{HALT}(x, y)\}$
$\{\langle x, \langle y, z\rangle\rangle : \phi_x(y) = z\}$

Teorema (Caracterización de conjuntos c.e.)

Son equivalentes :

1. $A \subseteq \mathbb{N}$ *es c.e (hay MT* M *tal que* $\mathcal{L}(M) = A$*.)*
2. *Existe función* $f : \mathbb{N} \to \mathbb{N}$ *parcial computable,* $A = dom(f) = \{x : f(x)\downarrow\}$*.*
3. A *es vacío o existe función* $g : \mathbb{N} \to \mathbb{N}$ *total computable,* $A = imagen(g)$*.*

Demostración de teorema

(1 $\Leftrightarrow$ 2.) El lenguaje aceptado por una máquina de Turing $\mathcal{M} = (Q, \Sigma, \delta, q_0, q_f)$ es el conjunto de palabras $w \in (\Sigma \setminus \{\mathbf{B}\})^*$ que causan que $\mathcal{M}$ llegue al estado final,

$$\mathcal{L}(\mathcal{M}) = \{w \in (\Sigma \setminus \{\mathbf{B}\})^* : q_0 w \overset{*}{\underset{\mathcal{M}}{\vdash}} \alpha q_f \beta, \text{ donde } \alpha\beta \in \Sigma^*\}$$

Sea $A \subseteq \mathbb{N}$ el conjunto aceptado por $\mathcal{M}$. Sea $f : \mathbb{N} \to \mathbb{N}$ la función parcial computble calculada por $\mathcal{M}$:

$$w \in A \Leftrightarrow f(w) = \alpha \Leftrightarrow q_0 w \overset{*}{\underset{\mathcal{M}}{\vdash}} \alpha q_f \beta \Leftrightarrow f(w) \downarrow .$$

Un lema intermedio

Lema

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

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

Demostración.

Asumamos $A = \{x : \phi_e(x) \downarrow\}$. Entonces $x \in A$ cuando en algún tiempo t, el programa e con entrada x termina,

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

$\square$

Demostración del teorema, continuación

($2 \Rightarrow 3$). Si A es vacío vale la implicación. Asumimos A es c.e. no vacío. Por el lema, existe predicado computable R tal que

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

Sea $a \in A$ y sea $g : \mathbb{N} \to \mathbb{N}$ total computable tal que

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

Debemos ver que $A = \{g(0), g(1), g(2), \dots\}$. Veamos la inclusión de uno en otro.

Si $x \in A$ entonces existe t tal que $R(x,t)$. Por lo tanto $x = g(\langle x, t\rangle)$.

Si $x = g(u)$ para algún u entonces o bien, $x = a$, o bien existe t tal que $u = \langle x, t\rangle$ y $R(x,t)$. En ambos casos $x \in A$.

Concluimos que A es la imagen de la función total computable g.

Demostración, continuación

$(3 \Rightarrow 2)$ Sea $g : \mathbb{N} \to \mathbb{N}$ total computable tal que $A = imagen(g)$.
Definimos $f : \mathbb{N} \to \mathbb{N}$ parcial computable. Para cada $s \in \mathbb{N}$, $f(g(s)) = 0$.
Luego,
$x \in \ imagen(g) \Leftrightarrow \exists s \ g(s) = x \Leftrightarrow f(x) = 0 \Leftrightarrow x \in dominio(f)$.
$\square$

Conjuntos computables

Definition (Conjunto computable)

Un conjunto $A \subseteq \mathbb{N}$ es computable si existe una función total computable $f : \mathbb{N} \to \{0, 1\}$ tal que para todo x, $f(x) = 1 \Leftrightarrow x \in A$.

Ejemplos:
$\{p : p$ es primo$\}$
$\{x : x$ es par$\}$

Propiedades de los conjuntos computables y los conjuntos c.e.

Notación: $\overline{A} = \mathbb{N} \setminus A$.
Notación: 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 ($P_A(X) = 1$ si $X \in A$, y $P_A(X) = 0$ si $x \notin A$).

Consideremos el siguiente programa Q con entrada X: **while** ($P_A(X) = 0$) **do** $\{$ **pass** $\}$
$Y \leftarrow 1$

Tenemos

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

y por lo tanto

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

□

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:
while (STP$^{(1)}(X,p,T) = 0$ and STP$^{(1)}(X,q,T) = 0$){
    $T++$
}
$Y \leftarrow$ STP$^{(1)}(X,p,T)$
Para cada x, $x \in A$ o bien $x \in \overline{A}$. Entonces Ψ_P computa la función carecteristica de A.

$\square$

Problema de la detención (visto como conjunto)

Definimos

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

Definimos

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

Observar que

$$n \in W_n \qquad \text{sii} \qquad \phi_n(n)\downarrow \qquad \text{sii} \qquad \mathsf{HALT}(n,n)$$

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 conjunto computable. Entonces $\overline{K}$ también lo sería. Luego existe un e tal que $\overline{K} = W_e$. Por lo tanto

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

Absurdo $\square$

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 $A \cup B$:

while ($STP(X, p, T) = 0$ and $STP(X, q, T) = 0$) **do** $\{T++\}$

En efecto, $\Psi_{R'}(x)\downarrow$ sii $\phi_p(x)\downarrow$ o $\phi_q(x)\downarrow$. $\square$

Teorema

Todo conjunto infinito c.e. contiene un conjunto infinito computable.

Demostración.

Sea $A = imagen(f)$ donde $f : \mathbb{N} \to \mathbb{N}$ total computable.
Definimos $g : \mathbb{N} \to \mathbb{N}$ total computable
$g(0) = f(0)$, y para $x > 0$,
$g(x) = f(w)$ donde $w = \min_z (f(z) > g(x-1))$.
Sea $B = imagen(g)$. Veamos que B es computable.
Notemos que $B = \{g(1), g(2), \ldots\}$ está ordenado de menor a mayor:

$$g(1) < g(2) < g(3) < \ldots$$

Sea G un programa con una entrada que computa g (es decir, $\Psi_G^{(1)} = g$).
El siguiente programa Q con una entrada computa la función característica de B

$Z \leftarrow 0$
while $(G(Z) < X)$ $\{$
$Z{+}{+}$
$\}$
$Y \leftarrow (G(Z) = X)$ (funcion booleana de comparacion)

$\Psi_Q(x) = 1$ implica $x \in imagen(g)$ implica $x \in B$.
$\Psi_Q(x) = 0$ implica $\exists z\ (\Psi_G(z-1) < x < \Psi_G(z))$
implica $x \notin imagen(g)$ implica $x \notin B$.

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.

Otro conjunto no c.e. : El conjunto de índices Elegantes

Ya definimos $C : \mathbb{N} \to \mathbb{N}$,

$$C(x) = \min_{\#P}(\Psi_P^{(0)} = x),$$

es menor el índice de programa P que computa computa x.

$$Elegante = imagen(C)$$

es el conjunto de códigos más chicos que computan valores distintos.

Teorema

Elegante no es c.e .

Demostración: Elegante no es c.e.

Supongamos Elegante es c.e.. Entonces es la imagen de una función total, y se enumera con un programa F. Entonces hay un programa Q:

$\alpha \leftarrow$ constante igual o mayor que el codigo de este programa
$Z \leftarrow 0$
while $(F(Z)) \leq \alpha)${
 $Z++$
}
$Y \leftarrow \Psi_R^{(0)}$ (donde $\#R = F(Z)$)

Entonces Y es el ouput del programa elegante $F(Z)$ con $F(Z) > \alpha = \#Q$. El programa Q garantiza

$$C(Y) = F(Z) > \#Q$$

Pero obtuvimos Y usando Q, asi que

$$C(Y) \leq \#Q$$

Entonces

$$\#Q < C(Y) \leq \#Q$$

Absurdo. Por suponer que Elegante es c.e. $\square$

Ejemplos de conjuntos no c.e.

Teorema

$Tot = \{x : \phi_x$ es total$\}$ no es c.e.:

Demostración.

Supongamos que Tot es c.e..
Existe f computable total que $Tot = \{f(0), f(1), f(2), \dots\}$

Entonces también 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 $f(u) = e$

$\phi_{f(u)}(x) = \phi_e(x) = \phi_{f(x)}(x) + 1$. Absurdo para $x = u$.

□

Ejemplos de conjuntos no c.e.

$\overline{Tot} = \{x : \phi_x$ no es total computable$\}$ no es c.e.

Demostración.

Supongamos que $\overline{Tot}$ es c.e. Existe f total computable tal que $\overline{Tot} = \{f(0), f(1), f(2), \dots\}$

Sea $P_{e,x}(y) = \begin{cases} 0 & \text{si } \phi_e(x)\downarrow \\ \uparrow, & \text{en caso contrario} \end{cases}$

Sea $i_{e,x}$ el codigo de programa de $P_{e,x}$.

Si $i_{e,x}$ en imagen de f entonces $P_{e,x}$ es no funcion total (esta idefinida siempre) y $\phi_e(x)\uparrow$

Si $i_{e,x}$ no esta en imagen de f entonces $P_{e,x}$ es funcion total (constante 0) y $\phi_e(x)\downarrow$

Luego conjunto $\overline{\mathsf{HALT}} = \{\langle e, x\rangle : \phi_e(x)\uparrow\}$ es c.e. Absurdo

$\square$

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.

Para dar esta demostración necesitamos otros teoremas de computabilidad. Queda para la clase que viene.

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!

Conjuntos no computables

- $K = \{x : \phi_x(x) \downarrow\}$ no es computable pero K es c.e.
- $Tot = \{x : \phi_x$ es total$\}$ no es computable y Tot no es c.e.
- $\overline{Tot} = \{x : \phi_x$ no es total$\}$ no es c.e.
- $Elegantes$ es el conjunto $\left\{x : \phi_x^{(0)} \downarrow \text{y} \left(\text{ para todo } z < x\big(\phi_z^{(0)} \uparrow \vee \phi_z^{(0)} \neq \phi_x^{(0)}\big)\right)\right\}$ no es c.e.