# Lenguajes Formales, Autómatas y Computabilidad

Clase Teórica
Conjuntos computablemente enumerables

Primer 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 $f : \mathbb{N} \to \mathbb{N}$ parcialmente 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** $(\Sigma \setminus \{\mathbf{B}\})^*$
$L$ 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 $\Sigma$ hay una correspondencia uno a uno entre $\mathbb{N}$ y $\Sigma^*$.
Las definciones que siguen las podemos dar en $\mathbb{N}$ o en $\Sigma^*$.

## 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.*
2. $A$ *es vacío o 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,

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

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

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

# 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

($2 \Rightarrow 3$). Asumimos $A$ es c.e. Por el lema, existe predicado computable $P$ tal que

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

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

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

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 $P(x,t)$. 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 $P(x,t)$. 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 talk 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 $A$. Consideremos el siguiente programa $P$:

$$\textbf{while } (P_A(X) = 0) \textbf{ do } \{ \textbf{ pass } \}$$

Tenemos

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

y por lo tanto

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

□

# 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 $\Psi_P$ computa $A$.

□

# 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 □

# 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 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.

$$W_0, W_1, W_2, \dots$$

# 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 $\Phi_q = f$ y el programa $Q$ donde $\Psi_Q(X)\downarrow$ si existen $Z, T$ tales que $Z \leq T$, $\mathrm{STP}^{(1)}(Z, q, T)$ es verdadero, y $X = f(Z)$

$Z_1 \leftarrow 0$
**while** $(not Z_1)$**do** {
$Z \leftarrow 0$
$T \leftarrow 0$
**while** $(\mathrm{STP}^{(1)}(Z, q, T) = 0)$**do**
**if** $(Z < T)\{\ Z++\ \}$
**else** $\{T++, Z \leftarrow 0\}$
$Z_1 \leftarrow (\Phi_p(Z) = X)$
}

$\Psi_Q(x) = 0$ si $x \in A$, y $\Psi_Q(x)\uparrow$ si no. Luego $A$ es c.e.. □

## Teorema

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

## Demostración.

Sea $A = imagen(f)$ donde $f : \mathbb{N} \to \mathbb{N}$ totla computable.
Defnimos $g : \mathbb{N} \to \mathbb{N}$ 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 $B$ está ordenado de menor a mayor. Si el programa $P$ computa $g$, el siguiente programa la computa la función característica de $B$

$$
\begin{array}{l}
Z \leftarrow 0 \\
\textbf{while } (\Psi_P^{(1)}(Z) < X) \ \{ \\
\quad Z++ \\
\} \\
Y \leftarrow (\Psi_P^{(1)}(Z) = X)
\end{array}
$$

$\square$

# 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.

# Funciones de Kolmogorov

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

$$C(x) = \min_{w}(\Psi_w^{(0)} = x)$$

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

## Teorema

*La función $C$ es total y no computable.*

# Demostración de que $C$ no es computable

Supongamos $P$ es un programa que computa $C$.
Consideremos el siguiente programa $Q$:

$\alpha \leftarrow$ constante mas grande que el codigo de este programa
$Z \leftarrow 0$
**while** $(\Psi_P^{(1)}(Z) \leq \alpha)$
    $Z++$
}
$Y \leftarrow Z$

Tenemos que $\#Q$ es el numero de Gödel de la secuencia de instrucciones de $Q$, y quirurgicamente elegimos $\alpha$ de manera tal que

$$\alpha > \#Q.$$

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

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

Pero por otro lado $Y$ es la salida de $Q$ etonces

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

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

# Un conjunto Elegante

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

$$C(x) = \min_{w}(\Psi_w^{(0)} = x),$$

es el menor código de programa 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 $P$. Entonces hay un programa $Q$:

$\alpha \leftarrow$ constante igual o mayor que el codigo de este programa
$Z \leftarrow 0$
**while** $(\Psi_P^{(1)}(Z)) \leq \alpha)\{$
$\quad Z++$
$\}$
$Y \leftarrow \Psi^{(0)}_{\Psi_P^{(1)}(Z)}$

Como en la diapositiva anterior, el programa $Q$ garantiza

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

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.

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

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 $\Phi_e$ 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 $x = u$. $\square$

# Ejemplos de conjuntos no c.e.

$\overline{Tot} = \{x : \Phi_x$ no es total$\}$ no es c.e.
Supongamos que $\overline{Tot}$ es c.e. Existe $f$ 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 $\overline{\text{HALT}} = \{\langle e, x\rangle : \phi_e(x)\uparrow\}$ es c.e. Absurdo

# 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 $\Phi_e(x) = h(e,x)$.

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

# 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.
- $Tot = \{x : \Phi_x \text{ es total}\}$ no es computable y $Tot$ no es c.e.
- $\overline{Tot} = \{x : \Phi_x \text{ no es total}\}$ 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.