

4.1. Indecidibilidad y reducciones

Teorema. Sea

$$A_{\text{MT}} = \{\langle M, w \rangle : M \text{ es una MT y acepta } w\}.$$

A_{MT} es reconocible, pero indecidible.

Teorema. Sea

$$\text{HALT}_{\text{MT}} = \{\langle M, w \rangle : M \text{ es una MT y se detiene sobre } w\}.$$

HALT_{MT} es reconocible, pero decidable.

Teorema. Sea

$$E_{\text{MT}} = \{\langle M \rangle : M \text{ es una MT y } \mathcal{L}(M) = \emptyset\}.$$

E_{MT} no es decidable.

Teorema.

$$\text{REGULAR}_{\text{MT}} = \{\langle M \rangle : M \text{ es una MT y } \mathcal{L}(M) \text{ es regular}\}.$$

$\text{REGULAR}_{\text{MT}}$ no es decidable.

Teorema.

$$\text{EQ}_{\text{MT}} = \{\langle M_1, M_2 \rangle : M_1, M_2 \text{ son MT y } \mathcal{L}(M_1) = \mathcal{L}(M_2)\}.$$

EQ_{MT} no es decidable.

Teorema. Sea L un lenguaje. L es decidable si y sólo si L y L^c son reconocibles.

Corolario. A_{MT}^c y $\text{HALT}_{\text{MT}}^c$ no son reconocibles.

Definición. Un *historial de computación que acepta* de M sobre w es una secuencia finita $C_0, \dots, C_t$, con $t \geq 0$, tal que:

- C_0 es la configuración inicial de M sobre w .
- $C_i \vdash_M C_{i+1}$ para todo $0 \leq i < t$.
- C_t es una configuración de aceptación.

Definición. Una *MT linealmente acotada* (LBA) no mueve el cabezal fuera de las celdas ocupadas por la entrada. Para la entrada vacía, reservamos una celda en blanco.

Teorema.

$$A_{\text{LBA}} = \{\langle B, w \rangle : B \text{ es una LBA y acepta } w\}.$$

A_{LBA} es decidable.

Teorema.

$$E_{\text{LBA}} = \{\langle B \rangle : B \text{ es una LBA y } \mathcal{L}(B) = \emptyset\}.$$

E_{LBA} no es decidable.

Teorema.

$$\text{ALL}_{\text{GLC}} = \{\langle G \rangle : G \text{ es una GLC y } \mathcal{L}(G) = \Sigma_G^*\}.$$

ALL_{GLC} no es decidable.

Corolario. No podemos decidir si dos GLC son equivalentes.

Definición. Input: un conjunto finito de dominós $P = \{(t_1, b_1), \dots, (t_k, b_k)\} \subseteq (\Sigma^*)^2$.

$\text{PCP} = \{\langle P \rangle : \text{hay una secuencia no vacía de dominós de } P \text{ con el mismo string arriba y abajo}\}.$

Teorema. PCP no es decidable.

Chequeo rápido

Elija una respuesta en cada pregunta.

1. Sabemos que A es indecidible y queremos probar que B también lo es. ¿Cuál de los siguientes argumentos sirve?
 - a) Usar un supuesto decididor de B para construir un decididor de A .
 - b) Usar un supuesto decididor de A para construir un decididor de B .
 - c) Construir un reconocedor de B .
2. Una reducción transforma $\langle M, w \rangle$ en la descripción de una MT $N_{M,w}$. ¿Qué debe terminar para toda instancia $\langle M, w \rangle$?
 - a) La computación de M sobre w .
 - b) La construcción de la descripción de $N_{M,w}$.
 - c) La computación de $N_{M,w}$ sobre cualquier entrada.
3. Como A_{LBA} es decidable, también podemos decidir si el lenguaje de una LBA es vacío.
 - a) Verdadero.
 - b) Falso.
4. Para comprobar que una secuencia de configuraciones es un historial de computación que acepta, basta verificar que empieza en la configuración inicial y termina en una configuración de aceptación.
 - a) Verdadero.
 - b) Falso.

Ejercicios

1. Demuestre, mediante una reducción en cada caso, que los siguientes problemas son indecidibles.
 - a) Dada una MT M , decidir si acepta el string vacío ε .
 - b) Dados una MT M y un string w , decidir si M rechaza w .
 - c) Dada una MT M sobre $\{0, 1\}$, decidir si $\mathcal{L}(M) = \{0, 1\}^*$.
 - d) Dadas dos GLC sobre el mismo alfabeto, decidir si generan el mismo lenguaje.

2. Demuestre la indecidibilidad de los siguientes problemas.

- a) Dada una MT M , decidir si acepta w^R cada vez que acepta w , para todo string w . Aquí w^R denota el reverso de w .

Indicación: Reduzca desde el problema de la detención.

- b) Un estado de una MT es *inútil* si ninguna computación de la máquina lo visita, cualquiera sea la entrada. Formule como un lenguaje el problema de decidir si una MT tiene al menos un estado inútil y demuestre que es indecible.

Indicación: Reduzca desde chequear si una MT tiene lenguaje vacío.

- c) Una GLC es *ambigua* si algún string tiene dos árboles de derivación distintos. Demuestre que el siguiente lenguaje no es decidable:

$$\text{AMBIG}_{\text{GLC}} = \{\langle G \rangle : G \text{ es una GLC ambigua}\}.$$

Indicación: Reduzca desde el problema de correspondencia de Post.

3. Sea $\mathcal{P}$ una familia de lenguajes Turing-reconocibles sobre $\{0, 1\}$. Diremos que $\mathcal{P}$ es *no trivial* si existen lenguajes Turing-reconocibles $L_0, L_1 \subseteq \{0, 1\}^*$ tales que $L_0 \in \mathcal{P}$ y $L_1 \notin \mathcal{P}$. Definimos

$$\text{PROP}_{\mathcal{P}} = \{\langle M \rangle : M \text{ es una MT sobre } \{0, 1\} \text{ y } \mathcal{L}(M) \in \mathcal{P}\}.$$

Pruebe el *teorema de Rice*: si $\mathcal{P}$ es no trivial, entonces $\text{PROP}_{\mathcal{P}}$ no es decidable.

- a) Demuestre primero el resultado cuando $\emptyset \notin \mathcal{P}$. Elija un lenguaje $L \in \mathcal{P}$ y una MT T que lo reconozca. Dada una instancia $\langle M, w \rangle$ de A_{MT} , construya una MT $N_{M,w}$ sobre $\{0, 1\}$ tal que

$$\mathcal{L}(N_{M,w}) = \begin{cases} L, & \text{si } M \text{ acepta } w, \\ \emptyset, & \text{si } M \text{ no acepta } w. \end{cases}$$

Indicación: determine en qué orden $N_{M,w}$ debe simular a M sobre w y a T sobre su propia entrada. Considere tanto el caso en que M rechaza w como aquel en que no se detiene.

- b) Justifique que la descripción de $N_{M,w}$ puede construirse efectivamente sin resolver previamente si M acepta w . Pruebe que

$$\langle M, w \rangle \in A_{\text{MT}} \iff \langle N_{M,w} \rangle \in \text{PROP}_{\mathcal{P}},$$

y concluya la indecidibilidad en este caso.

- c) Deduzca el caso general.
- d) Aplique el resultado para demostrar que no es decidable si el lenguaje reconocido por una MT sobre $\{0, 1\}$ es finito.