

# La noción de reducción

Dado: Lenguajes  $L_1$  y  $L_2$  con alfabeto  $A$ .

Intuición: Decimos que un lenguaje  $L_1$  es **reducible** a un lenguaje  $L_2$  si existe un algoritmo tal que dado  $w_1 \in A^*$ , calcula  $w_2 \in A^*$  tal que  **$w_1 \in L_1$  si y sólo si  $w_2 \in L_2$** .

# La noción de reducción

Dado: Lenguajes  $L_1$  y  $L_2$  con alfabeto  $A$ .

Intuición: Decimos que un lenguaje  $L_1$  es **reducible** a un lenguaje  $L_2$  si existe un algoritmo tal que dado  $w_1 \in A^*$ , calcula  $w_2 \in A^*$  tal que  **$w_1 \in L_1$  si y sólo si  $w_2 \in L_2$** .

Para formalizar esta idea necesitamos introducir la noción de **algoritmo que calcula**.

- Tenemos que introducir máquinas de Turing que calculan

# Máquinas de Turing que calculan

## Definición

Una MT  $M$  calcula una función  $f : A^* \rightarrow A^*$  si:

*con entrada  $w \in A^*$ , la máquina se detiene en un estado final teniendo en la memoria una cadena infinita de símbolos  $B$  seguida de  $f(w)$  seguida de otra cadena infinita de símbolos  $B$ .*

# Máquinas de Turing que calculan

## Definición

Una MT  $M$  calcula una función  $f : A^* \rightarrow A^*$  si:

*con entrada  $w \in A^*$ , la máquina se detiene en un estado final teniendo en la memoria una cadena infinita de símbolos  $B$  seguida de  $f(w)$  seguida de otra cadena infinita de símbolos  $B$ .*

## Definición

Una función  $f$  es **computable** si es que existe una máquina de Turing que calcula  $f$ .

# Máquinas de Turing que calculan

## Ejemplo

Sea  $g : \{0, 1\}^* \rightarrow \{0, 1\}^*$  una función definida como  $g(w) = w0000$ . Queremos demostrar que  $g$  es computable.

# Máquinas de Turing que calculan

## Ejemplo

Sea  $g : \{0, 1\}^* \rightarrow \{0, 1\}^*$  una función definida como  $g(w) = w0000$ . Queremos demostrar que  $g$  es computable.

Sea  $M = (Q, A, q_1, \delta, F)$ , donde

- ▶  $Q = \{q_1, q_2, q_3, q_4, q_F\}$ .
- ▶  $A = \{0, 1\}$ .
- ▶  $F = \{q_F\}$ .
- ▶  $\delta$  es definida como:

$$\delta(q_1, 0) = (q_1, 0, \rightarrow)$$

$$\delta(q_1, 1) = (q_1, 1, \rightarrow)$$

$$\delta(q_1, B) = (q_2, 0, \rightarrow)$$

$$\delta(q_2, B) = (q_3, 0, \rightarrow)$$

$$\delta(q_3, B) = (q_4, 0, \rightarrow)$$

$$\delta(q_4, B) = (q_F, 0, \rightarrow)$$

# La noción de reducción: Formalización

## Definición

*Dados lenguajes  $L_1$  y  $L_2$  con alfabeto  $A$ ,  $L_1$  es **reducible** a  $L_2$  si existe una función computable  $f$  tal que para todo  $w \in A^*$ :*  
 *$w \in L_1$  si y sólo si  $f(w) \in L_2$ .*

# La noción de reducción: Formalización

## Definición

Dados lenguajes  $L_1$  y  $L_2$  con alfabeto  $A$ ,  $L_1$  es *reducible* a  $L_2$  si existe una función computable  $f$  tal que para todo  $w \in A^*$ :  
 *$w \in L_1$  si y sólo si  $f(w) \in L_2$ .*

## Teorema

Suponga que  $L_1$  es reducible a  $L_2$ .

- ▶ Si  $L_2$  es decidable entonces  $L_1$  es decidable
- ▶ Si  $L_1$  es indecidible entonces  $L_2$  es indecidible



# La noción de reducción: Formalización

## Definición

Dados lenguajes  $L_1$  y  $L_2$  con alfabeto  $A$ ,  $L_1$  es *reducible* a  $L_2$  si existe una función computable  $f$  tal que para todo  $w \in A^*$ :  
 *$w \in L_1$  si y sólo si  $f(w) \in L_2$ .*

## Teorema

Suponga que  $L_1$  es reducible a  $L_2$ .

- ▶ Si  $L_2$  es decidable entonces  $L_1$  es decidable
- ▶ Si  $L_1$  es indecidible entonces  $L_2$  es indecidible

## Ejercicio

Demuestre el teorema.

# El problema de la parada: Demostración

Por fin estamos listos para demostrar que el problema de la parada  $U$  es indecidible.

# El problema de la parada: Demostración

Por fin estamos listos para demostrar que el problema de la parada  $U$  es indecidible.

**Demostración:** Sea  $g : \{0, 1\}^* \rightarrow \{0, 1\}^*$  definida como  $g(w) = w0000w$ .

Es fácil ver que  $g$  es computable y es una reducción de  $H$  a  $U$ .

Dado que  $H$  es indecidible, concluimos que  $U$  es indecidible.