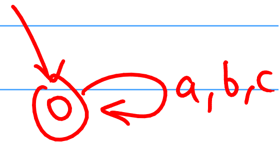
Un problema A se reduce a un problema B cuando un método para resolver el problema B puede servir para resolver el problema A.

Ej. $ALL_{AFD} = \{ \langle D \rangle \mid D$ es un A.F.D. tq. $\mathcal{L}(D) = \Sigma^* \}$



$EQ_{AFD} = \{ \langle D_1, D_2 \rangle \mid D_1, D_2$ son A.F.D.s tq. $\mathcal{L}(D_1) = \mathcal{L}(D_2) \}$.

El problema de decidir si una cadena está en ALL_{AFD} se puede reducir al problema de decidir si una cadena está en EQ_{AFD}.

<u>Observaciones</u>

1) Si A se reduce a B, y $\underbrace{\text{B es decidible}}_{P}$, entonces $\underbrace{\text{A es decidible}}_{Q}$.

$P \Rightarrow Q$

$\equiv$

$\neg Q \Rightarrow \neg P$

2) Si A se reduce a B, y $\underbrace{\text{A es indecidible}}_{\neg Q}$, entonces $\underbrace{\text{B es indecidible}}_{\neg P}$.

Recordemos

$$A_{MT} = \{ \langle M, w \rangle \mid M(w) = \text{acepta} \}$$

A_{MT} es indecidible.

"HALTING PROBLEM"

Def. $HALT_{MT} = \{ \langle M, w \rangle \mid M(w) \text{ termina} \}$

es decir $M(w) = \text{acepta}$ ó $M(w) = \text{rechaza}$.

Teorema. $HALT_{MT}$ es indecidible.

Idea. Reducir el problema A_{MT} al problema $HALT_{MT}$.

Dem. Supongamos, por el absurdo, que $HALT_{MT}$ es decidible. Es decir, existe una M.T. R tal que:

$$R(\langle M, w \rangle) = \begin{cases} \text{acepta} & \text{si } M(w) \text{ termina} \\ \text{rechaza} & \text{si } M(w) \text{ no termina} \end{cases}$$

Construimos una M.T. S así:

$$S(\langle M, w \rangle) = \begin{cases} \text{acepta} & \bullet \text{ si } R(\langle M, w \rangle) = \text{acepta} \text{ y } M(w) = \text{acepta} \\ \text{rechaza} & \bullet \text{ si } R(\langle M, w \rangle) = \text{acepta} \text{ y } M(w) = \text{rechaza} \\ \text{rechaza} & \bullet \text{ si } R(\langle M, w \rangle) = \text{rechaza} \end{cases}$$

Observemos que

- $S(\langle M, w \rangle) = \text{acepta} \iff M(w) = \text{acepta}$
  $\iff \langle M, w \rangle \in A_{MT}$
- $S(\langle M, w \rangle) = \text{rechaza} \iff M(w) \neq \text{acepta}$
  $\iff \langle M, w \rangle \notin A_{MT}$

es decir, la M.T. S decide el lenguaje A_{MT}.
Absurdo, porque ya sabíamos que A_{MT} es indecidible.

Recordemos que $\mathcal{L}(M) = \{ w \mid M(w) = \text{acepta} \}$.

Def. $E_{MT} = \{ \langle M \rangle \mid M \text{ es una M.T. tal que } \mathcal{L}(M) = \emptyset \}$

Teorema. El lenguaje E_{MT} es indecidible.

Idea. Reducir A_{MT} a E_{MT}.

Dem. Por el absurdo, supongamos que E_{MT} es decidible.

- Es decir, existe una M.T. R tal que

$$R(\langle M \rangle) = \begin{cases} \text{acepta} & \text{si } \mathcal{L}(M) = \emptyset \\ \text{rechaza} & \text{si } \mathcal{L}(M) \neq \emptyset \end{cases}$$

- Dada una máquina de Turing M y una cadena w, podemos construir otra máquina M_w así:

$$M_w(v) = \begin{cases} M(w) & \text{si } v = w \\ \text{rechaza} & \text{si } v \neq w \end{cases}$$

Observemos que

$$\mathcal{L}(M_w) = \emptyset \iff M(w) \neq \text{acepta}.$$

Es decir:

$$M(w) = \text{acepta} \iff \mathcal{L}(M_w) \neq \emptyset.$$

Construimos ahora una máquina de Turing S así:

$$S(\langle M, w \rangle) = \begin{cases} \text{rechaza} & \text{si } R(\langle M_w \rangle) = \text{acepta} \\ \text{acepta} & \text{si } R(\langle M_w \rangle) = \text{rechaza} \end{cases}$$

Ahora:

- $S(\langle M, w \rangle) = \text{acepta} \iff R(\langle M_w \rangle) = \text{rechaza} \iff \mathcal{L}(M_w) \neq \emptyset$
  $\iff M(w) = \text{acepta} \iff \langle M, w \rangle \in A_{MT}$
- $S(\langle M, w \rangle) = \text{rechaza} \iff R(\langle M_w \rangle) = \text{acepta} \iff \mathcal{L}(M_w) = \emptyset$
  $\iff M(w) \neq \text{acepta} \iff \langle M, w \rangle \notin A_{MT}$.

Por lo tanto S decide el lenguaje A_{MT}.

Absurdo, pues ya sabíamos que el lenguaje A_{MT} es indecidible.

Def. $\text{REGULAR}_{MT} = \{ \langle M \rangle \mid \mathcal{L}(M) \text{ es un lenguaje regular} \}$.

Teorema. REGULAR_{MT} es indecidible.

Idea. Reducir A_{MT} a REGULAR_{MT}.

Dem. Por el absurdo, supongamos que REGULAR_{MT} es decidible. Es decir, existe una M.T. R tal que

$$R(\langle M \rangle) = \begin{cases} \text{acepta} & \text{si } \mathcal{L}(M) \text{ es regular} \\ \text{rechaza} & \text{si } \mathcal{L}(M) \text{ no es regular.} \end{cases}$$

- $\emptyset$ es un lenguaje regular
- Σ^* es un lenguaje regular
- $X = \{ 0^n 1^n \mid n \in \mathbb{N} \}$.

- Dada una máquina de Turing M y una palabra w, podemos construir una máquina de Turing M_w así:

$$M_w(v) = \begin{cases} M(w) & \text{si } v \in X \\ \text{rechaza} & \text{si } v \notin X \end{cases}$$

observemos que

- $\mathcal{L}(M_w) = X$ si $M(w) = \text{acepta}$
- $\mathcal{L}(M_w) = \emptyset$ si $M(w) \neq \text{acepta}$

es decir:

$$M(w) = \text{acepta} \iff \mathcal{L}(M_w) \text{ no es regular}$$

Construimos ahora una M.T. S de la siguiente manera:

$$S(\langle M, w \rangle) = \begin{cases} \text{acepta} & R(\langle M_w \rangle) = \text{rechaza} \\ \text{rechaza} & R(\langle M_w \rangle) = \text{acepta} \end{cases}$$

Observemos que:

- $S(\langle M, w \rangle) = \text{acepta} \iff R(\langle M_w \rangle) = \text{rechaza}$
$\iff \mathcal{L}(M_w)$ no es regular
$\iff M(w) = \text{acepta}$
$\iff \langle M, w \rangle \in A_{MT}$.

- $S(\langle M, w \rangle) = \text{rechaza} \iff R(\langle M_w \rangle) = \text{acepta}$
$\iff \mathcal{L}(M_w)$ es regular
$\iff M(w) \neq \text{acepta}$
$\iff \langle M, w \rangle \notin A_{MT}$.

Por lo tanto S decide el lenguaje A_{MT}.

Absurdo, pues ya sabíamos que A_{MT} es indecidible.

Recordemos: $E_{MT} = \{ \langle M \rangle \mid \mathcal{L}(M) = \emptyset \}$ es indecidible.

Def. $EQ_{MT} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2$ son M.T.s tales que $\mathcal{L}(M_1) = \mathcal{L}(M_2) \}$

Teorema. El lenguaje EQ_{MT} es indecidible.

Idea. Reducir el lenguaje E_{MT} al lenguaje EQ_{MT}.

Dem. Por el absurdo, supongamos que EQ_{MT} es decidible.
Es decir existe una M.T. R tal que:

$$R(\langle M_1, M_2 \rangle) = \begin{cases} \text{acepta} & \text{si } \mathcal{L}(M_1) = \mathcal{L}(M_2) \\ \text{rechaza} & \text{si } \mathcal{L}(M_1) \neq \mathcal{L}(M_2) \end{cases}$$

Construimos una M.T. S así:

$$S(\langle M \rangle) = \begin{cases} \text{acepta} & \text{si } R(\langle M, M_0 \rangle) = \text{acepta} \\ \text{rechaza} & \text{si } R(\langle M, M_0 \rangle) = \text{rechaza} \end{cases}$$

donde M_0 es una M.T. que rechaza siempre todas las entradas.

Observemos que

- $S(\langle M \rangle) = \text{acepta} \Leftrightarrow \mathcal{L}(M) = \mathcal{L}(M_0) = \emptyset \Leftrightarrow \langle M \rangle \in E_{MT}$.
- $S(\langle M \rangle) = \text{rechaza} \Leftrightarrow \mathcal{L}(M) \neq \mathcal{L}(M_0) = \emptyset \Leftrightarrow \langle M \rangle \notin E_{MT}$.

Por lo tanto, S decide el lenguaje E_{MT}.
Absurdo, pues ya sabíamos que E_{MT} es indecidible.

Def. Si M es una máquina de Turing y w es una cadena, un historial de cómputos es una secuencia $C_1, C_2, \ldots, C_n$ donde cada C_i es una configuración y específicamente:

- C_1 es la configuración inicial de la máquina M cuando se la alimenta con la cadena w.
- C_{i+1} se obtiene a partir de C_i $(\forall i \in 1..n-1)$ aplicando las reglas de transición correspondientes de la máquina M.

Además, si C_n corresponde a un estado de aceptación, decimos que $C_1, \ldots, C_n$ es un historial de cómputos de aceptación.

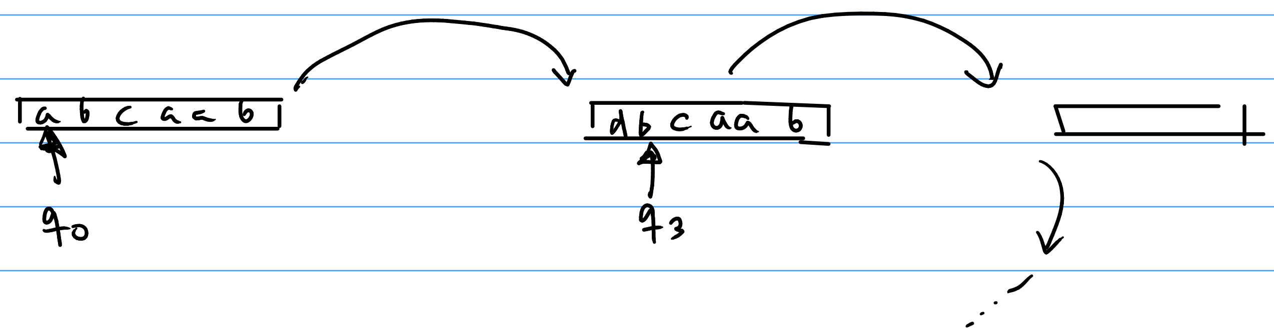
Def. Un autómata linealmente acotado (LBA) es una variante de las máquinas de Turing en la cual el cabezal No se puede mover más a la izquierda ni más a la derecha en la cinta que el fragmento de la cinta ocupado por la entrada.

Def. $A_{LBA} = \{ \langle B, w \rangle \mid B$ es un LBA, $w \in \Sigma^*$, $B(w) = \text{acepta} \}$.

Lema. Si B es un LBA que tiene q estados y g símbolos de cinta.
Si la entrada es de longitud n, la cantidad de configuraciones que se pueden alcanzar al alimentar B con la cadena w es a lo sumo:

$$g^n \cdot q \cdot n$$

Dem.



- El contenido de la cinta podría tener a lo sumo g^n cadenas posibles.
- El estado actual de la máquina podría ser cualquiera de los q estados posibles.
- La posición del cabezal podría ser cualquiera de las n posiciones posibles.

En total hay $g^n \cdot q \cdot n$ configuraciones posibles.

$A_{LBA} = \{ \langle B, w \rangle \mid B$ es un LBA, $w \in \Sigma^*$, $B(w) = \text{acepta} \}$.

Teorema. El lenguaje A_{LBA} es decidible.

Dem. Construimos una máquina de Turing M, así:

$n = |w|$

$$M(\langle B, w \rangle) = \begin{cases} \text{acepta} & \bullet\ B(w) \text{ termina en menos de } g^n \cdot q \cdot n \text{ pasos y } B(w) = \text{acepta} \\ \text{rechaza} & \bullet\ B(w) \text{ termine en menos de } g^n \cdot q \cdot n \text{ pasos y } B(w) = \text{rechaza} \\ \text{rechaza} & \bullet\ B(w) \text{ No termina en menos de } g^n \cdot q \cdot n \text{ pasos} \end{cases}$$

Veamos que efectivamente M decide el lenguaje A_{LBA}.

1) Si $M(\langle B, w \rangle) = \text{acepta}$, entonces $B(w) = \text{acepta}$. ✓

2) Si $M(\langle B, w \rangle) = \text{rechaza}$, entonces $B(w) \neq \text{acepta}$?

- si $B(w)$ termina en menos de $g^n \cdot q \cdot n$ pasos, en ese caso $B(w) = \text{rechaza}$.
- si $B(w)$ No termina en menos de $g^n \cdot q \cdot n$ pasos,

$B(w)$ entra en un ciclo !!

no termina ✓

Def. $E_{LBA} = \{ \langle B \rangle \mid \mathcal{L}(B) = \emptyset \}$.

Teorema. El lenguaje E_{LBA} es indecidible.

Idea. Reducir el problema A_{MT} a E_{LBA}.

Dem. Supongamos, por el absurdo, que E_{LBA} es decidible. Es decir, existe una M.T. R tal que

$$R(\langle B \rangle) = \begin{cases} \text{acepta} & \text{si } \mathcal{L}(B) = \emptyset \\ \text{rechazar} & \text{si } \mathcal{L}(B) \neq \emptyset. \end{cases}$$

- Dada una M.T. M y una cadena w, podemos construir un LBA $B_{M,w}$ así:

$$B_{M,w}(v) = \begin{cases} \text{acepta} & \text{• si } v \text{ es un historial de cómputos de aceptación de } M, w \\ \text{rechaza} & \text{• si no} \end{cases}$$

Observemos que

- $\mathcal{L}(B_{M,w}) \neq \emptyset$ si $M(w) = \text{acepta}$
- $\mathcal{L}(B_{M,w}) = \emptyset$ si $M(w) \neq \text{acepta}$

Construimos una M.T. S así:

$$S(\langle M, w \rangle) = \begin{cases} \text{acepta} & R(\langle B_{M,w} \rangle) = \text{rechaza} \\ \text{rechaza} & R(\langle B_{M,w} \rangle) = \text{acepta} \end{cases}$$

Observemos que:

$$\begin{aligned} S(\langle M, w \rangle) = \text{acepta} &\Leftrightarrow R(\langle B_{M,w} \rangle) = \text{rechaza} \\ &\Leftrightarrow \mathcal{L}(B_{M,w}) \neq \emptyset \\ &\Leftrightarrow M(w) = \text{acepta} \Leftrightarrow \langle M, w \rangle \in A_{MT} \end{aligned}$$

$$
\begin{aligned}
S(\langle M, w\rangle) = \text{rechaza} &\iff R(\langle B_M, w\rangle) = \text{acepta} \\
&\iff \mathcal{L}(B_{M,w}) = \emptyset \\
&\iff M(w) \neq \text{acepta} \\
&\iff \langle M, w\rangle \notin A_{MT}.
\end{aligned}
$$

Por lo tanto,

$$S \text{ decide el lenguaje } A_{MT}.$$

Absurdo, pues ya sabíamos que A_{MT} es indecidible.

Problema de Correspondencia de Post

Fijamos un alfabeto Σ.

Un **bloque** es un par (w_1, w_2) donde $w_1, w_2 \in \Sigma^*$.

$$\left[\frac{w_1}{w_2}\right]$$

Un **rompecabezas** es un conjunto finito de bloques.

Por ejemplo:

$$P_0 = \left\{ \left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right] \right\}$$ es un rompecabezas.

Una **solución** a un rompecabezas P_0 es una secuencia finita de bloques

$$\left[\frac{w_1}{v_1}\right] \left[\frac{w_2}{v_2}\right] \cdots \left[\frac{w_n}{v_n}\right]$$

tales que: (1) $n > 0$

(2) $w_1 w_2 \ldots w_n = v_1 v_2 \ldots v_n$

(3) $\left[\frac{w_i}{v_i}\right] \in P \quad \forall i \in 1..n$

Por ejemplo, una solución para este rompecabezas

$$\left\{ \left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right] \right\}$$

podría ser:

$$\left[\frac{a}{ab}\right] \left[\frac{b}{ca}\right] \left[\frac{ca}{a}\right] \left[\frac{a}{ab}\right] \left[\frac{abc}{c}\right]$$

Def.

$$PCP = \{ \langle P \rangle \mid P \text{ es un rompecabezas que tiene solución} \}$$

$$P_0 \in PCP$$

Pero hay rompecabezas que no tienen solución.
Por ejemplo:

$$P_1 = \left\{ \left[\frac{a}{b}\right], \left[\frac{a}{c}\right] \right\} \qquad \langle P_1 \rangle \notin PCP.$$

$$P_2 = \left\{ \left[\frac{ab}{a}\right], \left[\frac{ba}{b}\right] \right\} \qquad \langle P_2 \rangle \notin PCP$$

$PCP = \{ \langle P \rangle \mid P$ es un rompecabezas que tiene solución$\}$

Teorema. El lenguaje PCP no es decidible.

Idea: Reducir el problema A_{MT} a PCP.

Dem. Supongamos, por el absurdo, que PCP es decidible.

- Entonces existe una MT R tq.

$$R(\langle P \rangle) = \begin{cases} \text{acepta} & \text{si } P \text{ tiene solución} \\ \text{rechaza} & \text{si } P \text{ no tiene solución} \end{cases}$$

- **Objetivo:** Construir una M.T. S que decida A_{MT}.

Es decir queremos que S cumpla lo siguiente:

$$\boxed{\begin{array}{lcl} S(\langle M, w \rangle) = \text{acepta} & \iff & M(w) = \text{acepta} \\ S(\langle M, w \rangle) = \text{rechaza} & \iff & M(w) \neq \text{acepta}. \end{array}}$$

- **Idea:** Construir un rompecabezas que tenga solución si y solamente si hay un historial de cómputos de aceptación para $\langle M, w \rangle$.

  Una vez hecho esto, podemos aplicar la máquina R para determinar si el rompecabezas tiene o no tiene solución.

Dadas M, w,
construir un rompecabezas que
tenga solución si y sólo si
existe un historial de cómputos de aceptación para M, w.

abbc

$\bullet \#$	$\bullet q_0 \bullet a$	$\bullet b$	$\bullet b$	$\bullet c$	$\bullet \#$
$\bullet \# \bullet q_0 \bullet a \bullet b \bullet b \bullet c \bullet \# \bullet$	$d \bullet q_2 \bullet$	$b \bullet$	$b \bullet$	$c \bullet$	$\# \bullet$

$$\delta(q_0, a) = (q_2, d, R)$$

Una reducción funcional de un problema A a un problema B es una <u>función computable</u> que convierte instancias del problema A en instancias del problema B de tal modo que la solución a la instancia del problema B es también la solución a la instancia del problema A original.

Def. Una función $f : \Sigma^* \to \Sigma^*$ es computable si existe una máquina de Turing M tal que para cada palabra $w \in \Sigma^*$ si se inicializa la máquina M con la palabra w en la cinta, la máquina se detiene y el contenido de la cinta es exactamente la palabra $f(w)$.

Ej. Las funciones aritméticas entre enteros (escritos en base 10) son computables.

+, −, *, /, %, potencia

Def (Reducción funcional, reducción "many-one").

"de muchos a uno"

Sean Σ un alfabeto,
y $A, B \subseteq \Sigma^*$ lenguajes.
Decimos que A se reduce "muchos a uno" a B si existe una función computable $f : \Sigma^* \to \Sigma^*$ tal que:

$$\forall w \in \Sigma^*, \quad w \in A \iff f(w) \in B.$$

Notamos $A \leq_m B$.

Teorema. Si $A \leq_m B$ y B es decidible, entonces A es decidible.

Dem.

1) Como $A \leq_m B$, existe una función computable

$$f : \Sigma^* \longrightarrow \Sigma^*$$

tal que $\forall w \in \Sigma^*$. $w \in A \iff f(w) \in B$.

2) Como B es decidible, existe una máquina de Turing R tal que

$$R(w) = \begin{cases} \text{acepta} & \text{si } w \in B \\ \text{rechaza} & \text{si } w \notin B. \end{cases}$$

3) Construimos una M.T. S así:

$$S(w) = R(f(w))$$

Observemos que:

- $S(w) = \text{acepta} \iff R(f(w)) = \text{acepta}$
$\iff f(w) \in B$
$\iff w \in A.$

- $S(w) = \text{rechaza} \iff R(f(w)) = \text{rechaza}$
$\iff f(w) \notin B$
$\iff w \notin A$

Por lo tanto S decide el lenguaje A.

Por lo tanto A es decidible.

$P \Rightarrow Q$

Teorema. Si $A \leq_m B$ y $\underbrace{B \text{ es decidible}}_{P}$, entonces $\underbrace{A \text{ es decidible}}_{Q}$.

Corolario. Si $A \leq_m B$ y $\underbrace{A \text{ es indecidible}}_{\neg Q}$, entonces $\underbrace{B \text{ es indecidible}}_{\neg P}$.

$\neg Q \Rightarrow \neg P$

Teorema. Si $A \leq_m B$ y B es semi-decidible, entonces A es semi-decidible.

Corolario. Si $A \leq_m B$ y A no es semi-decidible, entonces B no es semi-decidible.

Observación. Si $A \leq_m B$ entonces $\overline{A} \leq_m \overline{B}$.

$$\left[\begin{array}{l} f : \Sigma^* \longrightarrow \Sigma^* \text{ computable} \\ w \in A \Leftrightarrow f(w) \in B \\ w \notin A \Leftrightarrow f(w) \notin B \\ w \in \overline{A} \Leftrightarrow f(w) \in \overline{B} \end{array}\right]$$

$$\mathcal{L}(M) = \{ w \mid M(w) = \text{acepta} \}$$

Def. $EQ_{MT} = \{ \langle M_1, M_2 \rangle \mid \mathcal{L}(M_1) = \mathcal{L}(M_2) \}$

Teorema.

1) EQ_{MT} no es semi-decidible.

2) $\overline{EQ_{MT}}$ no es semi-decidible.

Dem.

1) Para ver que EQ_{MT} no es semi-decidible, recordemos que

$$\overline{A_{MT}} \text{ no es semi-decidible.}$$

Bastaría encontrar una reducción:

$$\overline{A_{MT}} \leq_m EQ_{MT}.$$

Esto es lo mismo que:

$$A_{MT} \leq_m \overline{EQ_{MT}}$$

Es decir, deberíamos encontrar una función f computable tal que

$$\langle M, w \rangle \in A_{MT} \iff f(\langle M, w \rangle) \in \overline{EQ_{MT}}.$$

Para computar f construimos un par de máquinas M_1, M_2:

- M_1 es la máquina que rechaza todo input.
- M_2 es la máquina que ignora su input, sobreescribe la cinta con la entrada w y prosigue a comportarse igual que la máquina M.

$\langle M, w \rangle \in A_{MT} \iff M(w) = \text{acepta} \iff M_2$ acepta todo input

$\iff \mathcal{L}(M_1) \neq \mathcal{L}(M_2) \iff \langle M_1, M_2 \rangle \notin EQ_{MT}$

2) Para ver que $\overline{EQ_{MT}}$ no es semi-decidible, como ya sabemos que $\overline{A_{MT}}$ no es semi-decidible,

bastaría encontrar una reducción $\overline{A_{MT}} \leq_m \overline{EQ_{MT}}$
es decir, queremos encontrar una reducción:

$$A_{MT} \leq_m EQ_{MT}$$

Es decir, queremos encontrar una función computable f tal que

$$\langle M, w \rangle \in A_{MT} \Leftrightarrow f(\langle M, w \rangle) \in EQ_{MT}.$$

Para computar f construimos un par de máquinas de Turing:

- M_1 acepta cualquier
- M_2 ignora su input, sobreescribe la cinta con la palabra w, y prosigue comportándose igual que M.

$$
\begin{aligned}
\langle M, w \rangle \in A_{MT} &\Leftrightarrow M(w) = \text{acepta} \\
&\Leftrightarrow M_2 \text{ acepta todo input} \\
&\Leftrightarrow \mathcal{L}(M_1) = \mathcal{L}(M_2) \\
&\Leftrightarrow \langle M_1, M_2 \rangle \in EQ_{MT}
\end{aligned}
$$