

Versión: 21/03/2020

Máquinas de Turing y Decidibilidad

1. Máquinas de Turing

Recuerde que una máquina de Turing (MT) es una tupla $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$ donde

- Q es el conjunto finito de estados de M ,
- Σ es el alfabeto de entrada,
- Γ es el alfabeto de la cinta que satisface $\Sigma \subseteq \Gamma$ y Γ tiene un símbolo especial ' $\sqcup$ ' $\notin \Sigma$ para el blanco,
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$,
- q_0 en Q es el estado inicial,
- q_{accept} en Q es el estado de aceptación,
- q_{reject} en Q es el estado de rechazo.

Ejecutar la máquina de M sobre la entrada (palabra) ω consiste en correr la máquina sobre la configuración inicial que consiste de colocar la palabra ω en la parte más a la izquierda de la cinta, colocar la cabeza de lectura/escritura sobre el primer símbolo de ω , o la posición mas a la izquierda de la cinta si ω es la palabra vacía, y colocar el estado de la máquina en q_0 ; el resto de la cinta desde el último símbolo en ω contiene el símbolo $\sqcup$. La máquina acepta la palabra si durante su ejecución entra al estado q_{accept} . Si en cambio, la máquina entra al estado q_{reject} , la máquina rechaza la palabra. La otra posibilidad es que la máquina no termine la ejecución. El lenguaje reconocido por M son todas aquellas palabras ω que son aceptadas por M . Un lenguaje L es **Turing reconocible** si existe una máquina de Turing que lo reconoce. Un lenguaje L es **Turing decidible** si existe una máquina de Turing M que lo reconoce y M termina ejecución en todas las entradas. Si L es Turing decidible decimos simplemente que L es decidible. Si no lo es, decimos que L es indecidible.

Existen diferentes versiones del modelo computacional: máquinas con cinta infinita en ambas direcciones (izquierda y derecha), con múltiples cintas, y también máquinas no deterministas. Todas estas versiones son equivalentes a la MT simple. Es decir, si un lenguaje es reconocido/decidido por una version de la MT, entonces es reconocido/decidido por una MT simple. La tesis de **Church-Turing** estipula que la MT simple es computacionalmente equivalente a cualquier model computacional razonable. En otras palabras, cualquier modelo computacional que sea más “poderoso” que una MT simple se considera que no es razonable.

Ejercicios

Resuelva los siguientes ejercicios, consultado el libro u otras fuentes. Debe hacer referencia a todas las fuentes consultadas. En los ejercicios que se piden demostraciones, debe dar argumentos convincentes y correctos que establezcan lo que se quiere demostrar aún cuando no se cubran todos los detalles de una demostración formal.

- 1.a) Construya una máquina de Turing, con todos los detalles, que *decida* el lenguaje $L = \{\omega \in \{0, 1\}^* : \omega \text{ tiene el doble de 0s que 1's}\}$.
- 1.b) Construya una máquina de Turing, con todos los detalles, que *decida* el lenguaje $L = \{\omega \in \{0, 1\}^* : \omega = \omega^R\}$.
- 1.c) Muestre que un lenguaje es decidible si y sólo si existe un enumerador que enumera el lenguaje en orden lexicográfico.
- 1.d) Utilice el resultado anterior para mostrar que todo lenguaje infinito que es Turing reconocible contiene un lenguaje infinito que es Turing decidible; i.e., para todo lenguaje Turing reconocible A tal que $|A| = \text{infy}$, existe un subconjunto $B \subseteq A$ tal que $|B| = \infty$ y B es Turing decidible.

2. Decidibilidad

Un lenguaje L es decidible si existe una MT que termina en todas las entradas y reconoce L . Los siguientes lenguajes son decidibles:

- $A_{\text{DFA}} = \{\langle B, \omega \rangle : B \text{ es un AFD y } \omega \text{ es aceptado por } B\}$,
- $A_{\text{NFA}} = \{\langle B, \omega \rangle : B \text{ es un AFND y } \omega \text{ es aceptado por } B\}$,
- $A_{\text{REX}} = \{\langle R, \omega \rangle : R \text{ es una expresión regular y } \omega \in R\}$,
- $E_{\text{DFA}} = \{\langle A \rangle : A \text{ es un AFD y } L(A) = \emptyset\}$,
- $EQ_{\text{DFA}} = \{\langle A, B \rangle : A \text{ y } B \text{ son AFDs y } L(A) = L(B)\}$,
- $A_{\text{CFG}} = \{\langle G, \omega \rangle : G \text{ es una gramática libre de contexto y } \omega \in L(G)\}$,
- $E_{\text{CFG}} = \{\langle G, \omega \rangle : G \text{ es una gramática libre de contexto y } L(G) = \emptyset\}$.

Problema de Aceptación para MTs

En esta sección demostraremos la existencia de problemas (lenguajes) para los cuales *no existe* máquinas de Turing que los resuelvan. Este resultado establece limitaciones fuertes sobre lo que se puede computar. Por la tesis de Church-Turing, si no existe una MT para un problema dado, dicho problema no puede ser resuelto bajo ningún modelo computacional razonable.

Las principales herramientas para demostrar que un lenguaje no es decidible son el método de diagonalización y el método de reducción. Ambos métodos establecen la *indecidibilidad* de un lenguaje por reducción al absurdo: si suponemos que L es decidible, entonces se puede derivar una contradicción. Lo que cambia en ambos métodos es la forma de derivar la contradicción. En el segundo método, el método de reducción, la contradicción viene de poder construir una máquina de Turing que decide un lenguaje L' que ya se conoce que es indecidible. Luego, para usar el segundo método se necesita conocer al menos un lenguaje L' que sea indecidible. El primer método, el

método de diagonalización, logra derivar una contradicción directamente sin necesidad de recurrir a otro lenguaje. Para hacer esto, se construye una máquina de Turing M^* especial que se diferencia a todas las máquinas de Turing existentes lo que en si mismo es una contradicción.

Considere el problema $A_{TM} = \{\langle M, \omega \rangle : M \text{ es una MT y } \omega \in L(M)\}$, análogo a los problemas A_{DFA} y A_{NFA} pero donde el modelo computacional cambiar de autómata finito se cambia por máquina de Turing. Este es el problema de *aceptación para máquinas de Turing*.

Observe que la siguiente MT U reconoce el lenguaje A_{TM} .

MT U : En entrada $\langle M, \omega \rangle$, donde $\langle M \rangle$ es la descripción de una MT y ω es una palabra sobre el alfabeto de entrada de M :

1. Simula la máquina M en la entrada ω .
2. Si durante la simulación, M entra en el estado de aceptación, ACEPTAR. Si M entra en el estado de rechazo, RECHAZAR.

Lo primero que hace U es verificar que la palabra de entrada tiene el formato correcto $\langle M, \omega \rangle$. Es decir, que según una codificación pre-establecida, la palabra codifica un par donde la primera componente se corresponde a una MT M y la segunda componente a una palabra ω sobre el alfabeto Σ de M .

La máquina U reconoce el lenguaje A_{TM} ya que si $\langle M, \omega \rangle \in A_{TM}$, entonces U acepta $\langle M, \omega \rangle$. Sin embargo, U *no decide* A_{TM} ya que si la simulación no termina, entonces U no termina la ejecución sobre la entrada $\langle M, \omega \rangle$. Como demostraremos a continuación, el lenguaje A_{TM} no es decidible, y en consecuencia habremos demostrado que existen lenguajes que son Turing reconocibles pero no Turing decidibles.

Teorema 1. $A_{TM} = \{\langle M, \omega \rangle : M \text{ es una MT y } \omega \in L(M)\}$ es indecidible.

Demostración. Por el método de diagonalización. Suponga que A_{TM} es decidible y sea H una MT que decide A_{TM} . Construiremos una MT D que utiliza a H como *subrutina*.

MT D : En entrada $\langle M \rangle$ donde $\langle M \rangle$ es la descripción de una MT:

1. Ejecuta H en entrada $\langle M, \langle M \rangle \rangle$.
2. Si H acepta, RECHAZAR. Si H rechaza, ACEPTAR.

Es decir, D hace lo opuesto a H en entrada $\langle M, \langle M \rangle \rangle$. Observe que D es una MT que termina en todas las entradas porque H también termina en todas las entradas. Formalmente, la MT D se puede expresar de la siguiente manera:

$$D(\langle M \rangle) = \begin{cases} \text{ACEPTA} & \text{Si } M \text{ no acepta } \langle M \rangle, \\ \text{RECHAZA} & \text{Si } M \text{ acepta } \langle M \rangle. \end{cases}$$

No confundirse con correr la máquina M en entrada $\langle M \rangle$ ya que $\langle M \rangle$ es una palabra como cualquier otra palabra, excepto que ella codifica una MT.

Ahora corremos D en entrada $\langle D \rangle$. Veamos que, por construcción, esto origina una contradicción. De hecho, por definición, D acepta $\langle D \rangle$ si y sólo si D no acepta $\langle D \rangle$.

Lo que ha pasado es lo siguiente. Al haber supuesto la existencia de H , hemos podido construir la MT D que es *diferente a todas las máquinas de Turing*. En particular, D se diferencia de la MT M en su comportamiento para la palabra $\langle M \rangle$: D acepta $\langle M \rangle$ si y sólo si M rechaza $\langle M \rangle$. Por lo

tanto, al ser D diferente de cualquier máquina de Turing, D no puede ser una máquina de Turing. Contradicción.

La contradicción viene de haber supuesto la existencia de la MT H que decide A_{TM} . Por lo tanto, el lenguaje A_{TM} no es decidable. $\square$

Una consecuencia inmediata es que el complemento de A_{TM} , es decir el lenguaje $\overline{A_{TM}} = \{\omega : \omega \text{ no codifica un par } \langle M, \omega' \rangle \text{ o } \omega' \notin L(M)\}$, es un lenguaje que no es ni siquiera Turing reconocible. De hecho,

Teorema 2. Sea L un lenguaje. Si L y su complemento $\overline{L}$ son ambos Turing reconocibles, entonces L y su complemento son ambos Turing decidibles.

Demostración. Sea L y $\overline{L}$ lenguajes Turing reconocibles, y sea M y M' MTs que reconocen L y $\overline{L}$ respectivamente. Construimos una MT D como sigue.

MT D : En entrada ω :

1. Simular de forma *concurrente* M y M' sobre entrada ω . La simulación concurrente la podemos hacer sobre dos cintas distintas. En cada paso de la simulación, se simula un paso de M y un paso de M' . Como M y M' reconocen L y $\overline{L}$ respectivamente, entonces una de las dos simulaciones *debe terminar*. Es decir, ambas simulaciones no pueden ejecutar por siempre sin terminar.
2. Si M acepta, ACEPTAR. Si M' acepta, RECHAZAR.

Claramente, D reconoce L y termina en todas las entradas. Por lo tanto, D decide L . De forma similar podemos construir un MT que decide $\overline{L}$. $\square$

Ejercicios

- 2.a) Muestre que $A = \{\langle R, S \rangle : R \text{ y } S \text{ son expresiones regulares y } L(R) \subseteq L(S)\}$ es decidable.
- 2.b) Muestre que $A = \{\langle M \rangle : M \text{ es un AFD y } L(M) \text{ contiene una palabra } \omega \text{ tal que } \omega = \omega^R\}$ es decidable.
- 2.c) Sea C un lenguaje. Muestre que C es Turing reconocible si y sólo si existe un lenguaje decidable D tal que $C = \{x : \exists y. \langle x, y \rangle \in D\}$.
- 2.d) Utilice el problema anterior para mostrar que

$$A = \{\langle G \rangle : G \text{ es una gramática libre de contexto y } G \text{ es ambigua}\}$$

es Turing reconocible.

3. Reducciones

Teorema 3. El lenguaje $HALT_{TM} = \{\langle M, \omega \rangle : \langle M \rangle \text{ es una MT y termina en entrada } \omega\}$ es indecidible. El lenguaje $HALT_{TM}$ es conocido como el **problema de parada**.

Demostración. Demostraremos que $HALT_{TM}$ es indecidible por el método de reducción: si suponemos que es decidable, entonces podemos probar que A_{TM} es decidable. Como sabemos que A_{TM} no es decidable, entonces $HALT_{TM}$ no puede ser decidable.

Suponga que $HALT_{TM}$ es decidable y sea R una MT que decide $HALT_{TM}$. Construimos la MT S de la siguiente forma.

MT S : En entrada $\langle M, \omega \rangle$ donde $\langle M \rangle$ codifica una MT y ω es una palabra:

1. Ejecutar R en entrada $\langle M, \omega \rangle$.
2. Si R rechaza, RECHAZAR.
3. Si R acepta, ejecutar M en entrada ω . Si M acepta, ACEPTAR. Si M rechaza, RECHAZAR.

Observe que S termina en todas las entradas ya que R termina en todas las entradas por suposición, y la ejecución de M en entrada ω está garantizada de terminar ya que $\langle M, \omega \rangle \in \text{HALT}_{\text{TM}}$. Por construcción, si S acepta, entonces M acepta ω ; si M acepta ω , entonces M termina en ω (y por lo tanto $\langle M, \omega \rangle \in \text{HALT}_{\text{TM}}$) y S acepta $\langle M, \omega \rangle$. Por lo tanto, S decide A_{TM} contradiciendo el hecho que A_{TM} es indecidible. La contradicción viene de suponer que HALT_{TM} es decidable. En consecuencia, HALT_{TM} es indecidible. $\square$

Teorema 4. *El lenguaje $E_{\text{TM}} = \{\langle M \rangle : \langle M \rangle \text{ es una MT y } L(M) = \emptyset\}$ es indecidible.*

Demostración. Suponga que E_{TM} es decidable y sea R una MT que decide E_{TM} . Utilizaremos R para construir una MT S que decide A_{TM} . Para hacer esta construcción, dada una codificación $\langle M \rangle$ de una máquina M y la palabra ω debemos computar la codificación $\langle M_\omega \rangle$ de la MT que hace lo siguiente.

MT M_ω : En entrada ω' :

1. Si $\omega' \neq \omega$, RECHAZAR.
2. Si $\omega' = \omega$, correr M en entrada ω y aceptar/rechazar si M acepta/rechaza ω .

Observe $L(M_\omega) = L(M) \cap \{\omega\}$. Por lo tanto, $L(M_\omega)$ es vacío si y sólo si M rechaza ω . Aunque no se dan los detalles, debe convencerse que construir $\langle M_\omega \rangle$ a partir de $\langle M, \omega \rangle$ es perfectamente factible. La máquina S se describe a continuación.

MT S : En entrada $\langle M, \omega \rangle$ donde $\langle M \rangle$ es la descripción de una MT y ω es una palabra:

1. Construir $\langle M_\omega \rangle$ a partir de $\langle M, \omega \rangle$.
2. Ejecutar la MT R que decide E_{TM} sobre $\langle M_\omega \rangle$.
3. Si R acepta, RECHAZAR. Si R rechaza, ACEPTAR.

Claramente, S termina en todas las entradas y por lo tanto decide el lenguaje que reconoce. Si M acepta ω , entonces S acepta $\langle M, \omega \rangle$ ya que $L(M_\omega) \neq \emptyset$. Si M no acepta ω , entonces $L(M_\omega) = \emptyset$ y S rechaza $\langle M, \omega \rangle$. Por lo tanto, S decide A_{TM} contradiciendo la indecidibilidad de A_{TM} . En consecuencia, E_{TM} es indecidible. $\square$

Teorema 5. *El lenguaje $EQ_{\text{TM}} = \{\langle M_1, M_2 \rangle : \langle M_1 \rangle \text{ y } \langle M_2 \rangle \text{ son MTs y } L(M_1) = L(M_2)\}$ es indecidible.*

Demostración. Suponga que EQ_{TM} es decidable y sea R una MT que decide EQ_{TM} . Construimos una MT S que decide E_{TM} . Sea $M_\emptyset$ la MT que acepta el lenguaje vacío; e.g., $M_\emptyset$ lo único que hace es entrar al estado q_{reject} . La máquina S hace lo siguiente:

MT S : En entrada $\langle M \rangle$ donde $\langle M \rangle$ es la descripción de una MT:

1. Ejecutar R en entrada $\langle M, M_\emptyset \rangle$.
2. Si R acepta, ACEPTAR. Si R rechaza, RECHAZAR.

Claramente, S decide E_{TM} contradiciendo el hecho que E_{TM} es indecidible. Por lo tanto, EQ_{TM} no es decidable. $\square$

En los Teoremas 3, 4 y 5 hemos utilizado el método de reducción para probar que ciertos lenguajes son indecidibles. Ahora, definimos un tipo de reducción simple llamada reducción de mapeo (*mapping reducibility*).

Una función $f : \Sigma^* \rightarrow \Sigma^*$ que mapea palabras en palabras es **computable** si existe una máquina de Turing M que en entrada ω , termina la ejecución con $f(\omega)$ en la cinta. En esta definición no importa si la máquina termina en un estado de aceptación o rechazo. Lo único que importa es que la máquina termine y que al terminar el contenido de la cinta sea $f(\omega)$.

Definición 6. Un lenguaje A es m -reducible a un lenguaje B , denotado por $A \leq_m B$, si y sólo si existe una función computable f tal que para toda palabra ω :

$$\omega \in A \iff f(\omega) \in B.$$

La función f se llama una reducción de A en B .

Esta noción formal resume algunas de las estrategias que hemos usado hasta ahora para demostrar la indecidibilidad de lenguajes. Sin embargo, también podemos utilizar este tipo de reducciones de las siguientes formas:

- A) Si $A \leq_m B$ y A es indecidible, entonces B es indecidible.
- B) Si $A \leq_m B$ y B es Turing reconocible, entonces A es Turing reconocible.
- C) Si $A \leq_m B$ y A no es Turing reconocible, entonces B no es Turing reconocible.

Teorema 7. Los lenguajes EQ_{TM} y $\overline{EQ_{TM}}$ no son Turing reconocibles.

Demostración. Para mostrar que EQ_{TM} no es Turing reconocible, mostramos que $\overline{A_{TM}}$ es m -reducible a EQ_{TM} . Entonces, utilizando el hecho que $\overline{A_{TM}}$ no es Turing reconocible y (C), podemos concluir que EQ_{TM} no es Turing reconocible.

Por definición, $\overline{A_{TM}} \leq_m EQ_{TM}$ es equivalente a $A_{TM} \leq_m \overline{EQ_{TM}}$. La función f que reduce A_{TM} en $\overline{EQ_{TM}}$ hace lo siguiente.

MT f : En entrada $\langle M, \omega \rangle$ donde $\langle M \rangle$ es la descripción de una MT y ω es una palabra:

1. Construir las siguientes dos MTs:
 - M_1 : rechaza todas las entradas.
 - M_2 : en entrada ω' . Correr M en entrada ω . Si M acepta, ACEPTAR.
2. Escribir en la cinta $\langle M_1, M_2 \rangle$.

Veamos que f es la reducción deseada. Considere $\langle M, \omega \rangle$. Entonces,

$$\begin{aligned} \langle M, \omega \rangle \in A_{\text{TM}} &\implies L(M_1) = \emptyset \wedge L(M_2) = \Sigma^* \implies L(M_1) \neq L(M_2) \implies \langle M_1, M_2 \rangle \in \overline{EQ_{\text{TM}}}, \\ \langle M, \omega \rangle \notin A_{\text{TM}} &\implies L(M_1) = \emptyset \wedge L(M_2) = \emptyset \implies L(M_1) = L(M_2) \implies \langle M_1, M_2 \rangle \notin \overline{EQ_{\text{TM}}}. \end{aligned}$$

Para mostrar que $\overline{EQ_{\text{TM}}}$ no es Turing reconocible, mostramos que A_{TM} es m -reducible a EQ_{TM} y utilizamos (C). La función g que reduce A_{TM} en EQ_{TM} hace lo siguiente.

MT g : En entrada $\langle M, \omega \rangle$ donde $\langle M \rangle$ es la descripción de una MT y ω es una palabra:

1. Construir las siguientes dos MTs:
 - M_1 : acepta todas las entradas.
 - M_2 : en entrada ω' . Correr M en entrada ω . Si M acepta, ACEPTAR.
2. Escribir en la cinta $\langle M_1, M_2 \rangle$.

Veamos que g es la reducción deseada. Considere $\langle M, \omega \rangle$. Entonces,

$$\begin{aligned} \langle M, \omega \rangle \in A_{\text{TM}} &\implies L(M_1) = \Sigma^* \wedge L(M_2) = \Sigma^* \implies L(M_1) = L(M_2) \implies \langle M_1, M_2 \rangle \in EQ_{\text{TM}}, \\ \langle M, \omega \rangle \notin A_{\text{TM}} &\implies L(M_1) = \Sigma^* \wedge L(M_2) = \emptyset \implies L(M_1) \neq L(M_2) \implies \langle M_1, M_2 \rangle \notin EQ_{\text{TM}}. \end{aligned}$$

Por lo tanto, $A_{\text{TM}} \leq_m EQ_{\text{TM}}$. Esto es equivalente a $\overline{A_{\text{TM}}} \leq_m \overline{EQ_{\text{TM}}}$. Utilizando (C) y el hecho que $\overline{A_{\text{TM}}}$ no es Turing reconocible, concluimos que $\overline{EQ_{\text{TM}}}$ no es Turing reconocible. $\square$

Ejercicios

3.a) Muestre las propiedades A, B y C.

3.b) Muestre que $A = \{\langle M \rangle : M \text{ es una MT y } M \text{ acepta } \omega^R \text{ si } M \text{ acepta } \omega\}$ es indecidible.

3.c) Muestre que A es Turing reconocible si y sólo si $A \leq_m A_{\text{TM}}$.

3.d) Sea A un lenguaje decidable diferente a $\emptyset$ y Σ^* . Muestre que para todo lenguaje B , B es decidable si y sólo si $B \leq_m A$.

3.1. Teorema de Rice

El Teorema de Rice nos permite concluir rápidamente que ciertos lenguajes no son decidibles. Los lenguajes en los que se enfoca el Teorema de Rice son aquellos lenguajes L que consisten de descripciones $\langle M \rangle$ de máquinas de Turing y que satisfacen la propiedad: si $\langle M \rangle \in L$ y M' es una MT equivalente a M , entonces $\langle M' \rangle \in L$.

Una propiedad P sobre las máquinas de Turing es un subconjunto de descripciones $\langle M \rangle$ de MTs. Una MT M satisface la propiedad P si y sólo si $\langle M \rangle \in P$. Decimos que la propiedad P es cerrada bajo equivalencias si para toda MT M tal que $\langle M \rangle \in P$ y toda MT M' que sea equivalente a M , se cumple $\langle M' \rangle \in P$. Por ejemplo, $P = "M \text{ tiene 10 estados}"$ es una propiedad sobre las máquinas de Turing pero no es una propiedad cerrada bajo equivalencias ya que para toda máquina de Turing M con n estados y todo entero $m \geq n$, existe una MT M' con m estados que es equivalente a M . La propiedad $P = "M \text{ reconoce el lenguaje } 0^n 1^n"$ es una propiedad cerrada bajo equivalencias. Decimos que una propiedad P es trivial si y sólo si P contiene todas las máquinas de Turing ó P no contiene ninguna máquina de Turing.

Teorema 8 (Rice). Sea P una propiedad sobre máquinas de Turing cerrado bajo equivalencias. Si P es no trivial, entonces P es indecidible.

Demostración. Demostración por el método de reducción, usando m -reducciones. Sea $M_\emptyset$ la MT que reconoce el lenguaje vacío. Sin pérdida de generalidad, suponemos que $\langle M_\emptyset \rangle \notin P$. (Si esto no se cumple, trabajamos con $\bar{P}$ para demostrar que $\bar{P}$ es indecidible lo que implica que P también es indecidible.) Como P es no trivial, existe una MT M^* tal que $\langle M^* \rangle \in P$.

Suponga ahora que P es decidible y sea R una MT que decide P . Mostraremos que A_{TM} es decidible. Considere la máquina de Turing S :

MT S : En entrada $\langle M, \omega \rangle$ donde $\langle M \rangle$ es la descripción de una MT y ω es una palabra:

1. Construir la siguiente MT M_ω : En entrada ω' .
 - a) Simular M en entrada ω . Si M termina y rechaza, RECHAZAR.
 - b) Simular M^* en entrada ω' . Si M^* acepta, ACEPTAR.
2. Ejecutar R en entrada $\langle M_\omega \rangle$ para determinar si $\langle M_\omega \rangle \in P$. Si $\langle M_\omega \rangle \in P$, ACEPTAR. En caso contrario, RECHAZAR.

Observe que M_ω simula M^* en entrada ω' si M acepta ω . Por lo tanto, $L(M_\omega) = L(M^*)$ si M acepta ω , y $L(M_\omega) = L(M_\emptyset) = \emptyset$ si M no acepta ω (i.e., si M rechaza o no termina en entrada ω). Entonces, $\langle M_\omega \rangle \in P$ si y sólo si $\langle M, \omega \rangle \in A_{\text{TM}}$. Por otro lado, la MT S termina en todas las entradas ya que R termina en todas las entradas, y la máquina M_ω es construible a partir de $\langle M, \omega \rangle$. Entonces, S decide A_{TM} en contradicción con el hecho que A_{TM} es indecidible. La contradicción viene de haber supuesto que P es decidible. $\square$

Ejercicios

Para los siguientes ejercicios utilice el Teorema de Rice.

- 4.a) Muestre que $A = \{\langle M \rangle : M \text{ es una MT y } M \text{ acepta } \omega^R \text{ si } M \text{ acepta } \omega\}$ es indecidible.
- 4.b) Muestre que $ALL_{\text{TM}} = \{\langle M \rangle : M \text{ es una MT y } L(M) = \Sigma^*\}$ es indecidible.