



Problemas Insolubles

Ejemplos

A continuación enunciaremos una serie de problemas y demostraremos, mediante la técnica de reducibilidad, que los mismos son indecidibles (no solubles a través de una máquina de Turing o de un procedimiento efectivo)

A través de **Pdet**, haremos referencia al problema de la detención, ya conocido:

Pdet: ¿Existe un procedimiento efectivo capaz de determinar si una máquina de Turing se detiene sobre una determinada cadena α ?

Ahora consideremos los problemas que queremos demostrar insolubles.

Ejemplo 1:

Sea el problema **Paccept** el siguiente problema:

Paccept: ¿Existe un procedimiento efectivo capaz de determinar si una máquina de Turing acepta una determinada cadena α ?

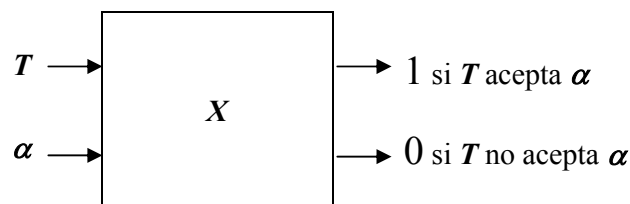
Otra manera de formular el problema sería:

Paccept: ¿Existe un procedimiento efectivo capaz de determinar si una determinada cadena, α , pertenece al lenguaje que reconoce la máquina de Turing T ?

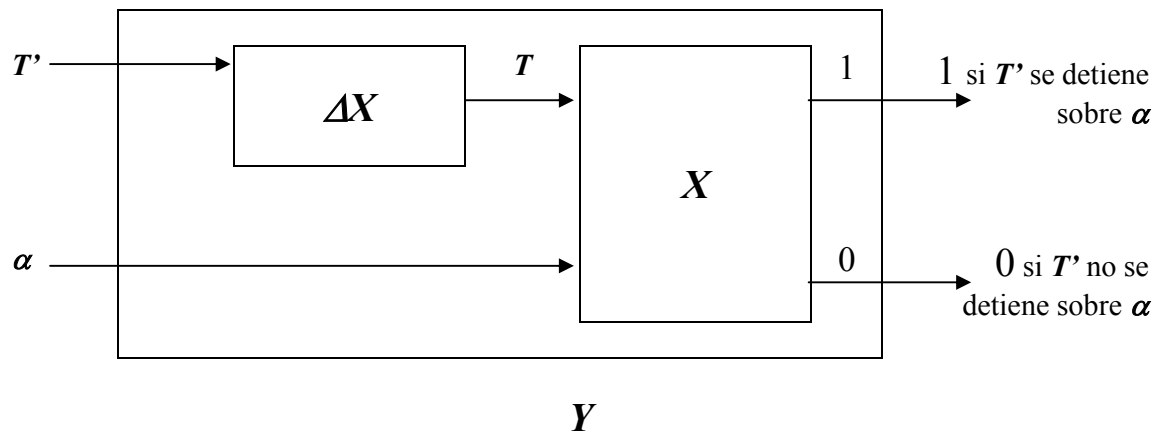
Para demostrar que **Paccept** no es soluble, comenzaremos suponiendo que lo es, llegando de esta manera a un absurdo. Este absurdo tendrá lugar debido a que la solubilidad de **Paccept** implica la solubilidad de **Pdet**. Expresado de otra manera, demostramos que **Pdet** se reduce a **Paccept**.

Demostración:

Supongamos que **Paccept** es un problema de decisión soluble. Al ser un problema soluble, entonces existe un procedimiento efectivo (o máquina universal de Turing), X , que resuelve **Paccept**. X toma como datos de entrada la descripción de una máquina de Turing T y una cadena α y determina en un tiempo finito si T acepta o no a la cadena α . Es decir X recibe como entrada al par (T, α) y retorna un **1** si T acepta α , mientras que devuelve la salida **0** si T no acepta α .



Construyamos a partir de X , un procedimiento efectivo (o máquina universal de Turing) Y



La máquina **Y** recibe como entrada un par (T', α) , la cadena de ambos problemas es la misma, lo que necesitamos es un proceso adicional que modifique T' de manera que las respuestas de X , respondan el problema de la detención. Este proceso adicional se lleva a cabo en la máquina universal ΔX que realiza lo siguiente:

- Recibe como dato de entrada la máquina de Turing T' .
- Modifica T' manteniendo su definición de quintuplas pero indicando que **todos** los estados son aceptadores.
- Retorna la máquina modificada con el nombre T .

Combinando las máquinas X y ΔX , tenemos la máquina universal **Y** que tiene el siguiente comportamiento:

- Y** recibe como entrada el par (T', α) .
- La máquina universal **Y** utiliza ΔX , la cual a partir de T' construye T .
- El par (T, α) es ingresado a la máquina universal X .
- Si la respuesta de X es 1 entonces T acepta α pero entonces T' se detiene sobre α . Dado que T se comporta como T' solo que siempre que se detiene acepta, podemos afirmar que T' se detiene sobre α y que por lo tanto **Y** emite un 1
- Si la respuesta de X es 0 entonces T no acepta α pero entonces T' no se detiene sobre α . Dado que T tiene a todos sus estados como aceptadores, significa que para no aceptar la cadena, la única posibilidad es que T no se haya detenido. Como T se comporta como T' , podemos afirmar que esta tampoco se detiene y por lo tanto **Y** emite un 0, ya que T' no se detiene sobre α .

Conclusión:

- La máquina universal **Y** retorna 1 (T' se detiene sobre α) si X retorna 1 (T acepta α).
- La máquina universal **Y** retorna 0 (T' no se detiene sobre α) si X retorna 0 (T no acepta α).

Hemos mostrado como construir la máquina universal **Y** que resuelve el problema de la detención a partir de la máquina universal X que resuelve un problema que supusimos soluble.

Sabemos por hipótesis que el problema de la detención es un problema insoluble, por lo tanto la solución encontrada mediante la máquina universal Y no puede existir. Lo que implica que alguna de sus componentes no puede existir, es decir o bien X , o bien ΔX no existe. Como por construcción ΔX existe, luego X no.

Podemos entonces concluir que **Paccept** es un problema no-soluble.

Ejemplo 2:

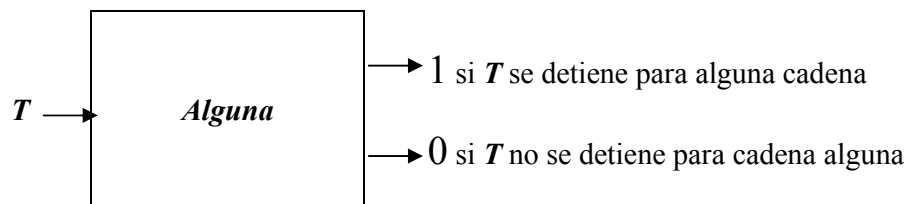
Sea el problema **Palguna** el siguiente problema:

Paccept: ¿Existe un procedimiento efectivo capaz de determinar si una máquina de Turing se detiene sobre alguna cadena?

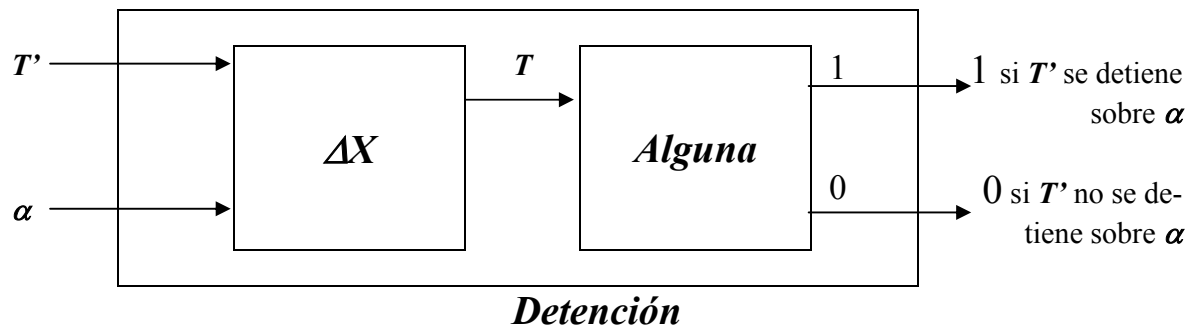
Para demostrar que **Palguna** no es soluble, comenzaremos suponiendo que lo es, llegando de esta manera a un absurdo. Este absurdo tendrá lugar debido a que la solubilidad de **Palguna** implica la solubilidad de **Pdet**. Expresado de otra manera, demostramos que **Pdet** se reduce a **Palguna**.

Demostración:

Supongamos que **Palguna** es un problema de decisión soluble. Al ser un problema soluble, entonces existe un procedimiento efectivo (o máquina universal de Turing), **Alguna**, que resuelve **Palguna**. **Alguna** toma como datos de entrada la descripción de una máquina de Turing T y determina en un tiempo finito si T se detiene sobre alguna cadena o no. Es decir **Alguna** recibe como entrada (T) y retorna un **1** si T se detiene para alguna cadena, mientras que devuelve la salida **0** si T no se detiene para ninguna cadena.



Construyamos a partir de **Alguna**, un procedimiento efectivo (o máquina universal de Turing) **Detención**



La máquina **Detención** recibe como entrada un par (T', α) , el problema **Palguna** solo tiene como entrada una máquina de Turing, por lo que necesitamos un proceso adicional que



combine T' y α de manera que las respuestas de *Alguna*, respondan el problema de la detención. Este proceso adicional se lleva a cabo en la máquina universal ΔX que realiza lo siguiente:

- a. Recibe como dato de entrada la máquina de Turing T' y la cadena α .
- b. Construye una máquina de Turing T tal que:
 - i. Para la cadena α se comporta como la máquina de Turing T' .
 - j. Para toda cadena que no sea α la nueva máquina T nunca se detiene o cicla indefinidamente.

Combinando la máquinas universales *Alguna* y ΔX , tenemos la máquina universal *Detención* que tiene el siguiente comportamiento:

1. *Detención* recibe como entrada el par (T', α) .
2. La máquina universal *Detención* utiliza ΔX , la cual a partir de T' y α construye la nueva máquina de Turing T .
3. La máquina de Turing T obtenida en el paso anterior es ingresada a la máquina universal *Alguna*.
4. Si la respuesta de *Alguna* es 1 entonces T se detiene para alguna cadena. De la manera en la que construimos T esa cadena necesariamente es la cadena α (ya que para el resto sabemos que la máquina siempre cicla) como el comportamiento de T frente a la cadena α es el mismo que tiene la máquina T' entonces podemos afirmar que T' se detiene sobre α y que por lo tanto *Detención* emite un 1
5. Si la respuesta de *Alguna* es 0 entonces T no se detiene para ninguna cadena. De la manera en la que construimos T la única cadena sobre la que T podía detenerse era α y que el comportamiento frente a esta cadena era el mismo que el de la máquina T' . Podemos entonces afirmar, que la máquina de Turing T' tampoco se detiene frente a la cadena α y que por lo tanto la respuesta que emite *Detención* es igual a 0.

Conclusión:

- La máquina universal *Detención* retorna 1 (T' se detiene sobre α) si *Alguna* retorna 1 (T se detiene sobre alguna cadena).
- La máquina universal *Detención* retorna 0 (T' no se detiene sobre α) si *Alguna* retorna 0 (T se detiene sobre ninguna cadena).

Hemos mostrado como construir la máquina universal *Detención* que resuelve el problema de la detención a partir de la máquina universal *Alguna* que resuelve un problema que supusimos soluble.

Sabemos por hipótesis que el problema de la detención es un problema insoluble, por lo tanto la solución encontrada mediante la máquina universal *Detención* no puede existir. Lo que implica que alguna de sus componentes no puede existir, es decir o bien *Alguna*, o bien ΔX no existe. Como por construcción ΔX existe, luego *Alguna* no puede existir.

Podemos entonces concluir que *Palguna* es un problema no-soluble.

Ejemplo 3:

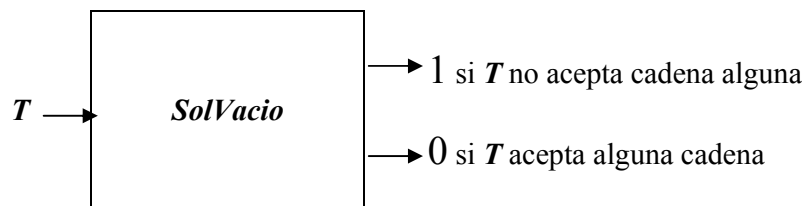
Sea el problema *Pvacío* el siguiente problema:

Pvacio: ¿Existe un procedimiento efectivo capaz de determinar si una máquina de Turing no acepta ninguna cadena?

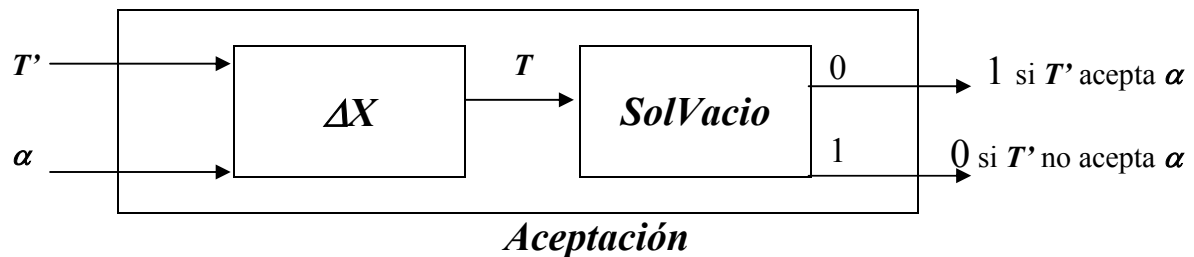
En esta caso tomaremos como el problema insoluble a reducir a este al problema de la aceptación **Paccept**, demostrado como insoluble anteriormente.

Demostración:

Supongamos que **Pvacio** es un problema de decisión soluble. Al ser un problema soluble, entonces existe un procedimiento efectivo (o máquina universal de Turing), **SolVacio**, que resuelve **Pvacio**. **SolVacio** toma como datos de entrada la descripción de una máquina de Turing **T** y determina en un tiempo finito si **T** no acepta cadena alguna. Es decir **SolVacio** recibe como entrada **T** y retorna un **1** si **T** no acepta cadena alguna, mientras que devuelve la salida **0** si **T** acepta alguna cadena.



Construyamos a partir de **SolVacio**, un procedimiento efectivo (o máquina universal de Turing) **Aceptación**



La máquina **Aceptación** recibe como entrada un par (T', α) , el problema **Pvacio** solo tiene como entrada una máquina de Turing, por lo que necesitamos un proceso adicional que combine **T'** y α de manera que las respuestas de **SolVacio**, respondan el problema de la aceptación. Este proceso adicional se lleva a cabo en la máquina universal ΔX que realiza lo siguiente:

- a. Recibe como dato de entrada la máquina de Turing **T'** y la cadena α .
- b. Construye una máquina de Turing **T** tal que:
 - i. Para la cadena α se comporta como la máquina de Turing **T'**.
 - j. Para toda cadena que no sea α la nueva máquina **T** nunca se detiene o cicla indefinidamente.

Combinando la máquinas universales **SolVacio** y ΔX , tenemos la máquina universal **Aceptación** que tiene el siguiente comportamiento:



1. **Aceptación** recibe como entrada el par (T', α) .
2. La máquina universal **Aceptación** utiliza ΔX , la cual a partir de T' y α construye la nueva máquina de Turing T .
3. La máquina de Turing T obtenida en el paso anterior es ingresada a la máquina universal **SolVacio**.
4. Si la respuesta de **SolVacio** es 1 entonces T no acepta cadena alguna. De la manera en la que construimos T la única cadena sobre la que había incertidumbre con respecto a la aceptación o no es α (ya que para el resto sabemos que la máquina siempre cicla). Como el comportamiento de T frente a la cadena α es el mismo que tiene la máquina T' entonces podemos afirmar que T' no acepta la cadena α ya que de lo contrario habría una cadena que T aceptaría. Por lo tanto **Aceptación** debe emitir un 0.
5. Si la respuesta de **SolVacio** es 0 entonces T acepta alguna cadena. De la manera en la que construimos T la única cadena que pudo ser aceptada es α . Como la aceptación de α depende de T' (debido a la manera en que T fue construida) podemos asegurar que T' acepta α y por lo tanto la máquina universal **Aceptación** retorna el valor 1.

Conclusión:

- La máquina universal **Aceptación** retorna 1 (T' acepta α) si **SolVacio** retorna 0 (T acepta alguna cadena).
- La máquina universal **Aceptación** retorna 0 (T' no acepta α) si **SolVacio** retorna 1 (T no acepta cadena alguna).

Hemos mostrado como construir la máquina universal **Aceptación** que resuelve el problema de la aceptación (**Paccept**) a partir de la máquina universal **SolVacio** que resuelve un problema **Pvacio**.

Sabemos por hipótesis que el problema de la aceptación es un problema insoluble, por lo tanto la solución encontrada mediante la máquina universal **Aceptación** no puede existir. Lo que implica que alguna de sus componentes no puede existir, es decir o bien **SolVacio**, o bien ΔX no existe. Como por construcción ΔX existe, luego **SolVacio** no puede existir y por lo tanto **Pvacio** es un problema insoluble.