Límites del Cálculo Algorítmico

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Lenguajes Formales y Autómatas CCOS 014

Contenido

Motivación I

- Hemos hablado sobre lo que pueden hacer las máquinas de Turing, ahora veremos lo que no pueden hacer.

Motivación I

- Aunque la tesis de Turing nos lleva a creer que existen pocas limitaciones en el poder de una máquina de Turing, hemos afirmado en varias ocasiones que no podría existir ningún algoritmo para la solución de ciertos problemas.

Motivación I

- Ahora hacemos más explícito lo que queremos decir con esta afirmación.

Motivación I

- Algunos de los resultados se obtuvieron de forma bastante sencilla; si un lenguaje no es recursivo, entonces, por definición, no existe un algoritmo de pertenencia para él.

Motivación I

- Si esto fuera todo lo que hay en este tema, no sería muy interesante; los lenguajes no recursivos tienen poco valor práctico.

Motivación II

- Pero el problema es más profundo.

Motivación II

- Por ejemplo, hemos declarado (pero aún no probado) que no existe un algoritmo para determinar si una gramática libre de contexto no es ambigua.

Motivación II

- Esta pregunta es claramente de importancia práctica en el estudio de los lenguajes de programación.

Motivación II

- Primero definimos los conceptos de decidibilidad y computabilidad para precisar lo que queremos decir cuando decimos que una máquina de Turing no puede hacer algo.

Motivación II

- A continuación, examinamos varios problemas clásicos de este tipo, entre ellos el conocido problema de paro de las máquinas de Turing.

Motivación II

- De esto se siguen una serie de problemas relacionados con las máquinas de Turing y los lenguajes enumerables de forma recursiva.

Computabilidad y decidibilidad I

- El argumento de que el poder de los cálculos mecánicos es limitado no es sorprendente.

Computabilidad y decidibilidad I

- Intuitivamente sabemos que muchas preguntas vagas y especulativas requieren una comprensión y un razonamiento especiales mucho más allá de la capacidad de cualquier computadora que ahora podamos construir o incluso prever.

Computabilidad y decidibilidad I

- Lo que es más interesante para los científicos de la computación es que hay cuestiones que pueden plantearse de forma clara y sencilla, con una aparente posibilidad de una solución algorítmica, pero que se sabe que no pueden ser resueltas por ninguna computadora.

Computabilidad y decidibilidad I

- Dijimos que se dice que una función f en un dominio determinado es computable si existe una máquina de Turing que calcula el valor de f para todos los argumentos en su dominio.

Computabilidad y decidibilidad I

- Una función no es computable si no existe tal máquina de Turing.

Computabilidad y decidibilidad II

- Puede haber una máquina de Turing que pueda calcular f en parte de su dominio, pero llamamos a la función computable sólo si hay una máquina de Turing que calcula la función en la totalidad de su dominio.

Computabilidad y decidibilidad II

- Vemos de esto que, cuando clasificamos una función como computable o no computable, debemos tener claro cuál es su dominio.

Computabilidad y decidibilidad II

- Nuestra preocupación aquí será la configuración algo simplificada donde el resultado de un cálculo es un simple “sí” o “no”.

Computabilidad y decidibilidad II

- En este caso, hablamos de que un problema es **decidible** o **indecidible**.

Computabilidad y decidibilidad II

- Por problema entenderemos un conjunto de afirmaciones relacionadas, cada una de las cuales debe ser verdadera o falsa.

Computabilidad y decidibilidad II

- Por ejemplo, consideramos la afirmación "Para una gramática G libre de contexto, el lenguaje $L(G)$ es ambiguo".

Computabilidad y decidibilidad II

- Para algunas G esto es cierto, para otras es falso, pero claramente debemos tener uno u otro. El problema es decidir si el enunciado es verdadero para cualquier G que se nos dé.

Computabilidad y decidibilidad III

- Nuevamente, hay un dominio subyacente, el conjunto de todas las gramáticas libres de contexto.

Computabilidad y decidibilidad III

- Decimos que un problema es decidible si existe una máquina de Turing que da la respuesta correcta para cada enunciado en el dominio del problema.

Computabilidad y decidibilidad III

- Cuando enunciamos resultados de decidibilidad o indecidibilidad, siempre debemos saber cuál es el dominio, porque esto puede afectar la conclusión.

Computabilidad y decidibilidad III

- El problema puede resolverse en algún dominio pero no en otro.

Computabilidad y decidibilidad III

- Específicamente, una sola instancia de un problema siempre es decidible, ya que la respuesta es verdadera o falsa.

Computabilidad y decidibilidad III

- En el primer caso, una máquina de Turing que siempre responde "verdadero" da la respuesta correcta, mientras que en el segundo caso es apropiada una que siempre responde "falso".

Computabilidad y decidibilidad III

- Esto puede parecer una respuesta graciosa, pero enfatiza un punto importante.

Computabilidad y decidibilidad III

- El hecho de que no sepamos cuál es la respuesta correcta no hace ninguna diferencia; lo que importa es que existe alguna máquina de Turing que da la respuesta correcta.

El problema de paro de la Máquina de Turing I

- Partimos de algunos problemas que tienen trascendencia histórica y que a la vez nos dan un punto de partida para desarrollar resultados posteriores.

El problema de paro de la Máquina de Turing I

- El más conocido de ellos es el problema de paro de la máquina de Turing.

El problema de paro de la Máquina de Turing I

- En pocas palabras, el problema es: Dada la descripción de una máquina de Turing M y una entrada w, ¿M, cuando se inicia en la configuración inicial q_0w, realiza un cálculo que finalmente se detiene?

El problema de paro de la Máquina de Turing I

- Usando una forma abreviada de hablar sobre el problema, preguntamos si M aplicada a w, o simplemente (M, w), se detiene o no.

El problema de paro de la Máquina de Turing I

- El dominio de este problema debe tomarse como el conjunto de todas las máquinas de Turing y todas las w; es decir, estamos buscando una sola máquina de Turing que, dada la descripción de una M y w arbitrarias, predecirá si el cálculo de M aplicada a w se detendrá o no.

El problema de paro de la Máquina de Turing II

- No podemos encontrar la respuesta simulando la acción de M sobre w, digamos realizándola en una máquina de Turing universal, porque no hay límite en la duración del cálculo.

El problema de paro de la Máquina de Turing II

- Si M entra en un ciclo infinito, no importa cuánto esperemos, nunca podremos estar seguros de que M está de hecho en un ciclo.

El problema de paro de la Máquina de Turing II

- Puede ser simplemente un caso de un cálculo muy largo.

El problema de paro de la Máquina de Turing II

- Lo que necesitamos es un algoritmo que pueda determinar la respuesta correcta para cualquier M y w realizando un análisis de la descripción de la máquina y la entrada.

El problema de paro de la Máquina de Turing II

- Pero, como mostraremos, no existe tal algoritmo.

El problema de paro de la Máquina de Turing II

- Para una discusión posterior, es conveniente tener una idea precisa de lo que entendemos por problema de paro; por esta razón, hacemos una definición específica de lo que dijimos en forma un tanto vaga.

El problema de paro de la Máquina de Turing III

Definición 1

Sea w_M una cadena que describe una máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, y sea w una cadena en el alfabeto de M. Supondremos que w_M y w están codificados como una cadena de ceros y unos, como se sugiere en la sección donde hablamos de máquina de Turing Universal. Una solución del problema de paro es una máquina de Turing H, que para cualquier w_M y w realiza el cálculo

$$q_0 w_M w \overset{*}{\vdash} x_1 q_y x_2$$

si M aplicada a w para, y

$$q_0 w_M w \overset{*}{\vdash} y_1 q_n y_2$$

si M aplicada a w no para. Aquí q_y y q_n son estados finales de H.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

Demostración:

- Suponemos lo contrario, es decir, que existe un algoritmo y, en consecuencia, alguna máquina de Turing H, que resuelve el problema de paro.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

Demostración:

- La entrada a H será la cadena $w_M w$.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

Demostración:

- El requisito es entonces que, dado cualquier $w_M w$, la máquina de Turing H se detendrá con una respuesta de sí o no.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

Demostración:

- Logramos esto pidiendo que H se detenga en uno de los dos estados finales correspondientes, digamos, q_y o q_n.

El problema de paro de la Máquina de Turing IV

Teorema 1

No existe ninguna máquina de Turing H que se comporte como lo requiere la Definición 1. Por tanto, el problema de paro es indecidible.

Demostración:

- La situación se puede visualizar mediante un diagrama de bloques como el de la Figura 1.

El problema de paro de la Máquina de Turing V

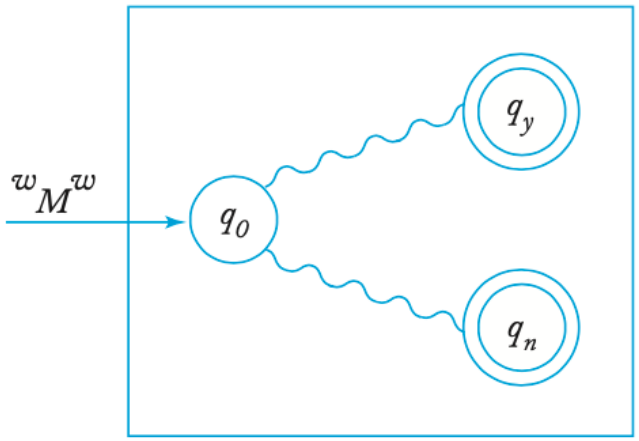

Figure 1: H se inicia en el estado q_0 con la entrada $w_M w$ y eventualmente se detiene en el estado q_y o q_n.

- La intención de este diagrama es indicar que, si H se inicia en el estado q_0 con la entrada $w_M w$, eventualmente se detendrá en el estado q_y o q_n.

El problema de paro de la Máquina de Turing VI

- Como lo requiere la Definición 1, queremos que H opere de acuerdo con las siguientes reglas:

El problema de paro de la Máquina de Turing VI

- $$q_0 w_M w \vdash_H^* x_1 q_y x_2$$

  si M aplicada a w para, y

El problema de paro de la Máquina de Turing VI

- $$q_0 w_M w \overset{*}{\vdash_H} y_1 q_n y_2$$

  si M aplicada a w no para.

El problema de paro de la Máquina de Turing VI

- A continuación, modificamos H para producir una máquina de Turing H' con la estructura que se muestra en la Figura 2.

El problema de paro de la Máquina de Turing VII

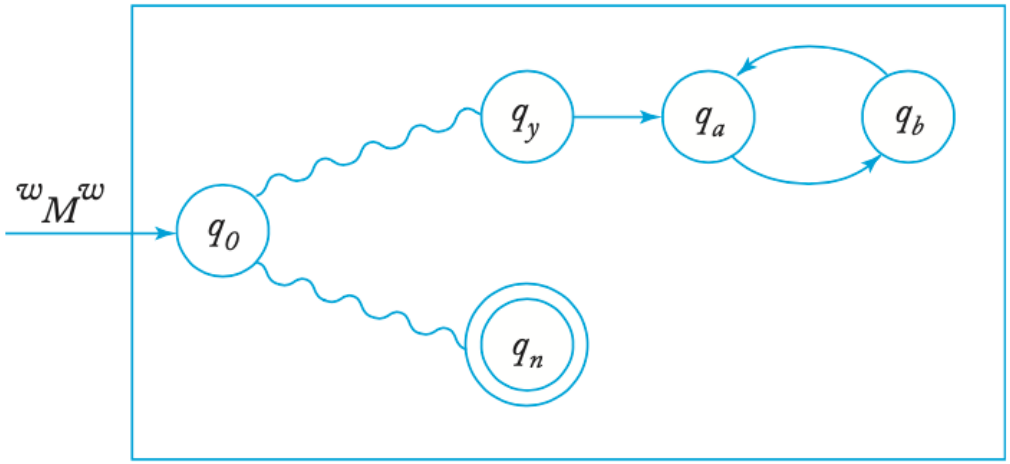

Figure 2: H' se comporta como H excepto que cuando alcanza el estado q_y independientemente de lo que haya en la cinta no lo cambia y se cicla.

- Con los estados agregados en la Figura 2 queremos expresar que las transiciones entre el estado q_y y los nuevos estados q_a y q_b deben realizarse, independientemente del símbolo de la cinta, de tal manera que la cinta permanezca sin cambios.

El problema de paro de la Máquina de Turing VIII

- La forma en que se hace esto es sencilla.

El problema de paro de la Máquina de Turing VIII

- Comparando H y H' vemos que, en situaciones donde H alcanza q_y y se detiene, la máquina modificada H' entrará en un ciclo infinito.

El problema de paro de la Máquina de Turing VIII

- Formalmente, la acción de H' se describe por

$$q_0 w_M w \vdash^*_{H'} \infty$$

si M aplicada a w para, y

El problema de paro de la Máquina de Turing VIII

- $$q_0 w_M w \overset{*}{\vdash}_{H'} y_1 q_n y_2$$

  si M aplicada a w no para.

El problema de paro de la Máquina de Turing VIII

- A partir de H' construimos otra máquina de Turing $\widehat{H}$.

El problema de paro de la Máquina de Turing IX

- Esta nueva máquina toma como entrada w_M y la copia, terminando en su estado inicial q_0.

El problema de paro de la Máquina de Turing IX

- Esta nueva máquina toma como entrada w_M y la copia, terminando en su estado inicial q_0.
- Después de eso, se comporta exactamente como H'.

El problema de paro de la Máquina de Turing IX

- Esta nueva máquina toma como entrada w_M y la copia, terminando en su estado inicial q_0.
- Después de eso, se comporta exactamente como H'.
- Entonces la acción de $\widehat{H}$ es tal que

$$q_0 w_M \vdash^*_{\widehat{H}} q_0 w_M w_M \vdash^*_{\widehat{H}} \infty$$

si M aplicada a w_M para, y

El problema de paro de la Máquina de Turing IX

- Esta nueva máquina toma como entrada w_M y la copia, terminando en su estado inicial q_0.
- Después de eso, se comporta exactamente como H'.
- Entonces la acción de $\widehat{H}$ es tal que

$$q_0 w_M \vdash^*_{\widehat{H}} q_0 w_M w_M \vdash^*_{\widehat{H}} \infty$$

si M aplicada a w_M para, y

- $$q_0 w_M \vdash^*_{\widehat{H}} q_0 w_M w_M \vdash^*_{\widehat{H}} y_1 q_n y_2$$

si M aplicada a w_M no para.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.
- Por lo tanto, podemos preguntarnos legítimamente qué sucedería si $\widehat{H}$ se aplicara a $\widehat{w}$.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.
- Por lo tanto, podemos preguntarnos legítimamente qué sucedería si $\widehat{H}$ se aplicara a $\widehat{w}$.
- De lo anterior, identificando M con $\widehat{H}$, obtenemos

$$q_0\widehat{w} \vdash^*_{\widehat{H}} \infty$$

si $\widehat{H}$ aplicada a $\widehat{w}$ para, y

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.
- Por lo tanto, podemos preguntarnos legítimamente qué sucedería si $\widehat{H}$ se aplicara a $\widehat{w}$.
- De lo anterior, identificando M con $\widehat{H}$, obtenemos

$$q_0\widehat{w} \vdash^*_{\widehat{H}} \infty$$

si $\widehat{H}$ aplicada a $\widehat{w}$ para, y

- $$q_0\widehat{w} \vdash^*_{\widehat{H}} y_1 q_n y_2$$

si $\widehat{H}$ aplicada a $\widehat{w}$ no para.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.
- Por lo tanto, podemos preguntarnos legítimamente qué sucedería si $\widehat{H}$ se aplicara a $\widehat{w}$.
- De lo anterior, identificando M con $\widehat{H}$, obtenemos

$$q_0\widehat{w} \vdash^*_{\widehat{H}} \infty$$

si $\widehat{H}$ aplicada a $\widehat{w}$ para, y

- $$q_0\widehat{w} \vdash^*_{\widehat{H}} y_1 q_n y_2$$

si $\widehat{H}$ aplicada a $\widehat{w}$ no para.

- Esto es claramente un sinsentido.

El problema de paro de la Máquina de Turing X

- Ahora, $\widehat{H}$ es una máquina de Turing, por lo que tiene una descripción en $\{0,1\}^*$, digamos, $\widehat{w}$.
- Esta cadena, además de ser la descripción de $\widehat{H}$, también se puede utilizar como cadena de entrada.
- Por lo tanto, podemos preguntarnos legítimamente qué sucedería si $\widehat{H}$ se aplicara a $\widehat{w}$.
- De lo anterior, identificando M con $\widehat{H}$, obtenemos

$$q_0\widehat{w} \overset{*}{\vdash}_{\widehat{H}} \infty$$

si $\widehat{H}$ aplicada a $\widehat{w}$ para, y

- $$q_0\widehat{w} \overset{*}{\vdash}_{\widehat{H}} y_1 q_n y_2$$

si $\widehat{H}$ aplicada a $\widehat{w}$ no para.

- Esto es claramente un sinsentido.
- La contradicción nos dice que nuestra suposición de la existencia de H, y por tanto la suposición de la decidibilidad del problema de paro, debe ser falsa.

El problema de paro de la Máquina de Turing XI

- Uno puede objetar la Definición 1, ya que requerimos que, para resolver el problema de paro, H tuviera que comenzar y terminar en configuraciones muy específicas.

El problema de paro de la Máquina de Turing XI

- Sin embargo, no es difícil ver que estas condiciones elegidas arbitrariamente juegan sólo un papel menor en el argumento, y que esencialmente el mismo razonamiento podría usarse con cualquier otra configuración inicial y final.

El problema de paro de la Máquina de Turing XI

- Hemos vinculado el problema a una definición específica por el bien de la discusión, pero esto no afecta la conclusión.

El problema de paro de la Máquina de Turing XII

- Es importante tener en cuenta lo que dice el Teorema 1.

El problema de paro de la Máquina de Turing XII

- No excluye la solución del problema de paro para casos específicos; a menudo podemos decir mediante un análisis de M y w si la máquina de Turing parará o no.

El problema de paro de la Máquina de Turing XII

- Lo que dice el teorema es que esto no siempre se puede hacer; no existe un algoritmo que pueda tomar una decisión correcta para todas las w_M y w.

El problema de paro de la Máquina de Turing XII

- Los argumentos para probar el Teorema 1 se dieron porque son clásicos y de interés histórico.

El problema de paro de la Máquina de Turing XII

- La conclusión del teorema está en realidad implícita en resultados anteriores, como muestra el siguiente argumento.

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

Demostración:

- Para ver esto, sea L un lenguaje recursivamente enumerable sobre Σ, y sea M una máquina de Turing que acepta L.

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

Demostración:

- Para ver esto, sea L un lenguaje recursivamente enumerable sobre Σ, y sea M una máquina de Turing que acepta L.
- Sea H la máquina de Turing que resuelve el problema de paro.

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

Demostración:

- Para ver esto, sea L un lenguaje recursivamente enumerable sobre Σ, y sea M una máquina de Turing que acepta L.
- Sea H la máquina de Turing que resuelve el problema de paro.
- Construimos a partir de esto el siguiente procedimiento:

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

Demostración:

- Para ver esto, sea L un lenguaje recursivamente enumerable sobre Σ, y sea M una máquina de Turing que acepta L.
- Sea H la máquina de Turing que resuelve el problema de paro.
- Construimos a partir de esto el siguiente procedimiento:
  1. Aplique H a $w_M w$. Si H dice "no", entonces, por definición, w no está en L.

El problema de paro de la Máquina de Turing XIII

Teorema 2

Si el problema de paro fuera decidible, entonces todo lenguaje enumerable recursivamente sería recursivo. En consecuencia, el problema de paro es indecidible.

Demostración:

- Para ver esto, sea L un lenguaje recursivamente enumerable sobre Σ, y sea M una máquina de Turing que acepta L.
- Sea H la máquina de Turing que resuelve el problema de paro.
- Construimos a partir de esto el siguiente procedimiento:
  1. Aplique H a $w_M w$. Si H dice "no", entonces, por definición, w no está en L.
  2. Si H dice "sí", aplique M a w. Pero M debe parar, entonces eventualmente nos dirá que w está en L.

El problema de paro de la Máquina de Turing XIV

- Esto constituye un algoritmo de pertenencia que hace que L sea recursivo.

El problema de paro de la Máquina de Turing XIV

- Esto constituye un algoritmo de pertenencia que hace que L sea recursivo.
- Pero ya sabemos que hay lenguajes enumerables recursivamente que no son recursivos.

El problema de paro de la Máquina de Turing XIV

- Esto constituye un algoritmo de pertenencia que hace que L sea recursivo.
- Pero ya sabemos que hay lenguajes enumerables recursivamente que no son recursivos.
- La contradicción implica que H no puede existir, es decir, que el problema de paro es indecidible.

■

Reducción de un problema indecidible a otro I

- El argumento anterior, que conecta el problema de paro con el problema de membresía, ilustra la técnica de reducción que es muy importante en Teoría de la Computación.

Reducción de un problema indecidible a otro I

- Decimos que un problema A se reduce a un problema B si la decidibilidad de A se deriva de la decidibilidad de B.

Reducción de un problema indecidible a otro I

- Entonces, si sabemos que A es indecidible, podemos concluir que B también es indecidible.

Reducción de un problema indecidible a otro I

- Veamos algunos ejemplos para ilustrar esta idea.

El problema de entrar en un estado I

Ejemplo 1

El **problema de entrar en un estado** es el siguiente. Dada cualquier máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ y cualquier $q \in Q$, $w \in \Sigma^+$, decida si se entra o no en el estado q cuando M se aplica a w. Este problema es indecidible.

El problema de entrar en un estado I

Ejemplo 1

El **problema de entrar en un estado** es el siguiente. Dada cualquier máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ y cualquier $q \in Q$, $w \in \Sigma^+$, decida si se entra o no en el estado q cuando M se aplica a w. Este problema es indecidible.

- Para reducir el problema de paro al problema de entrar en un estado, suponga que tenemos un algoritmo A que resuelve el problema de entrar en un estado.

El problema de entrar en un estado I

Ejemplo 1

El **problema de entrar en un estado** es el siguiente. Dada cualquier máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ y cualquier $q \in Q$, $w \in \Sigma^+$, decida si se entra o no en el estado q cuando M se aplica a w. Este problema es indecidible.

- Entonces podríamos usarlo para resolver el problema de paro.

El problema de entrar en un estado I

Ejemplo 1

El **problema de entrar en un estado** es el siguiente. Dada cualquier máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ y cualquier $q \in Q$, $w \in \Sigma^+$, decida si se entra o no en el estado q cuando M se aplica a w. Este problema es indecidible.

- Por ejemplo, dado cualquier M y w, primero modificamos M para obtener $\widehat{M}$ de tal manera que $\widehat{M}$ se detenga en el estado q si y sólo si M se detiene.

El problema de entrar en un estado II

- Podemos hacer esto simplemente mirando la función de transición δ de M.

El problema de entrar en un estado II

- Si M se detiene, lo hace porque alguna $\delta(q_i, a)$ no está definida.

El problema de entrar en un estado II

- Para obtener $\widehat{M}$, cambiamos cada δ indefinida a

$$\delta(q_i, a) = (q, a, R),$$

donde q es un estado final.

El problema de entrar en un estado II

- Aplicamos el algoritmo de entrar en un estado A a $\left(\widehat{M}, q, w\right)$.

El problema de entrar en un estado II

- Si A responde sí, es decir, se ingresa al estado q, entonces (M, w) se detiene. Si A dice que no, entonces (M, w) no se detiene.

El problema de entrar en un estado II

- Por tanto, la suposición de que el problema de entrar en un estado es decidible nos da un algoritmo para el problema de paro.

El problema de entrar en un estado II

- Debido a que el problema de paro es indecidible, el problema de entrar en un estado también tiene que ser indecidible.

El problema de paro de la cinta en blanco I

Ejemplo 2

El **problema de paro de la cinta en blanco** es otro problema al que se puede reducir al problema de paro. Dada una máquina de Turing M, determine si M se detiene o no si se inicia con una cinta en blanco. Esto es indecidible.

El problema de paro de la cinta en blanco I

Ejemplo 2

El **problema de paro de la cinta en blanco** es otro problema al que se puede reducir al problema de paro. Dada una máquina de Turing M, determine si M se detiene o no si se inicia con una cinta en blanco. Esto es indecidible.

- Para mostrar cómo se logra esta reducción, suponga que se nos da alguna M y alguna w.

El problema de paro de la cinta en blanco I

Ejemplo 2

El **problema de paro de la cinta en blanco** es otro problema al que se puede reducir al problema de paro. Dada una máquina de Turing M, determine si M se detiene o no si se inicia con una cinta en blanco. Esto es indecidible.

- Para mostrar cómo se logra esta reducción, suponga que se nos da alguna M y alguna w.
- Primero construimos a partir de M una nueva máquina M_w que comienza con una cinta en blanco, escribe w en ella y luego se posiciona en una configuración q_0w.

El problema de paro de la cinta en blanco I

Ejemplo 2

El **problema de paro de la cinta en blanco** es otro problema al que se puede reducir al problema de paro. Dada una máquina de Turing M, determine si M se detiene o no si se inicia con una cinta en blanco. Esto es indecidible.

- Para mostrar cómo se logra esta reducción, suponga que se nos da alguna M y alguna w.
- Primero construimos a partir de M una nueva máquina M_w que comienza con una cinta en blanco, escribe w en ella y luego se posiciona en una configuración q_0w.
- Después de eso, M_w actúa como M.

El problema de paro de la cinta en blanco I

Ejemplo 2

El **problema de paro de la cinta en blanco** es otro problema al que se puede reducir al problema de paro. Dada una máquina de Turing M, determine si M se detiene o no si se inicia con una cinta en blanco. Esto es indecidible.

- Para mostrar cómo se logra esta reducción, suponga que se nos da alguna M y alguna w.
- Primero construimos a partir de M una nueva máquina M_w que comienza con una cinta en blanco, escribe w en ella y luego se posiciona en una configuración q_0w.
- Después de eso, M_w actúa como M.
- Claramente, M_w se detendrá con una cinta en blanco si y sólo si M se detiene con w.

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.
- Dado cualquier (M, w), primero construimos M_w, luego le aplicamos el algoritmo del problema de paro de la cinta en blanco.

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.
- Dado cualquier (M, w), primero construimos M_w, luego le aplicamos el algoritmo del problema de paro de la cinta en blanco.
- La conclusión nos dice si M aplicada a w se detendrá.

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.
- Dado cualquier (M, w), primero construimos M_w, luego le aplicamos el algoritmo del problema de paro de la cinta en blanco.
- La conclusión nos dice si M aplicada a w se detendrá.
- Dado que esto se puede hacer para cualquier M y w, un algoritmo para el problema de paro de la cinta en blanco se puede convertir en un algoritmo para el problema de paro.

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.
- Dado cualquier (M, w), primero construimos M_w, luego le aplicamos el algoritmo del problema de paro de la cinta en blanco.
- La conclusión nos dice si M aplicada a w se detendrá.
- Dado que esto se puede hacer para cualquier M y w, un algoritmo para el problema de paro de la cinta en blanco se puede convertir en un algoritmo para el problema de paro.
- Dado que se sabe que este último es indecidible, lo mismo debe ser cierto para el problema de paro de la cinta en blanco.

□

El problema de paro de la cinta en blanco II

- Supongamos ahora que el problema de paro de la cinta en blanco fuera decidible.
- Dado cualquier (M, w), primero construimos M_w, luego le aplicamos el algoritmo del problema de paro de la cinta en blanco.
- La conclusión nos dice si M aplicada a w se detendrá.
- Dado que esto se puede hacer para cualquier M y w, un algoritmo para el problema de paro de la cinta en blanco se puede convertir en un algoritmo para el problema de paro.
- Dado que se sabe que este último es indecidible, lo mismo debe ser cierto para el problema de paro de la cinta en blanco.

□

- La construcción en los argumentos de estos dos ejemplos ilustra un enfoque común para establecer resultados de indecidibilidad. Un diagrama de bloques a menudo nos ayuda a visualizar el proceso.

Reducción de un problema indecidible a otro II

- La construcción del ejemplo 2 se resume en la Figura 3.

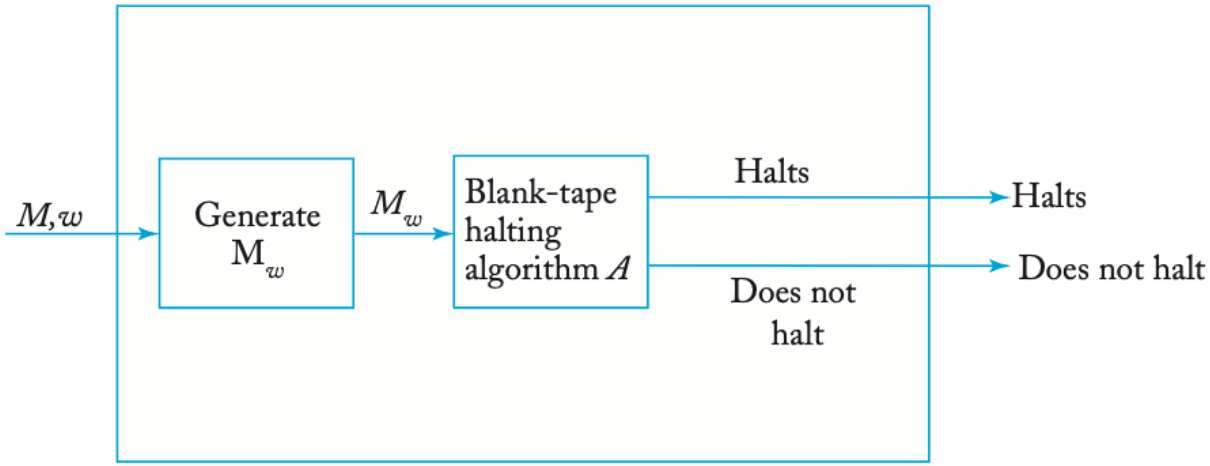

Figure 3: Algoritmo para el problema de paro.

Reducción de un problema indecidible a otro II

- La construcción del ejemplo 2 se resume en la Figura 3.

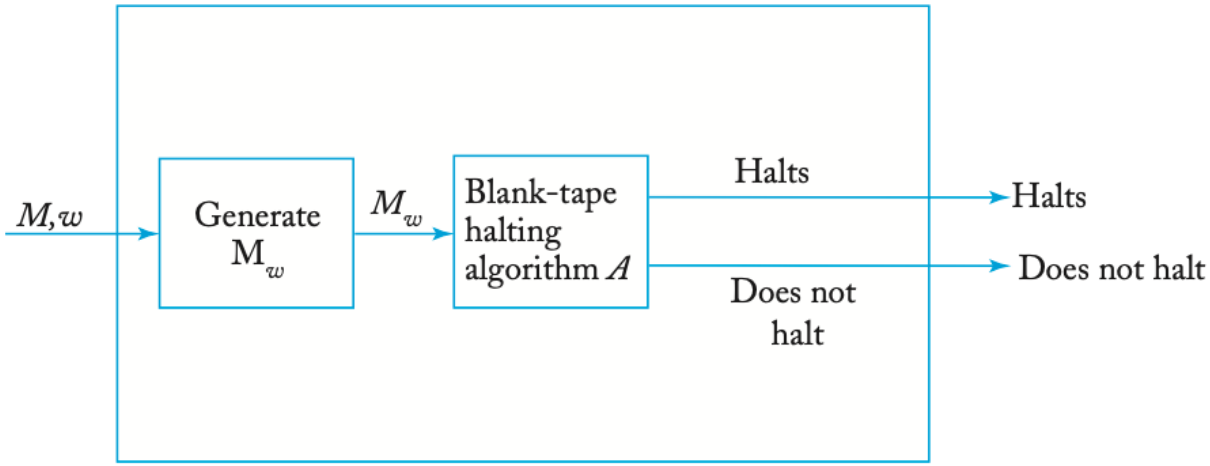

Figure 3: Algoritmo para el problema de paro.

- En ese diagrama, primero usamos un algoritmo que transforma (M, w) en M_w; tal algoritmo existe claramente.

Reducción de un problema indecidible a otro II

- La construcción del ejemplo 2 se resume en la Figura 3.

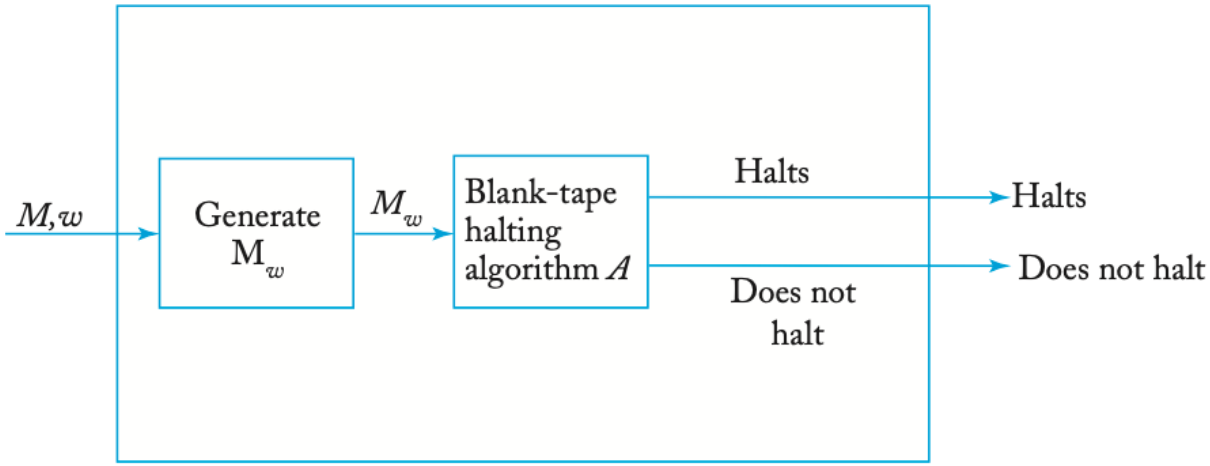

Figure 3: Algoritmo para el problema de paro.

- En ese diagrama, primero usamos un algoritmo que transforma (M, w) en M_w; tal algoritmo existe claramente.
- A continuación, usamos el algoritmo para resolver el problema de paro de la cinta en blanco, que asumimos que existe.

Reducción de un problema indecidible a otro II

- La construcción del ejemplo 2 se resume en la Figura 3.

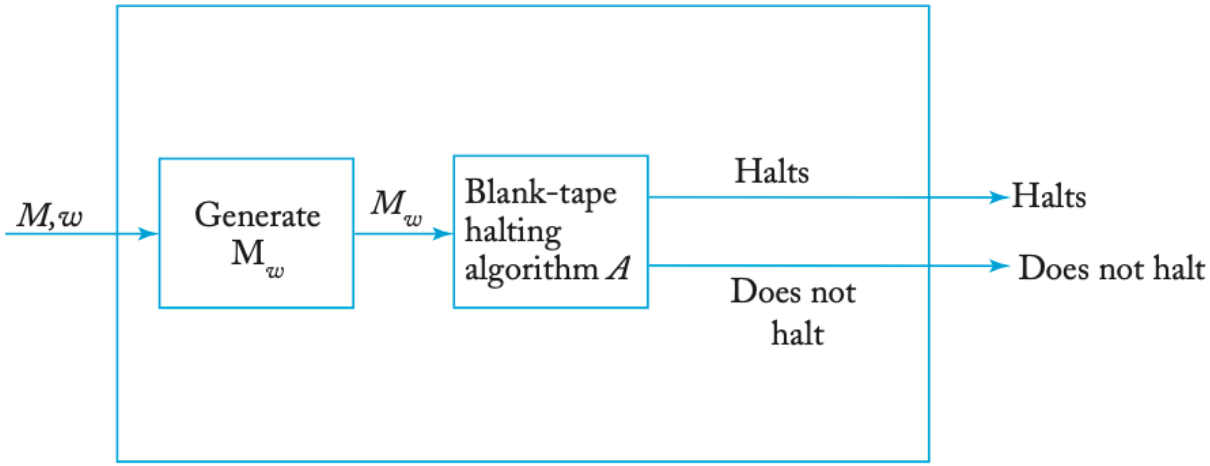

Figure 3: Algoritmo para el problema de paro.

- En ese diagrama, primero usamos un algoritmo que transforma (M, w) en M_w; tal algoritmo existe claramente.
- A continuación, usamos el algoritmo para resolver el problema de paro de la cinta en blanco, que asumimos que existe.
- Poner los dos juntos produce un algoritmo para el problema de paro.

Reducción de un problema indecidible a otro II

- La construcción del ejemplo 2 se resume en la Figura 3.

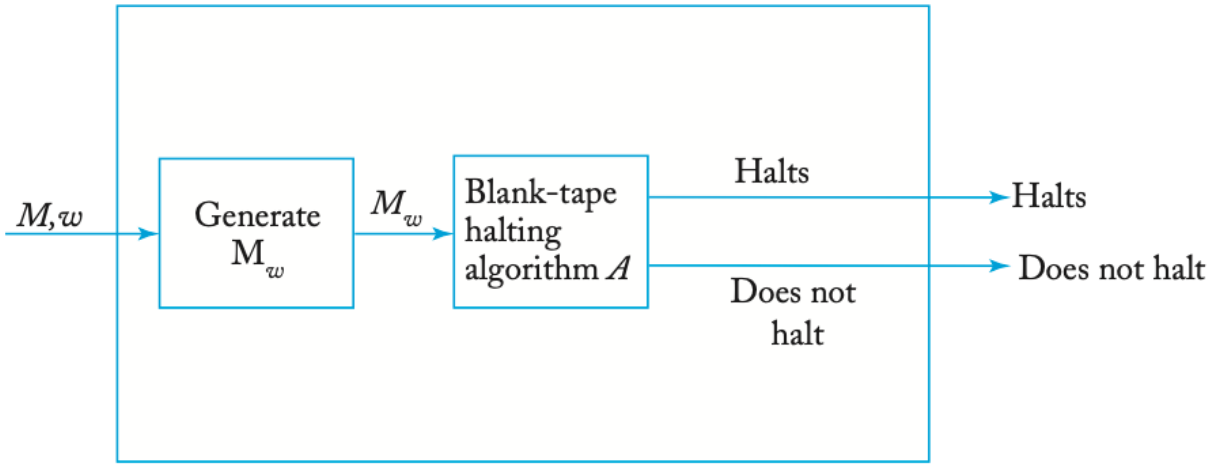

Figure 3: Algoritmo para el problema de paro.

- En ese diagrama, primero usamos un algoritmo que transforma (M, w) en M_w; tal algoritmo existe claramente.
- A continuación, usamos el algoritmo para resolver el problema de paro de la cinta en blanco, que asumimos que existe.
- Poner los dos juntos produce un algoritmo para el problema de paro.
- Pero esto es imposible y podemos concluir que A no puede existir.

Reducción de un problema indecidible a otro III

- Un problema de decisión es efectivamente una función con un rango $\{0, 1\}$, es decir, una respuesta verdadera o falsa.

Reducción de un problema indecidible a otro III

- También podemos mirar funciones más generales para ver si son computables; para ello, seguimos el método establecido y reducimos el problema de paro (o cualquier otro problema conocido como indecidible) al problema de calcular la función en cuestión.

Reducción de un problema indecidible a otro III

- Debido a la tesis de Turing, esperamos que las funciones que se encuentran en circunstancias prácticas sean computables, por lo que para obtener ejemplos de funciones no computables debemos buscar un poco más.

Reducción de un problema indecidible a otro III

- La mayoría de los ejemplos de funciones no computables se asocian con intentos de predecir el comportamiento de las máquinas de Turing.

Reducción de un problema indecidible a otro IV

Ejemplo 3

Sea $\Gamma = \{0, 1, \square\}$. Considere la función $f(n)$ cuyo valor es el número máximo de movimientos que puede realizar cualquier máquina de Turing de n estados que se detiene cuando se inicia con una cinta en blanco. Esta función no es computable.

Reducción de un problema indecidible a otro IV

Ejemplo 3

Sea $\Gamma = \{0, 1, \square\}$. Considere la función $f(n)$ cuyo valor es el número máximo de movimientos que puede realizar cualquier máquina de Turing de n estados que se detiene cuando se inicia con una cinta en blanco. Esta función no es computable.

- Antes de que nos propongamos demostrar esto, asegurémonos de que $f(n)$ esté definida para todo n.

Reducción de un problema indecidible a otro IV

Ejemplo 3

Sea $\Gamma = \{0, 1, \square\}$. Considere la función $f(n)$ cuyo valor es el número máximo de movimientos que puede realizar cualquier máquina de Turing de n estados que se detiene cuando se inicia con una cinta en blanco. Esta función no es computable.

- Observe primero que sólo hay un número finito de máquinas de Turing con n estados.

Reducción de un problema indecidible a otro IV

Ejemplo 3

Sea $\Gamma = \{0, 1, \square\}$. Considere la función $f(n)$ cuyo valor es el número máximo de movimientos que puede realizar cualquier máquina de Turing de n estados que se detiene cuando se inicia con una cinta en blanco. Esta función no es computable.

- Esto se debe a que Q y Γ son finitos, por lo que δ tiene un dominio y un rango finitos.

Reducción de un problema indecidible a otro IV

Ejemplo 3

Sea $\Gamma = \{0, 1, \square\}$. Considere la función $f(n)$ cuyo valor es el número máximo de movimientos que puede realizar cualquier máquina de Turing de n estados que se detiene cuando se inicia con una cinta en blanco. Esta función no es computable.

- Esto, a su vez, implica que sólo hay un número finito de δ diferentes y, por lo tanto, un número finito de máquinas de Turing de n estados diferentes.

Reducción de un problema indecidible a otro V

- De todas las máquinas de n estados, hay algunas que siempre se detienen, por ejemplo, las máquinas que sólo tienen estados finales y, por lo tanto, no hacen ningún movimiento.

Reducción de un problema indecidible a otro V

- Algunas de las máquinas de n estados no se detienen cuando se inician con una cinta en blanco, pero no entran en la definición de f.

Reducción de un problema indecidible a otro V

- Cada máquina que se detiene ejecutará un cierto número de movimientos; de estos, tomamos el más grande para dar $f(n)$.

Reducción de un problema indecidible a otro V

- Tome cualquier máquina de Turing M y número positivo m.

Reducción de un problema indecidible a otro V

- Es fácil modificar M para producir $\widehat{M}$ de tal manera que este último siempre se detendrá con una de dos respuestas: M aplicada a una cinta en blanco se detiene en no más de m movimientos, o M aplicada a una cinta en blanco hace más de m movimientos.

Reducción de un problema indecidible a otro VI

- Todo lo que tenemos que hacer para esto es hacer que M cuente sus movimientos y termine cuando este recuento exceda m.

Reducción de un problema indecidible a otro VI

- Suponga ahora que $f(n)$ es calculable por alguna máquina de Turing F.

Reducción de un problema indecidible a otro VI

- Entonces podemos poner $\widehat{M}$ y F juntos como se muestra en la Figura 4.

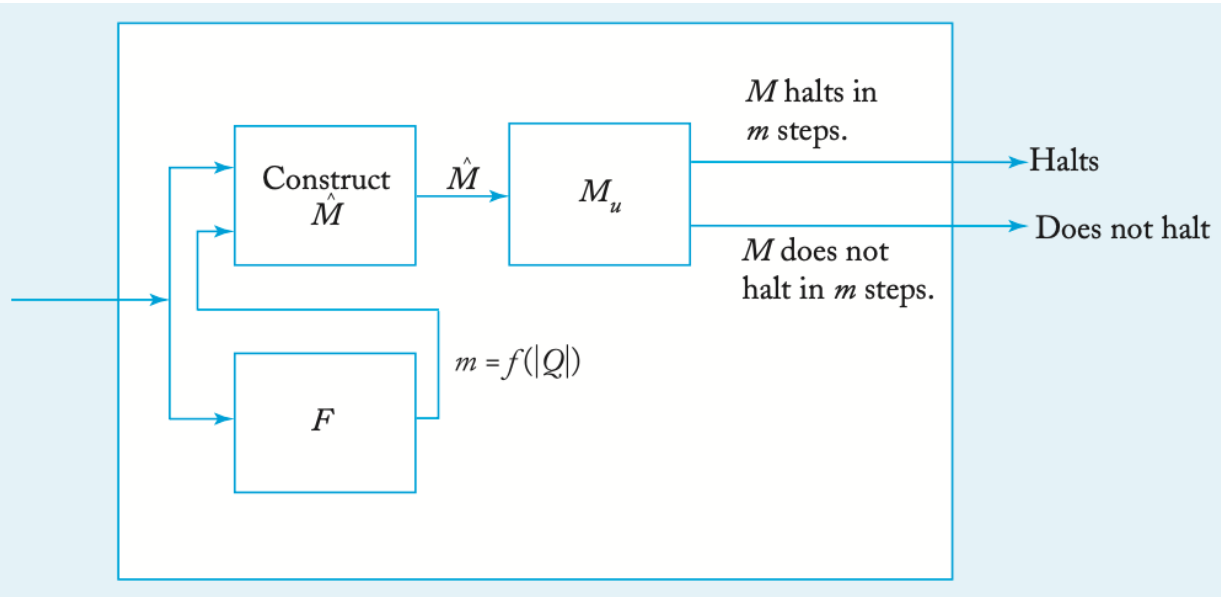

Figure 4: Algoritmo para el problema de paro de la cinta en blanco.

Reducción de un problema indecidible a otro VI

- Primero calculamos $f(|Q|)$, donde Q es el conjunto de estados de M.

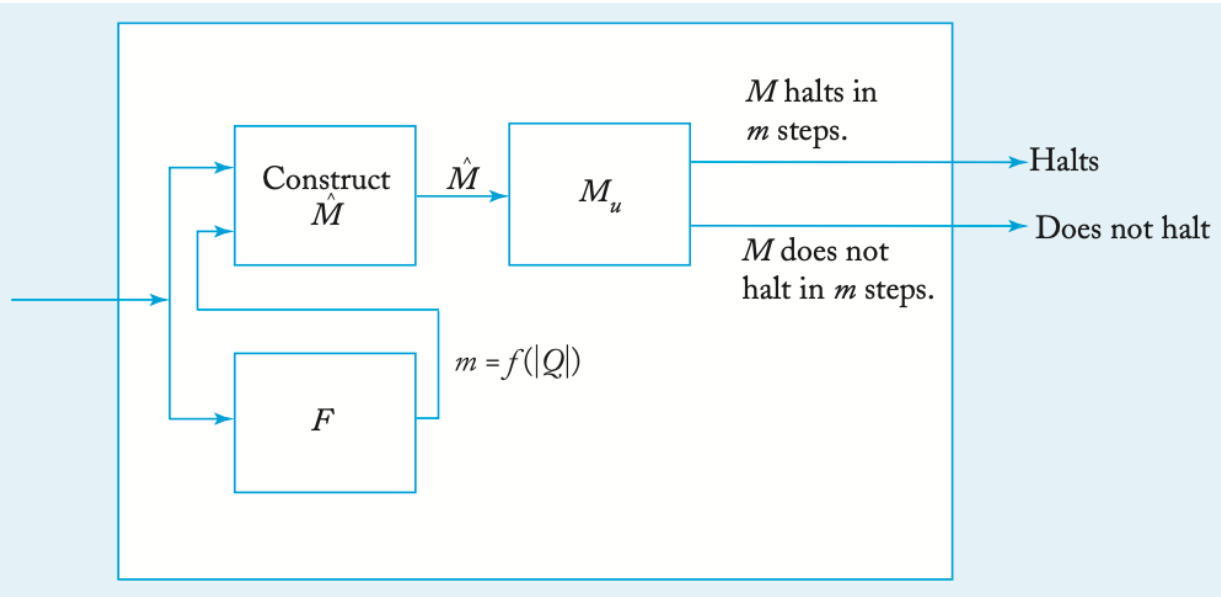

Figure 4: Algoritmo para el problema de paro de la cinta en blanco.

Reducción de un problema indecidible a otro VI

- Esto nos dice el número máximo de movimientos que M puede hacer si se detiene.

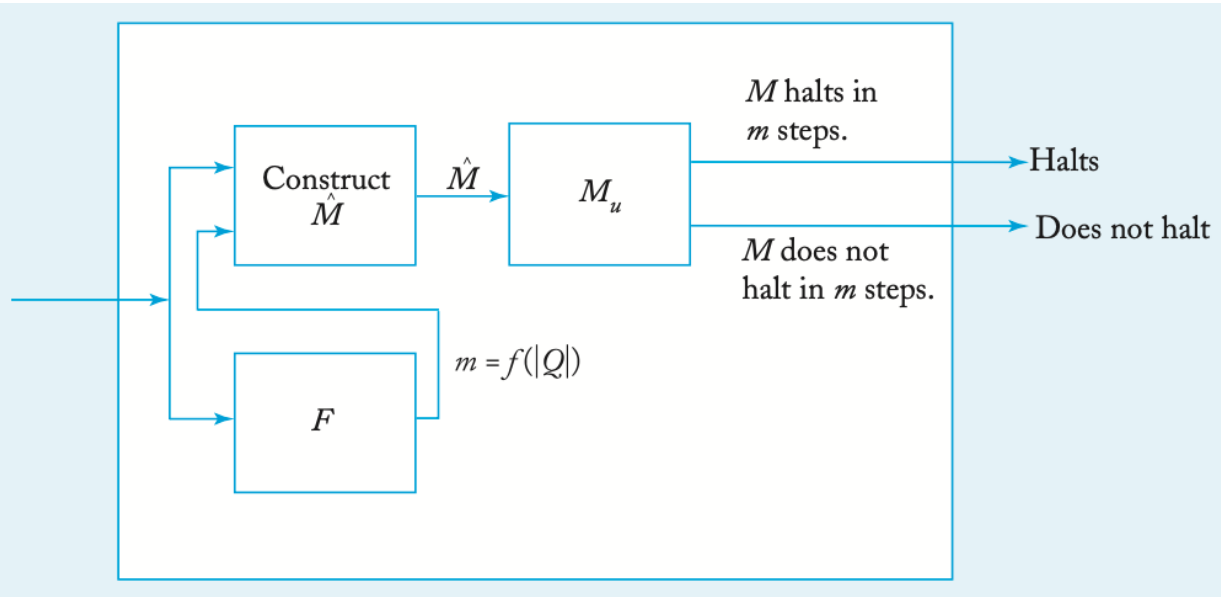

Figure 4: Algoritmo para el problema de paro de la cinta en blanco.

Reducción de un problema indecidible a otro VII

- El valor que obtenemos se usa entonces como m para construir $\widehat{M}$ como se describe, y se da una descripción de $\widehat{M}$ a una máquina de Turing universal para su ejecución.

Reducción de un problema indecidible a otro VII

- Esto nos dice si M aplicada a una cinta en blanco se detiene o no en menos de $f(|Q|)$ pasos.

Reducción de un problema indecidible a otro VII

- Si encontramos que M aplicada a una cinta en blanco hace más de $f(|Q|)$ movimientos, entonces, debido a la definición de f, la implicación es que M nunca se detiene.

Reducción de un problema indecidible a otro VII

- Por tanto, tenemos una solución al problema de paro de la cinta en blanco.

Reducción de un problema indecidible a otro VII

- La imposibilidad de la conclusión nos obliga a aceptar que f no es computable.

□

Ejercicios I

1. Muestre que la pregunta "¿Acepta una máquina de Turing todas las cadenas de longitud par?" es indecidible.
2. Demuestre que el siguiente problema es indecidible. Dada cualquier máquina de Turing M, $a \in \Gamma$ y $w \in \Sigma^+$, determine si el símbolo a se escribe alguna vez cuando M se aplica a w.
3. Demuestre que no existe un algoritmo para decidir si una máquina de Turing arbitraria se detiene o no en todas las entradas.
4. Muestre que no existe un algoritmo para decidir si dos máquinas de Turing M_1 y M_2 aceptan el mismo lenguaje.
5. Muestre que el problema de determinar si una máquina de Turing para con cualquier entrada es indecidible.

Fin del curso

¡Muchas gracias!