Máquinas de Turing No Deterministas y Máquina de Turing Universal

José de Jesús Lavalle Martínez

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

Primavera 2021

Contenido

Máquina de Turing

Definición 1

Una máquina de Turing se define por la septupla

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F),$$

donde

Q es el conjunto de estados internos de la unidad de control,

Σ es el alfabeto de entrada,

Γ es un conjunto finito de símbolos llamado el **alfabeto de la cinta**,

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ es la función de transición,

$q_0 \in Q$ es el estado inicial de la unidad de control,

$\square \in \Gamma$ es un símbolo especial llamado **blanco**,

$F \subseteq Q$ es el conjunto de estados finales.

Se asume que $\Sigma \subseteq \Gamma - \{\square\}$.

Motivación

- Si bien la tesis de Turing hace plausible que la estructura específica de la cinta sea irrelevante para el poder de la Máquina de Turing, no se puede decir lo mismo del no determinismo.

Motivación

- Dado que el no determinismo implica un elemento de elección y, por lo tanto, tiene un sabor no mecánico, apelar a la tesis de Turing es inapropiado.

Motivación

- Debemos analizar el efecto del no determinismo con más detalle si queremos argumentar que el no determinismo no agrega nada al poder de una Máquina de Turing.

Motivación

- Nuevamente recurrimos a la simulación, mostrando que el comportamiento no determinista puede manejarse de manera determinista.

Máquinas de Turing No Deterministas I

Definición 2

Una Máquina de Turing no determinista es un autómata según la Definición 1, excepto que δ es ahora una función

$$\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}.$$

Como siempre, cuando está involucrado el no determinismo, el rango de δ es un conjunto de posibles transiciones, cualquiera de las cuales puede ser elegida por la máquina.

Ejemplo 1

Ejemplo 1

Si una Máquina de Turing tiene transiciones especificadas por

$$\delta(q_0, a) = \{(q_1, b, R), (q_2, c, L)\},$$

es no determinista. Los dos movimientos

$$q_0aaa \vdash bq_1aa$$

y

$$q_0aaa \vdash q_2\square caa$$

son posibles.

□

Máquina de Turing No Determinista II

- Dado que no está claro qué papel juega el no determinismo al calcular funciones, los autómatas no deterministas generalmente se consideran aceptadores.

Máquina de Turing No Determinista II

- Se dice que una Máquina de Turing no determinista acepta w si hay alguna secuencia posible de movimientos tal que

$$q_0 w \overset{*}{\vdash} x_1 q_f x_2,$$

con $q_f \in F$.

Máquina de Turing No Determinista II

- Una máquina no determinista puede tener movimientos disponibles que conducen a un estado no final o a un bucle infinito.

Máquina de Turing No Determinista II

- Pero, como siempre ocurre con el no determinismo, estas alternativas son irrelevantes; todo lo que nos interesa es la existencia de una secuencia de movimientos que conducen a la aceptación.

Máquina de Turing No Determinista III

- Para mostrar que una Máquina de Turing no determinista no es más poderosa que una determinista, necesitamos proporcionar una equivalente determinista para la no determinista.

Máquina de Turing No Determinista III

- El no determinismo puede verse como un algoritmo de retroceso determinista, y una máquina determinista puede simular una no determinista siempre que pueda manejar la contabilidad involucrada en el retroceso.

Máquina de Turing No Determinista III

- Para ver cómo se puede hacer esto simplemente, consideremos una visión alternativa del no determinismo, una que es útil en muchos argumentos: una máquina no determinista puede verse como una que tiene la capacidad de replicarse a sí misma siempre que sea necesario.

Máquina de Turing No Determinista III

- Cuando es posible más de un movimiento, la máquina produce tantas réplicas como sea necesario y asigna a cada réplica la tarea de realizar una de las alternativas.

Máquina de Turing No Determinista IV

- Esta visión del no determinismo puede parecer particularmente no mecanicista, ya que la replicación ilimitada ciertamente no está al alcance de las computadoras actuales.

Máquina de Turing No Determinista IV

- Sin embargo, es posible una simulación.

Máquina de Turing No Determinista IV

- Una forma de visualizar la simulación es usar una Máquina de Turing estándar, manteniendo todas las posibles descripciones instantáneas de la máquina no determinista en su cinta, separadas por alguna convención.

Máquina de Turing No Determinista IV

- La figura 1 muestra la forma en que pueden aparecer las dos configuraciones aq_0aa y bbq_1a.

Máquina de Turing No Determinista IV

- El símbolo $\times$ se utiliza para delimitar el área de interés, mientras que $+$ separa descripciones instantáneas individuales.

Máquina de Turing No Determinista IV

- La máquina de simulación mira todas las configuraciones activas y las actualiza de acuerdo con el programa de la máquina no determinista.

Máquina de Turing No Determinista V

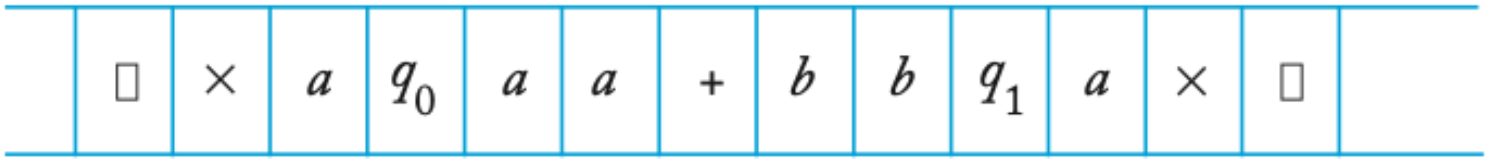

Figure 1: Representación en la cinta de las configuraciones aq_0aa y bbq_1a.

- Las nuevas configuraciones o la expansión de descripciones instantáneas implicarán mover los marcadores $\times$.

Máquina de Turing No Determinista V

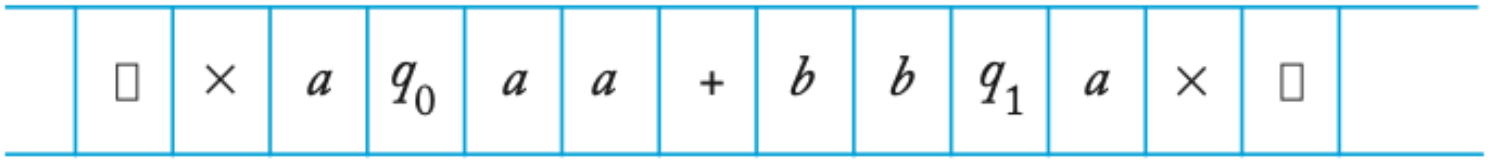

Figure 1: Representación en la cinta de las configuraciones aq_0aa y bbq_1a.

- Los detalles son ciertamente tediosos, pero no difíciles de visualizar.

Máquina de Turing No Determinista V

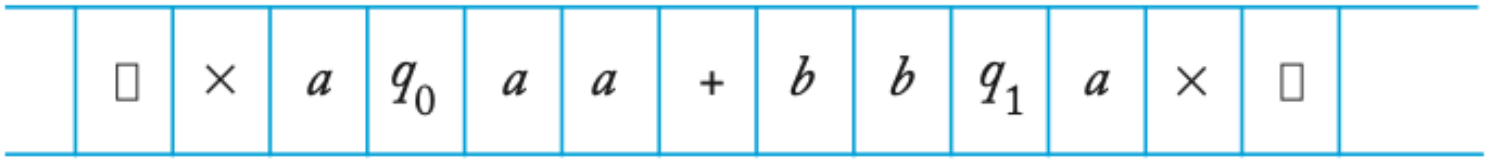

Figure 1: Representación en la cinta de las configuraciones aq_0aa y bbq_1a.

- Con base en esta simulación, llegamos a la conclusión de que para cada Máquina de Turing no determinista existe una máquina estándar determinista equivalente.

Teorema 1

Teorema 1

La clase de máquinas de Turing deterministas y la clase de máquinas de Turing no deterministas son equivalentes.

Teorema 1

La clase de máquinas de Turing deterministas y la clase de máquinas de Turing no deterministas son equivalentes.

Demostración:

- Utilice la construcción sugerida anteriormente para mostrar que cualquier Máquina de Turing no determinista puede ser simulada por una determinista.

Máquina de Turing No Determinista VI

- Más adelante reconsideraremos el efecto del no determinismo en situaciones prácticas, por lo que necesitamos agregar algunos comentarios.

Máquina de Turing No Determinista VI

- Como siempre, el no determinismo puede verse como una elección entre alternativas.

Máquina de Turing No Determinista VI

- Esto se puede visualizar como un árbol de decisiones (Figura 2).

Máquina de Turing No Determinista VII

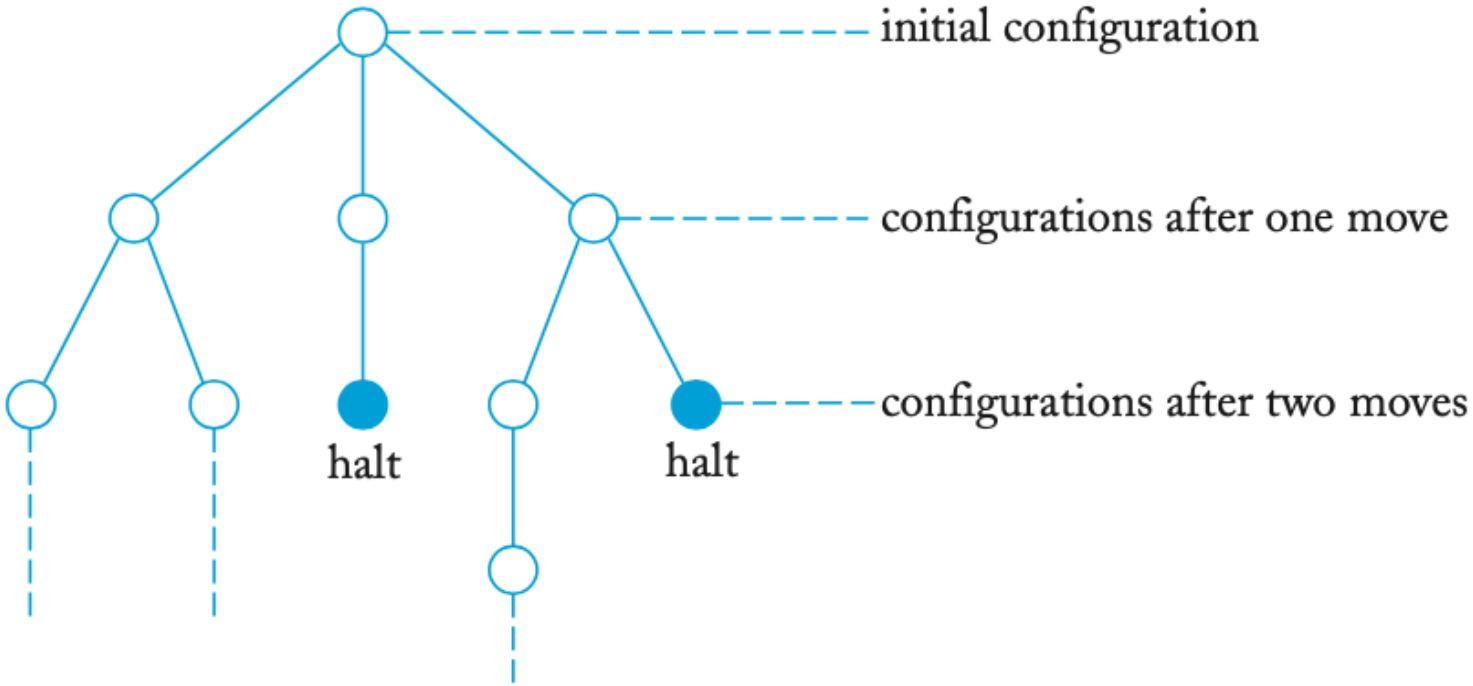

Figure 2: Árbol de decisiones para los movimientos de una Máquina de Turing No Determinista.

Máquina de Turing No Determinista VIII

- El ancho de dicho árbol de configuración depende del factor de ramificación, es decir, la cantidad de opciones disponibles en cada movimiento.

Máquina de Turing No Determinista VIII

- Si k denota la ramificación máxima, entonces

$$M = k^n \tag{1}$$

es el número máximo de configuraciones que pueden existir después de n movimientos.

Máquina de Turing No Determinista VIII

- Para fines posteriores, es necesario profundizar en la definición de aceptación del lenguaje y también incluir el asunto de la membresía.

Máquina de Turing No Determinista IX

Definición 3

Se dice que una Máquina de Turing no determinista M *acepta* un lenguaje L si, para todo $w \in L$, al menos una de las configuraciones posibles acepta w. Puede haber ramificaciones que lleven a configuraciones de no aceptación, mientras que algunas pueden poner la máquina en un bucle infinito. Pero estos son irrelevantes para la aceptación.
Se dice que una Máquina de Turing no determinista M *decide* un lenguaje L si, para todo $w \in \Sigma^*$, hay un camino que conduce a la aceptación o al rechazo.

Ejercicios

1. Escriba programas para máquinas de Turing no deterministas que acepten los siguientes lenguajes. En cada caso, explique si el no determinismo simplifica la tarea y cómo.
   1. $L = \{ww : w \in \{a, b\}^+\}$.
   2. $L = \{ww^R w : w \in \{a, b\}^+\}$.
   3. $L = \{xww^R y : x, y, w \in \{a, b\}^+, |x| \geq |y|\}$.
   4. $L = \{w : n_a(w) = n_b(w) = n_c(w)\}$.

Máquina de Turing Universal I

- Considere el siguiente argumento en contra de la tesis de Turing: "Una Máquina de Turing como se presenta en la Definición 1 es una computadora de propósito especial.

Máquina de Turing Universal I

- Una vez que se define δ, la máquina se limita a realizar un tipo particular de cálculo.

Máquina de Turing Universal I

- Las computadoras digitales, por otro lado, son máquinas de uso general que se pueden programar para realizar diferentes trabajos en diferentes momentos.

Máquina de Turing Universal I

- En consecuencia, las máquinas de Turing no pueden considerarse equivalentes a las computadoras digitales de uso general".

Máquina de Turing Universal II

- Esta objeción se puede superar diseñando una Máquina de Turing reprogramable, llamada **Máquina de Turing Universal**.

Máquina de Turing Universal II

- Una Máquina de Turing Universal M_u es un autómata que, dada como entrada la descripción de cualquier Máquina de Turing M y una cadena w, puede simular el cálculo de M sobre w.

Máquina de Turing Universal II

- Para construir tal M_u, primero elegimos una forma estándar para describir las máquinas de Turing.

Máquina de Turing Universal II

- Podemos, sin perdida de generalidad, suponer que

$$Q = \{q_1, q_2, \ldots, q_n\},$$

con q_1 el estado inicial, q_2 el estado final único y

$$\Gamma = \{a_1, a_2, \ldots, a_m\},$$

donde a_1 representa el espacio en blanco.

Máquina de Turing Universal II

- Luego seleccionamos una codificación en la que q_1 está representado por 1, q_2 está representado por 11, y así sucesivamente.

Máquina de Turing Universal II

- Luego seleccionamos una codificación en la que q_1 está representado por 1, q_2 está representado por 11, y así sucesivamente.
- De manera similar, a_1 se codifica como 1, a_2 como 11, etc. El símbolo 0 se utilizará como separador entre los unos. L se codifica como 1 y R como 11.

Máquina de Turing Universal II

- Luego seleccionamos una codificación en la que q_1 está representado por 1, q_2 está representado por 11, y así sucesivamente.
- De manera similar, a_1 se codifica como 1, a_2 como 11, etc. El símbolo 0 se utilizará como separador entre los unos. L se codifica como 1 y R como 11.
- Con el estado inicial y final y el blanco definidos por esta convención, cualquier Máquina de Turing puede describirse completamente solamente con δ.

Máquina de Turing Universal II

- Luego seleccionamos una codificación en la que q_1 está representado por 1, q_2 está representado por 11, y así sucesivamente.
- De manera similar, a_1 se codifica como 1, a_2 como 11, etc. El símbolo 0 se utilizará como separador entre los unos. L se codifica como 1 y R como 11.
- Con el estado inicial y final y el blanco definidos por esta convención, cualquier Máquina de Turing puede describirse completamente solamente con δ.
- La función de transición se codifica de acuerdo con este esquema, con los argumentos y el resultado en alguna secuencia prescrita.

Máquina de Turing Universal II

- Luego seleccionamos una codificación en la que q_1 está representado por 1, q_2 está representado por 11, y así sucesivamente.
- De manera similar, a_1 se codifica como 1, a_2 como 11, etc. El símbolo 0 se utilizará como separador entre los unos. L se codifica como 1 y R como 11.
- Con el estado inicial y final y el blanco definidos por esta convención, cualquier Máquina de Turing puede describirse completamente solamente con δ.
- La función de transición se codifica de acuerdo con este esquema, con los argumentos y el resultado en alguna secuencia prescrita.
- Por ejemplo, $\delta(q_1, a_2) = (q_2, a_3, L)$ podría aparecer como

$$\cdots 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \cdots$$

Máquina de Turing Universal III

- De esto se deduce que cualquier Máquina de Turing tiene una codificación finita como una cadena en $\{0,1\}^+$ y que, dada cualquier codificación de M, podemos decodificarla de forma única.

Máquina de Turing Universal III

- Algunas cadenas no representarán ninguna Máquina de Turing (por ejemplo, la cadena 00011), pero podemos detectarlas fácilmente, por lo que no son motivo de preocupación.

Máquina de Turing Universal III

- Una Máquina de Turing Universal M_u tiene entonces un alfabeto de entrada que incluye $\{0, 1\}$ y la estructura de una máquina de varias cintas, como se muestra en la Figura 3.

Máquina de Turing Universal IV

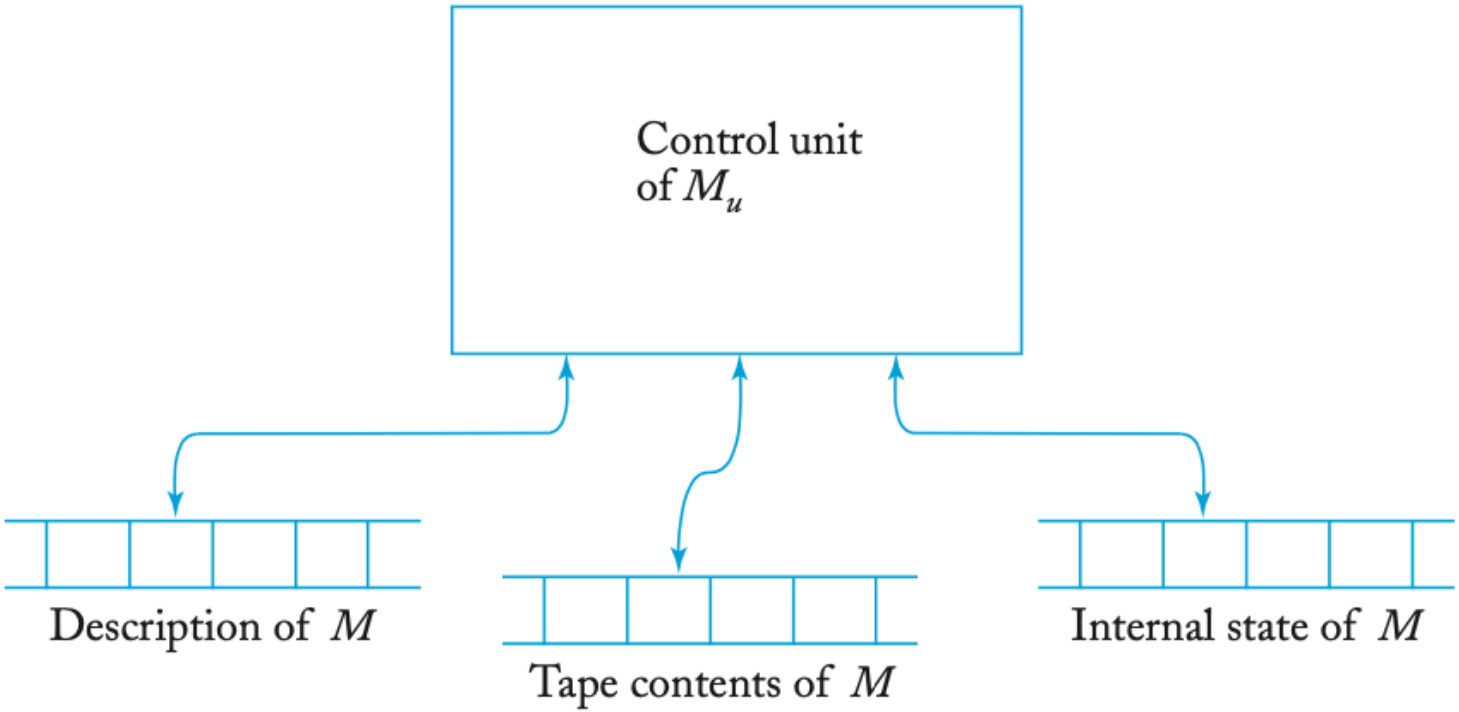

Figure 3: Máquina de Turing Universal M_u con tres cintas.

Máquina de Turing Universal V

- Para cualquier entrada M y w, la cinta 1 mantendrá una definición codificada de M.

Máquina de Turing Universal V

- La cinta 2 contendrá el contenido de la cinta M, y la cinta 3 el estado interno de M.

Máquina de Turing Universal V

- M_u examina primero el contenido de las cintas 2 y 3 para determinar la configuración de M.

Máquina de Turing Universal V

- A continuación, consulta la cinta 1 para ver qué haría M en esta configuración.

Máquina de Turing Universal V

- Finalmente, las cintas 2 y 3 se modificarán para reflejar el resultado del movimiento.

Máquina de Turing Universal V

- Está dentro de lo razonable construir una Máquina de Turing Universal real (ver, por ejemplo, Denning, Dennis y Qualitz 1978), pero el proceso carece de interés.

Máquina de Turing Universal V

- En cambio, preferimos apelar a la hipótesis de Turing.

Máquina de Turing Universal V

- La implementación claramente se puede hacer usando algún lenguaje de programación.

Máquina de Turing Universal VI

- Por lo tanto, esperamos que también se pueda realizar con una máquina Turing estándar.

Máquina de Turing Universal VI

- Entonces, tenemos justificación para afirmar la existencia de una Máquina de Turing que, dado cualquier programa, puede realizar los cálculos especificados por ese programa y que, por lo tanto, es un modelo adecuado para una computadora de propósito general.

Máquina de Turing Universal VI

- La observación de que cada Máquina de Turing puede representarse mediante una cadena de ceros y unos tiene implicaciones importantes.

Máquina de Turing Universal VI

- Pero antes de explorar estas implicaciones, necesitamos revisar algunos resultados de la teoría de conjuntos.

Enumeración I

- Algunos conjuntos son finitos, pero la mayoría de los conjuntos (y lenguajes) interesantes son infinitos.

Enumeración I

- Para conjuntos infinitos, distinguimos entre conjuntos que son contables y conjuntos que son incontables.

Enumeración I

- Se dice que un conjunto es contable si sus elementos se pueden poner en una correspondencia uno a uno con los números enteros positivos.

Enumeración I

- Con esto queremos decir que los elementos del conjunto se pueden escribir en algún orden, digamos, $x_1, x_2, x_3, \ldots$, de modo que cada elemento del conjunto tenga algún índice finito.

Enumeración I

- Por ejemplo, el conjunto de todos los enteros pares se puede escribir en el orden $0, 2, 4, \ldots$.

Enumeración I

- Dado que cualquier entero positivo $2n$ aparece en la posición $n+1$, el conjunto es contable.

Enumeración II

- Esto no debería ser demasiado sorprendente, pero hay ejemplos más complicados, algunos de los cuales pueden parecer contradictorios.

Enumeración II

- Tome el conjunto de todos los cocientes de la forma p/q, donde p y q son números enteros positivos.

Enumeración II

- ¿Cómo deberíamos ordenar este conjunto para demostrar que es contable?

Enumeración II

- No podemos usar la secuencia

$$\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$$

porque entonces el $\frac{2}{3}$ nunca aparecería.

Enumeración II

- Esto no implica que el conjunto sea incontable; en este caso, hay una forma inteligente de ordenar el conjunto para demostrar que, de hecho, es contable.

Enumeración III

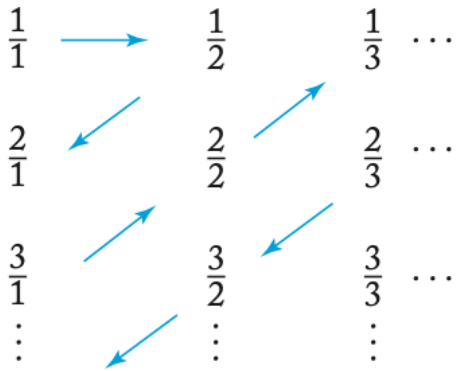

Figure 4: Diagonalización del conjunto de cocientes de enteros positivos.

- Observe el esquema que se muestra en la Figura 4 y escriba el elemento en el orden en que se encuentran siguiendo las flechas.

Enumeración III

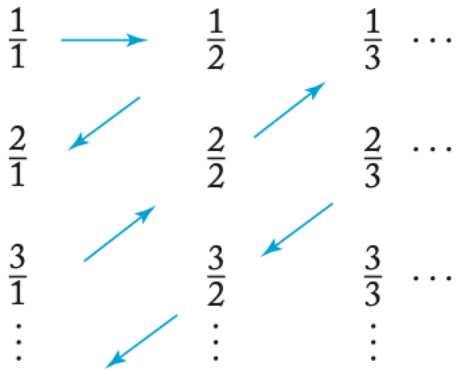

Figure 4: Diagonalización del conjunto de cocientes de enteros positivos.

- Esto nos da

$$\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{3}{1}, \frac{2}{2}, \frac{1}{3}, \frac{2}{3} \ldots$$

Enumeración III

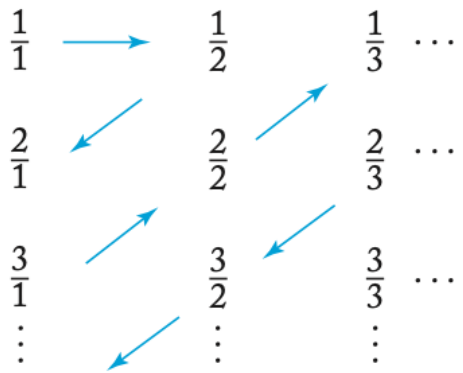

Figure 4: Diagonalización del conjunto de cocientes de enteros positivos.

- Esto nos da

$$\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{3}{1}, \frac{2}{2}, \frac{1}{3}, \frac{2}{3} \ldots$$

- Aquí el elemento $\frac{2}{3}$ aparece en el séptimo lugar y cada elemento tiene alguna posición en la secuencia.

Enumeración III

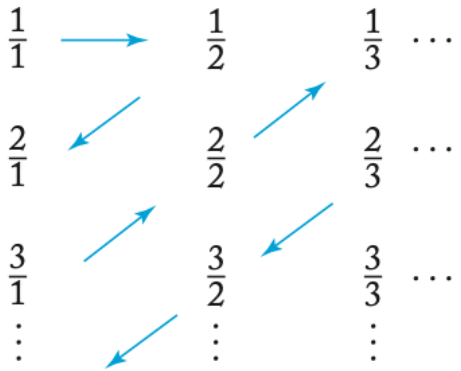

Figure 4: Diagonalización del conjunto de cocientes de enteros positivos.

- Por tanto, el conjunto es contable.

Enumeración IV

- Vemos en este ejemplo que podemos probar que un conjunto es contable si podemos producir un método mediante el cual sus elementos se pueden escribir en alguna secuencia.

Enumeración IV

- A este método lo llamamos **procedimiento de enumeración**.

Enumeración IV

- Dado que un procedimiento de enumeración es una especie de proceso mecánico, podemos usar un modelo de Máquina de Turing para definirlo formalmente.

Enumeración V

Definición 4

Sea S un conjunto de cadenas en algún alfabeto Σ. Entonces, un procedimiento de enumeración para S es una Máquina de Turing que puede realizar la secuencia de pasos

$$q_0\Box \overset{*}{\vdash} q_s x_1 \# s_1 \overset{*}{\vdash} q_s x_2 \# s_2 \overset{*}{\vdash} \cdots,$$

con $x_i \in \Gamma^* - \{\#\}$, $s_i \in S$, de tal manera que cualquier s en S se produce en un número finito de pasos. El estado q_s es un estado que significa pertenencia a S; es decir, siempre que se ingrese q_s, la cadena que sigue a $\#$ debe estar en S.

Enumeración VI

- No todos los conjuntos son contables.

Enumeración VI

- Como veremos en el próximo capítulo, hay algunos conjuntos incontables.

Enumeración VI

- Pero cualquier conjunto para el que existe un procedimiento de enumeración es contable porque la enumeración proporciona la secuencia requerida.

Enumeración VI

- Estrictamente hablando, un procedimiento de enumeración no puede llamarse algoritmo ya que no terminará cuando S sea infinito.

Enumeración VI

- Sin embargo, puede considerarse un proceso significativo, porque produce resultados bien definidos y predecibles.

Ejemplo 2 I

Ejemplo 2

Sea $\Sigma = \{a, b, c\}$. Demuestre que $S = \Sigma^+$ es contable.

Solución:

- Podemos demostrar que $S = \Sigma^+$ es contable si podemos encontrar un procedimiento de enumeración que produzca sus elementos en algún orden, digamos en el orden en que aparecerían en un diccionario.

Ejemplo 2 I

Ejemplo 2

Sea $\Sigma = \{a, b, c\}$. Demuestre que $S = \Sigma^+$ es contable.

Solución:

- Sin embargo, el orden utilizado en los diccionarios no es adecuado sin modificaciones.

Ejemplo 2 I

Ejemplo 2

Sea $\Sigma = \{a, b, c\}$. Demuestre que $S = \Sigma^+$ es contable.

Solución:

- En un diccionario, todas las palabras que comienzan con a se enumeran antes de la cadena b.

Ejemplo 2 I

Ejemplo 2

Sea $\Sigma = \{a, b, c\}$. Demuestre que $S = \Sigma^+$ es contable.

Solución:

- Pero cuando hay un número infinito de palabras a, nunca llegaremos a b, violando así la condición de la Definición 4 de que cualquier cadena dada se enumere después de un número finito de pasos.

Ejemplo 2 II

- En su lugar, podemos usar un orden modificado, en el que tomamos la longitud de la cadena como primer criterio, seguido de un orden alfabético de todas las cadenas de igual longitud.

Ejemplo 2 II

- Este es un procedimiento de enumeración que da la secuencia

$$a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \ldots$$

Ejemplo 2 II

- Como tendremos varios usos para tal orden, lo llamaremos el **orden propio**.

Ejemplo 2 II

- Una consecuencia importante de la discusión anterior es que las máquinas de Turing son contables.

□

Teorema 2

El conjunto de todas las máquinas de Turing, aunque infinito, es contable.

Teorema 2

El conjunto de todas las máquinas de Turing, aunque infinito, es contable.

Demostración:

- Podemos codificar cada Máquina de Turing usando 0 y 1. Con esta codificación, construimos el siguiente procedimiento de enumeración.
  1. Genere la siguiente cadena en $\{0,1\}^+$ en el orden propio.
  2. Verifique la cadena generada para ver si define una Máquina de Turing. Si es así, escríbalo en la cinta en la forma requerida por la Definición 4. Si no es así, ignore la cadena.
  3. Regrese al paso 1.

■

Enumeración VII

- Dado que cada Máquina de Turing tiene una descripción finita, este proceso eventualmente generará cualquier máquina específica.

Enumeración VII

- El orden particular de las máquinas de Turing depende de la codificación que usemos; si usamos una codificación diferente, debemos esperar un orden diferente.

Enumeración VII

- Sin embargo, esto no tiene importancia y muestra que el orden en sí no es importante.

Enumeración VII

- Lo que importa es la existencia de algún orden.

Ejercicios

1. Proporcione la codificación, utilizando el método sugerido, para

$$\delta(q_1, a_1) = (q_1, a_1, R),$$
$$\delta(q_1, a_2) = (q_3, a_1, L),$$
$$\delta(q_3, a_1) = (q_2, a_2, L).$$

2. Si a está codificado como 1, b como 11, R como 1, L como 11, decodifique la cadena 011010111011010.