Una Jerarquía de Lenguajes Formales y Autómatas I

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

- Volvamos ahora nuestra atención a nuestro principal interés, el estudio de los lenguajes formales.

Motivación I

- Nuestro objetivo inmediato será examinar los lenguajes asociados con las máquinas de Turing y algunas de sus restricciones.

Motivación I

- Debido a que las máquinas de Turing pueden realizar cualquier tipo de cálculo algorítmico, esperamos encontrar que la familia de lenguajes asociados con ellas sea bastante amplia.

Motivación I

- Incluye no solo lenguajes regulares y libres de contexto, sino también los diversos ejemplos que hemos encontrado que se encuentran fuera de estas familias.

Motivación II

- La pregunta no trivial es si hay algún lenguaje que no sea aceptado por alguna máquina de Turing.

Motivación II

- Primero responderemos a esta pregunta mostrando que hay más lenguajes que las máquinas de Turing, por lo que debe haber algunos lenguajes para los que no hay máquinas de Turing.

Motivación II

- La demostración es breve y elegante, pero no constructiva, y da poca información sobre el problema.

Motivación II

- Por esta razón, estableceremos la existencia de lenguajes no reconocibles por las máquinas de Turing a través de ejemplos más explícitos que realmente nos permitan identificar uno de esos lenguajes.

Motivación II

- Otra vía de investigación será observar la relación entre las máquinas de Turing y ciertos tipos de gramáticas y establecer una conexión entre estas gramáticas y las gramáticas regulares y libres de contexto.

Motivación III

- Esto conduce a una jerarquía de gramáticas y, a través de ella, a un método para clasificar familias de lenguajes.

Motivación III

- Algunos diagramas de teoría de conjuntos ilustran claramente las relaciones entre varias familias de lenguas.

Motivación III

- Estrictamente hablando, muchos de los argumentos de este capítulo son válidos solo para lenguajes que no incluyen la cadena vacía.

Motivación III

- Esta restricción surge del hecho de que las máquinas de Turing, como las hemos definido, no pueden aceptar la cadena vacía.

Motivación III

- Para evitar tener que reformular la definición o tener que agregar un descargo de responsabilidad repetido, asumimos tácitamente que los lenguajes discutidos en este capítulo, a menos que se indique lo contrario, no contienen λ.

Lenguajes recursivos y recursivamente enumerables I

- Comenzamos con algo de terminología para los lenguajes asociados con las máquinas de Turing.

- Al hacerlo, debemos hacer la importante distinción entre lenguajes para los que existe una máquina de Turing de aceptación y lenguajes para los que existe un algoritmo de pertenencia.

- Debido a que una máquina de Turing no se detiene necesariamente en una entrada que no acepta, la primera no implica la segunda.

Definición 1

Definición 1

Se dice que un lenguaje L es recursivamente enumerable si existe una máquina de Turing que lo acepta.

Definición 1

Definición 1

Se dice que un lenguaje L es recursivamente enumerable si existe una máquina de Turing que lo acepta.

- Esta definición implica sólo que existe una máquina de Turing M, tal que, para cada $w \in L$,

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

con q_f un estado final.

Definición 1

Se dice que un lenguaje L es recursivamente enumerable si existe una máquina de Turing que lo acepta.

- La definición no dice nada sobre lo que sucede para $w \notin L$; puede ser que la máquina se detenga en un estado no final o que nunca se detenga y entre en un bucle infinito.

Definición 2

- Podemos ser más exigentes y pedir que la máquina nos diga si una entrada determinada está en su lenguaje o no.

Definición 2

Definición 2

Se dice que un lenguaje L sobre Σ es recursivo si existe una máquina de Turing M que acepta L y que se detiene sobre cada w en Σ^+. En otras palabras, un lenguaje es recursivo si y sólo si existe un algoritmo de pertenencia para él.

Lenguajes recursivos y recursivamente enumerables II

- Si un lenguaje es recursivo, existe un procedimiento de enumeración fácil de construir.

Lenguajes recursivos y recursivamente enumerables II

- Supongamos que M es una máquina de Turing que determina la pertenencia a un lenguaje recursivo L.

- Primero construimos otra máquina de Turing, digamos $\widehat{M}$, que genera todas las cadenas en Σ^+ en el orden correcto, digamos $w_1, w_2, \ldots$.

Lenguajes recursivos y recursivamente enumerables II

- Como estas cadenas se generan, se convierten en la entrada a M, que se modifica para que escriba cadenas en su cinta sólo si están en L.

Lenguajes recursivos y recursivamente enumerables II

- No es tan fácil de ver que existe también un procedimiento de enumeración para cada lenguaje recursivamente enumerable.

- No podemos usar el argumento anterior tal como está, porque si alguna w_j no está en L, la máquina M, cuando se inicia con w_j en su cinta, puede que nunca se detenga y, por lo tanto, nunca llegue a las cadenas de L que siguen a w_j en la enumeración.

Lenguajes recursivos y recursivamente enumerables III

- Para asegurarse de que esto no suceda, el cálculo se realiza de una manera diferente.

Lenguajes recursivos y recursivamente enumerables III

- Primero obtenemos $\widehat{M}$ para generar w_1 y dejamos que M ejecute un movimiento en él.

Lenguajes recursivos y recursivamente enumerables III

- Luego dejamos que $\widehat{M}$ genere w_2 y dejamos que M ejecute un movimiento en w_2, seguido del segundo movimiento en w_1.

Lenguajes recursivos y recursivamente enumerables III

- Después de esto, generamos w_3 y damos un paso en w_3, el segundo paso en w_2, el tercer paso en w_1, y así sucesivamente.

- El orden de ejecución se muestra en la Figura 1.

Lenguajes recursivos y recursivamente enumerables III

- A partir de esto, está claro que M nunca entrará en un bucle infinito.

- Dado que cualquier $w \in L$ es generado por $\widehat{M}$ y aceptado por M en un número finito de pasos, cada cadena en L es finalmente producida por M.

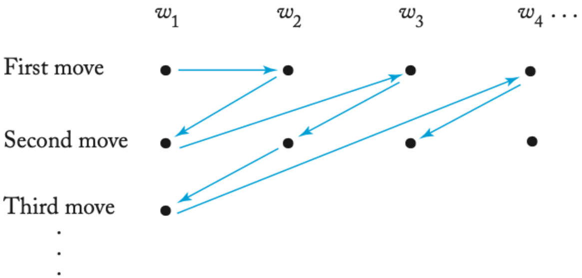

Figure 1: Orden de ejecución para enumerar un lenguaje recursivamente enumerable.

Lenguajes recursivos y recursivamente enumerables V

- Es fácil ver que todos los lenguajes para los que existe un procedimiento de enumeración son enumerables de forma recursiva.

- Simplemente comparamos la cadena de entrada dada con cadenas sucesivas generadas por el procedimiento de enumeración.

- Si $w \in L$, eventualmente obtendremos una coincidencia y el proceso se puede terminar.

Lenguajes recursivos y recursivamente enumerables V

- Las definiciones 1 y 2 nos dan muy poca información sobre la naturaleza de los lenguajes recursivos o recursivamente enumerables.

- Estas definiciones adjuntan nombres a las familias de lenguajes asociadas con las máquinas de Turing, pero no arrojan luz sobre la naturaleza de los lenguajes representativos en estas familias.

- Tampoco nos dicen mucho sobre las relaciones entre estos lenguajes o su conexión con las familias de lenguajes que hemos encontrado antes.

- Por lo tanto, nos enfrentamos de inmediato a preguntas como

- "¿Hay lenguajes que son recursivamente enumerables pero no recursivos?" y

Lenguajes recursivos y recursivamente enumerables VI

- "¿Hay lenguajes, que se puedan describir de alguna manera, que no sean enumerables de forma recursiva?"

Lenguajes recursivos y recursivamente enumerables VI

- Si bien podremos proporcionar algunas respuestas, no podremos producir ejemplos muy explícitos para ilustrar estas preguntas, especialmente la segunda.

- Podemos establecer la existencia de lenguajes que no son enumerables de forma recursiva de diversas formas.

- Una es muy corta y utiliza un resultado muy fundamental y elegante de las matemáticas.

Lenguajes que no son recursivamente enumerables I

Teorema 1

Sea S un conjunto numerable infinito. Entonces su conjunto potencia 2^S no es contable.

Lenguajes que no son recursivamente enumerables I

Teorema 1

Sea S un conjunto numerable infinito. Entonces su conjunto potencia 2^S no es contable.

Demostración:

- Sea $S = \{s_1, s_2, s_3, \ldots\}$. Entonces, cualquier elemento t de 2^S puede representarse mediante una secuencia de ceros y unos, con un 1 en la posición i si y sólo si s_i está en t.

Teorema 1

Sea S un conjunto numerable infinito. Entonces su conjunto potencia 2^S no es contable.

Demostración:

- Sea $S = \{s_1, s_2, s_3, \ldots\}$. Entonces, cualquier elemento t de 2^S puede representarse mediante una secuencia de ceros y unos, con un 1 en la posición i si y sólo si s_i está en t.
- Por ejemplo, el conjunto $\{s_2, s_3, s_6\}$ está representado por 011001, mientras que $\{s_1, s_3, s_5, \ldots\}$ está representado por $10101\ldots$.

Lenguajes que no son recursivamente enumerables II

- Claramente, cualquier elemento de 2^S puede ser representado por tal secuencia, y cualquier secuencia representa un elemento único de 2^S.

Lenguajes que no son recursivamente enumerables II

- Supongamos que 2^S fuera contable; entonces sus elementos podrían escribirse en algún orden, digamos $t_1, t_2, \ldots$, y podríamos ingresarlos en una tabla, como se muestra en la Figura 2.

Lenguajes que no son recursivamente enumerables II

- Supongamos que 2^S fuera contable; entonces sus elementos podrían escribirse en algún orden, digamos $t_1, t_2, \ldots$, y podríamos ingresarlos en una tabla, como se muestra en la Figura 2.

t_1	1	0	0	0	0	...
t_2	1	1	0	0	0	...
t_3	1	1	0	1	0	...
t_4	1	1	0	0	1	...
$\vdots$					$\ddots$	

Figure 2: Diagonalización de 2^S.

Lenguajes que no son recursivamente enumerables III

- En esta tabla, tome los elementos de la diagonal principal y complemente cada entrada, es decir, reemplace 0 con 1 y viceversa.

- En el ejemplo de la Figura 11.2, los elementos son $1100\ldots$, por lo que obtenemos $0011\ldots$ como resultado.

Lenguajes que no son recursivamente enumerables III

- La nueva secuencia a lo largo de la diagonal representa algún elemento de 2^S, digamos t_i para alguna i.

Lenguajes que no son recursivamente enumerables III

- Pero no puede ser t_1 porque difiere de t_1 en s_1.

Lenguajes que no son recursivamente enumerables III

- Por la misma razón, no puede ser t_2, t_3 ni ninguna otra entrada en la enumeración.

- Esta contradicción crea un callejón sin salida lógico que sólo puede eliminarse descartando la suposición de que 2^S es contable.

$\blacksquare$

- Este tipo de argumento, debido a que implica una manipulación de los elementos diagonales de una tabla, se llama *diagonalización*.

Lenguajes que no son recursivamente enumerables IV

- La técnica se atribuye al matemático G. F. Cantor, quien la utilizó para demostrar que el conjunto de números reales no es contable.

- En los próximos capítulos, veremos un argumento similar en varios contextos.

Lenguajes que no son recursivamente enumerables IV

- El Teorema 1 es la diagonalización en su forma más pura.

Lenguajes que no son recursivamente enumerables IV

- Como consecuencia inmediata de este resultado, podemos mostrar que, en cierto sentido, hay menos máquinas de Turing que lenguajes, por lo que debe haber algunos lenguajes que no sean recursivamente enumerables.

Lenguajes que no son recursivamente enumerables V

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Lenguajes que no son recursivamente enumerables V

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Demostración:

- Un lenguaje es un subconjunto de Σ^*, y cada uno de esos subconjuntos es un lenguaje.

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Demostración:

- Por lo tanto, el conjunto de todos los lenguajes es exactamente 2^{Σ^*}.

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Demostración:

- Dado que Σ^* es infinito, el Teorema 1 nos dice que el conjunto de todos los lenguajes sobre Σ no es contable.

Lenguajes que no son recursivamente enumerables V

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Demostración:

- Pero el conjunto de todas las máquinas de Turing se puede enumerar, por lo que el conjunto de todos los lenguajes enumerables de forma recursiva es contable.

Lenguajes que no son recursivamente enumerables V

Teorema 2

Para cualquier Σ no vacío, existen lenguajes que no son enumerables de forma recursiva.

Demostración:

- Esto implica que debe haber algunos lenguajes sobre Σ que no se puedan enumerar de forma recursiva. $\blacksquare$

Lenguajes que no son recursivamente enumerables VI

- Esta demostración, aunque breve y sencilla, es insatisfactoria en muchos sentidos.

Lenguajes que no son recursivamente enumerables VI

- Es completamente no constructiva y, si bien nos informa de la existencia de algunos lenguajes que no son enumerables de forma recursiva, no nos da ningún indicio sobre cómo podrían ser estos lenguajes.

Lenguajes que no son recursivamente enumerables VI

- En el siguiente conjunto de resultados, investigamos la conclusión de manera más explícita.

Un lenguaje que no es recursivamente enumerable I

- Dado que todo lenguaje que puede describirse de forma algorítmica directa puede ser aceptado por una máquina de Turing y, por tanto, es recursivamente enumerable, la descripción de un lenguaje que no sea recursivamente enumerable debe ser indirecta.

Un lenguaje que no es recursivamente enumerable I

- Sin embargo, es posible.

Un lenguaje que no es recursivamente enumerable I

- El argumento implica una variación del tema de la diagonalización.

Un lenguaje que no es recursivamente enumerable I

Teorema 3

Existe un lenguaje recursivamente enumerable cuyo complemento no es recursivamente enumerable.

Un lenguaje que no es recursivamente enumerable I

Teorema 3

Existe un lenguaje recursivamente enumerable cuyo complemento no es recursivamente enumerable.

Demostración:

- Sea $\Sigma = \{a\}$, y considere el conjunto de todas las máquinas de Turing con este alfabeto de entrada.

Un lenguaje que no es recursivamente enumerable I

Teorema 3

Existe un lenguaje recursivamente enumerable cuyo complemento no es recursivamente enumerable.

Demostración:

- Según el Teorema 10.3, este conjunto es contable, por lo que podemos asociar un orden $M_1, M_2, \ldots$ con sus elementos.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.
- Reciprocamente, para cada lenguaje enumerable recursivamente sobre Σ, hay alguna máquina de Turing que lo acepta.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.
- Reciprocamente, para cada lenguaje enumerable recursivamente sobre Σ, hay alguna máquina de Turing que lo acepta.
- Ahora consideramos un nuevo lenguaje L definido como sigue.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.
- Reciprocamente, para cada lenguaje enumerable recursivamente sobre Σ, hay alguna máquina de Turing que lo acepta.
- Ahora consideramos un nuevo lenguaje L definido como sigue.
- Para cada $i \geq 1$, la cadena a^i está en L si y sólo si $a^i \in L(M_i)$.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.
- Reciprocamente, para cada lenguaje enumerable recursivamente sobre Σ, hay alguna máquina de Turing que lo acepta.
- Ahora consideramos un nuevo lenguaje L definido como sigue.
- Para cada $i \geq 1$, la cadena a^i está en L si y sólo si $a^i \in L(M_i)$.
- Está claro que el lenguaje L está bien definido, ya que el enunciado $a^i \in L(M_i)$, y por lo tanto $a^i \in L$, debe ser verdadero o falso.

Un lenguaje que no es recursivamente enumerable II

- Para cada máquina de Turing M_i, hay un lenguaje asociado $L(M_i)$ enumerable recursivamente.
- Reciprocamente, para cada lenguaje enumerable recursivamente sobre Σ, hay alguna máquina de Turing que lo acepta.
- Ahora consideramos un nuevo lenguaje L definido como sigue.
- Para cada $i \geq 1$, la cadena a^i está en L si y sólo si $a^i \in L(M_i)$.
- Está claro que el lenguaje L está bien definido, ya que el enunciado $a^i \in L(M_i)$, y por lo tanto $a^i \in L$, debe ser verdadero o falso.
- A continuación, consideramos el complemento de L,

$$\overline{L} = \{a^i : a^i \notin L(M_i)\}, \tag{1}$$

que también está bien definido pero, como mostraremos, no es recursivamente enumerable.

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.
- Si es así, entonces debe haber alguna máquina de Turing, digamos M_k, tal que

$$\overline{L} = L(M_k). \tag{2}$$

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.
- Si es así, entonces debe haber alguna máquina de Turing, digamos M_k, tal que
$$\overline{L} = L(M_k). \tag{2}$$
- Considere la cadena a^k. ¿Está en L o en $\overline{L}$?

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.
- Si es así, entonces debe haber alguna máquina de Turing, digamos M_k, tal que

$$\overline{L} = L(M_k). \tag{2}$$

- Considere la cadena a^k. ¿Está en L o en $\overline{L}$?
- Suponga que $a^k \in \overline{L}$.

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.
- Si es así, entonces debe haber alguna máquina de Turing, digamos M_k, tal que

$$\overline{L} = L(M_k). \tag{2}$$

- Considere la cadena a^k. ¿Está en L o en $\overline{L}$?
- Suponga que $a^k \in \overline{L}$.
- Por (2) esto implica que

$$a^k \in L(M_k).$$

Un lenguaje que no es recursivamente enumerable III

- Mostraremos esto por contradicción, partiendo del supuesto de que $\overline{L}$ es recursivamente enumerable.
- Si es así, entonces debe haber alguna máquina de Turing, digamos M_k, tal que
$$\overline{L} = L(M_k). \tag{2}$$
- Considere la cadena a^k. ¿Está en L o en $\overline{L}$?
- Suponga que $a^k \in \overline{L}$.
- Por (2) esto implica que
$$a^k \in L(M_k).$$
- Pero (1) ahora implica que
$$a^k \notin \overline{L}.$$

Un lenguaje que no es recursivamente enumerable IV

- Alternativamente, si asumimos que a^k está en L, entonces $a^k \notin \overline{L}$ y (2) implica que
$$a^k \notin L(M_k).$$

Un lenguaje que no es recursivamente enumerable IV

- Alternativamente, si asumimos que a^k está en L, entonces $a^k \notin \overline{L}$ y (2) implica que
$$a^k \notin L(M_k).$$
- Pero luego de (1) obtenemos que
$$a^k \in \overline{L}.$$

Un lenguaje que no es recursivamente enumerable IV

- Alternativamente, si asumimos que a^k está en L, entonces $a^k \notin \overline{L}$ y (2) implica que
$$a^k \notin L(M_k).$$
- Pero luego de (1) obtenemos que
$$a^k \in \overline{L}.$$
- La contradicción es ineludible y debemos concluir que nuestra suposición de que L es recursivamente enumerable es falsa.

Un lenguaje que no es recursivamente enumerable IV

- Alternativamente, si asumimos que a^k está en L, entonces $a^k \notin \overline{L}$ y (2) implica que
$$a^k \notin L(M_k).$$
- Pero luego de (1) obtenemos que
$$a^k \in \overline{L}.$$
- La contradicción es ineludible y debemos concluir que nuestra suposición de que L es recursivamente enumerable es falsa.
- Para completar la demostración del teorema como se indica, aún debemos demostrar que L es recursivamente enumerable.

Un lenguaje que no es recursivamente enumerable IV

- Alternativamente, si asumimos que a^k está en L, entonces $a^k \notin \overline{L}$ y (2) implica que
$$a^k \notin L(M_k).$$
- Pero luego de (1) obtenemos que
$$a^k \in \overline{L}.$$
- La contradicción es ineludible y debemos concluir que nuestra suposición de que L es recursivamente enumerable es falsa.
- Para completar la demostración del teorema como se indica, aún debemos demostrar que L es recursivamente enumerable.
- Para ello, podemos utilizar el procedimiento de enumeración conocido para las máquinas de Turing.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.
- Luego usamos el procedimiento de enumeración de las máquinas de Turing para encontrar M_i.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.
- Luego usamos el procedimiento de enumeración de las máquinas de Turing para encontrar M_i.
- Finalmente, damos su descripción junto con a^i a una máquina universal de Turing M_u que simula la acción de M sobre a^i.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.
- Luego usamos el procedimiento de enumeración de las máquinas de Turing para encontrar M_i.
- Finalmente, damos su descripción junto con a^i a una máquina universal de Turing M_u que simula la acción de M sobre a^i.
- Si a^i está en L, el cálculo realizado por M_u finalmente se detendrá.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.
- Luego usamos el procedimiento de enumeración de las máquinas de Turing para encontrar M_i.
- Finalmente, damos su descripción junto con a^i a una máquina universal de Turing M_u que simula la acción de M sobre a^i.
- Si a^i está en L, el cálculo realizado por M_u finalmente se detendrá.
- El efecto combinado de esto es una máquina de Turing que acepta cada $a^i \in L$.

Un lenguaje que no es recursivamente enumerable V

- Dada a^i, primero encontramos i contando el número de símbolos a.
- Luego usamos el procedimiento de enumeración de las máquinas de Turing para encontrar M_i.
- Finalmente, damos su descripción junto con a^i a una máquina universal de Turing M_u que simula la acción de M sobre a^i.
- Si a^i está en L, el cálculo realizado por M_u finalmente se detendrá.
- El efecto combinado de esto es una máquina de Turing que acepta cada $a^i \in L$.
- Por lo tanto, según la definición 1, L es recursivamente enumerable.

$\blacksquare$

- La demostración de este teorema muestra explícitamente, a través de (1), un lenguaje bien definido que no es recursivamente enumerable.

- Esto no quiere decir que haya una interpretación fácil e intuitiva de $\overline{L}$.

Un lenguaje que no es recursivamente enumerable V

- Sería difícil exhibir más que unos pocos miembros triviales de este lenguaje.

Un lenguaje que no es recursivamente enumerable V

- Sin embargo está correctamente definido.

Un lenguaje que es recursivamente enumerable pero no es recursivo I

- A continuación, mostramos que hay algunos lenguajes que se pueden enumerar de forma recursiva pero que no son recursivos.

Un lenguaje que es recursivamente enumerable pero no es recursivo I

- Una vez más, debemos hacerlo de una manera bastante indirecta.

Un lenguaje que es recursivamente enumerable pero no es recursivo I

- Comenzamos por establecer un resultado subsidiario.

Un lenguaje que es recursivamente enumerable pero no es recursivo I

Teorema 4

Si un lenguaje L y su complemento $\overline{L}$ son recursivamente enumerables, ambos lenguajes son recursivos. Si L es recursivo, entonces $\overline{L}$ también es recursivo y, en consecuencia, ambos son recursivamente enumerables.

Un lenguaje que es recursivamente enumerable pero no es recursivo I

Demostración:

- Si L y $\overline{L}$ son recursivamente enumerables, entonces existen máquinas de Turing M y $\widehat{M}$ que sirven como procedimientos de enumeración para L y $\overline{L}$, respectivamente.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- El primero producirá $w_1, w_2, \ldots$ en L, el segundo $\widehat{w}_1, \widehat{w}_2, \ldots$ en $\overline{L}$.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Supongamos que ahora se nos da cualquier $w \in \Sigma^+$.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Primero dejamos que M genere w_1 y lo comparamos con w.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Si no son iguales, dejamos que $\widehat{M}$ genere $\widehat{w}_1$ y se vuelve a comparar.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Si necesitamos continuar, luego dejamos que M genere w_2, luego $\widehat{M}$ genera $\widehat{w}_2$, y así sucesivamente.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Cualquier $w \in \Sigma^+$ será generado por M o $\widehat{M}$, por lo que eventualmente obtendremos una coincidencia.

Un lenguaje que es recursivamente enumerable pero no es recursivo II

- Si la cadena coincidente es producida por M, w pertenece a L, de lo contrario está en $\overline{L}$.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- El proceso es un algoritmo de pertenencia tanto para L como para $\overline{L}$, por lo que ambos son recursivos.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- Recíprocamente, suponga que L es recursivo.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- Entonces existe un algoritmo de membresía para él.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- Pero esto se convierte en un algoritmo de pertenencia para $\overline{L}$ simplemente complementando su conclusión.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- Por tanto, $\overline{L}$ es recursivo.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- Dado que cualquier lenguaje recursivo es recursivamente enumerable, la prueba está completa

■

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- De esto, concluimos directamente que la familia de lenguajes recursivamente enumerables y la familia de lenguajes recursivos no son idénticas.

Un lenguaje que es recursivamente enumerable pero no es recursivo III

- El lenguaje L del Teorema 3 pertenece a la primera familia, pero no a la segunda.

Un lenguaje que es recursivamente enumerable pero no es recursivo IV

Teorema 5

Existe un lenguaje recursivamente enumerable que no es recursivo; es decir, la familia de lenguajes recursivos es un subconjunto propio de la familia de lenguajes recursivamente enumerables.

Un lenguaje que es recursivamente enumerable pero no es recursivo IV

Teorema 5

Existe un lenguaje recursivamente enumerable que no es recursivo; es decir, la familia de lenguajes recursivos es un subconjunto propio de la familia de lenguajes recursivamente enumerables.

Demostración:

- Considere el lenguaje L del Teorema 3.

Un lenguaje que es recursivamente enumerable pero no es recursivo IV

Teorema 5

Existe un lenguaje recursivamente enumerable que no es recursivo; es decir, la familia de lenguajes recursivos es un subconjunto propio de la familia de lenguajes recursivamente enumerables.

Demostración:

- Considere el lenguaje L del Teorema 3.
- Este lenguaje es enumerable de forma recursiva, pero su complemento no lo es.

Un lenguaje que es recursivamente enumerable pero no es recursivo IV

Teorema 5

Existe un lenguaje recursivamente enumerable que no es recursivo; es decir, la familia de lenguajes recursivos es un subconjunto propio de la familia de lenguajes recursivamente enumerables.

Demostración:

- Considere el lenguaje L del Teorema 3.
- Este lenguaje es enumerable de forma recursiva, pero su complemento no lo es.
- Por lo tanto, según el Teorema 4, no es recursivo, lo que nos da el ejemplo buscado.

■

Un lenguaje que es recursivamente enumerable pero no es recursivo IV

Teorema 5

Existe un lenguaje recursivamente enumerable que no es recursivo; es decir, la familia de lenguajes recursivos es un subconjunto propio de la familia de lenguajes recursivamente enumerables.

Demostración:

- Considere el lenguaje L del Teorema 3.
- Este lenguaje es enumerable de forma recursiva, pero su complemento no lo es.
- Por lo tanto, según el Teorema 4, no es recursivo, lo que nos da el ejemplo buscado.

$\blacksquare$

- Vemos de esto que, de hecho, hay lenguajes bien definidos para los que no se puede construir un algoritmo de pertenencia.

Ejercicios

1. Sea L un lenguaje finito. Demuestre que entonces L^+ es recursivamente enumerable. Sugiera un procedimiento de enumeración para L^+.
2. Sea L un lenguaje libre de contexto. Muestre que L^+ es recursivamente enumerable y sugiera un procedimiento de enumeración para éste.
3. Muestre que si un lenguaje no es recursivamente enumerable, su complemento no puede ser recursivo.
4. Demuestre que la familia de lenguajes recursivamente enumerables es cerrada bajo unión.
5. ¿La familia de lenguajes recursivamente enumerables es cerrada bajo intersección?
6. Demuestre que la familia de lenguajes recursivos es cerrada bajo unión e intersección.