Teoría de la Computación

Clase 28: Lenguajes indecidibles

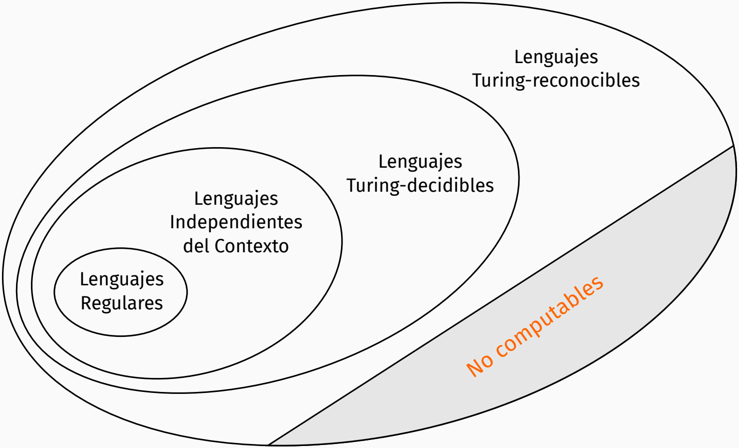
Mauro Artigiani

05 Noviembre 2021

Universidad del Rosario, Bogotá

Introducción

Los lenguajes que hemos visto



Muchos lenguajes, pocas máquinas de Turing

Ideas de la demostración

Para demostrar que hay más lenguajes que máquinas de Turing, procederemos así.

Podemos escribir una lista con todas las `TM`s. A cada `TM` corresponde su lenguaje, que por definición es Turing-reconocible. Esto nos dice que hay tantos lenguajes Turing-reconocibles cuantos números naturales.

Ideas de la demostración

Para demostrar que hay más lenguajes que máquinas de Turing, procederemos así.

Podemos escribir una lista con todas las `TMs`. A cada `TM` corresponde su lenguaje, que por definición es Turing-reconocible. Esto nos dice que hay tantos lenguajes Turing-reconocibles cuantos números naturales.

Un lenguaje es un sobconjunto de Σ^* y, al revés, un subconjunto de Σ^* es un lenguaje. Es decir, los lenguajes son $|\wp(\Sigma^*)|$.

Ideas de la demostración

Para demostrar que hay más lenguajes que máquinas de Turing, procederemos así.

Podemos escribir una lista con todas las `TM`s. A cada `TM` corresponde su lenguaje, que por definición es Turing-reconocible. Esto nos dice que hay tantos lenguajes Turing-reconocibles cuantos números naturales.

Un lenguaje es un sobconjunto de Σ^* y, al revés, un subconjunto de Σ^* es un lenguaje. Es decir, los lenguajes son $|\wp(\Sigma^*)|$.

Si demostramos que $|\mathbb{N}| < |\wp(\Sigma^*)|$ hemos acabado.

Un poco de teoría de conjuntos

Dado que existen infinitos lenguajes e infinitas `TM`s, hay que poner cuidado.

Un poco de teoría de conjuntos

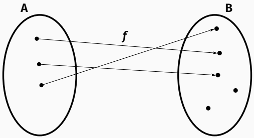
Dado que existen infinitos lenguajes e infinitas `TM`s, hay que poner cuidado.

Para entender en qué sentido un conjunto infinito es "más grande" que otro conjunto infinito, tomamos inspiración en los conjunto finitos.

Un poco de teoría de conjuntos

Dado que existen infinitos lenguajes e infinitas `TMs`, hay que poner cuidado.

Para entender en qué sentido un conjunto infinito es "más grande" que otro conjunto infinito, tomamos inspiración en los conjunto finitos.



Tenemos $|A| \leq |B|$ si y solo si existe una función inyectiva entre A y B.

Un poco de teoría de conjuntos

De manera parecida si A y B son conjuntos finitos, tienen la misma cardinalidad si y solo si existe una función biyectiva entre ellos.

De manera parecida si A y B son conjuntos finitos, tienen la misma cardinalidad si y solo si existe una función biyectiva entre ellos.

Definición

Sean A y B conjuntos cualesquiera. Decimos que $|A| \leq |B|$ si existe una función inyectiva entre A y B. De manera parecida, se tiene que $|A| \geq |B|$ si existe una función sobreyectiva entre A y B. Finalmente, $|A| = |B|$ si existe una función biyectiva entre A y B

Un poco de teoría de conjuntos

De manera parecida si A y B son conjuntos finitos, tienen la misma cardinalidad si y solo si existe una función biyectiva entre ellos.

Definición

Sean A y B conjuntos cualesquiera. Decimos que $|A| \leq |B|$ si existe una función inyectiva entre A y B. De manera parecida, se tiene que $|A| \geq |B|$ si existe una función sobreyectiva entre A y B. Finalmente, $|A| = |B|$ si existe una función biyectiva entre A y B

Por ejemplo, el conjunto de los números pares tiene la misma cardinalidad de los naturales, $\mathbb{N}$. También $|\mathbb{Q}| = |\mathbb{N}|$. Los conjuntos con la cardinalidad de los naturales se dicen numerables.

Codificación

Volvamos ahora a nuestro problema.

Sabemos que una máquina de Turing se puede codificar en una secuencia de 0s y 1s, es decir podemos identificar todas las máquinas de Turing con $\{0,1\}^* = \Sigma^*$.

Codificación

Volvamos ahora a nuestro problema.

Sabemos que una máquina de Turing se puede codificar en una secuencia de 0s y 1s, es decir podemos identificar todas las máquinas de Turing con $\{0,1\}^* = \Sigma^*$. Mostramos ahora que Σ^* es numerable: $|\Sigma^*| = |\mathbb{N}|$.

Codificación

Volvamos ahora a nuestro problema.

Sabemos que una máquina de Turing se puede codificar en una secuencia de 0s y 1s, es decir podemos identificar todas las máquinas de Turing con $\{0,1\}^* = \Sigma^*$. Mostramos ahora que Σ^* es numerable: $|\Sigma^*| = |\mathbb{N}|$.

De hecho, podemos ordenar los elementos de Σ^* por *longitud*, creciendo desde la cadena más corta y por orden lexicográfico a paridad de longitud (orden *shortlex*):

$$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}.$$

Este orden realiza la biyección entre Σ^* y $\mathbb{N}$.

Codificación

Para mostrar que hay más lenguajes que `TMs`, necesitamos contar los posibles lenguajes.

Codificación

Para mostrar que hay más lenguajes que `TM`s, necesitamos contar los posibles lenguajes.

Un lenguaje L es un subconjunto de palabras en Σ^*, y al revés. Es decir:

$$\{\text{lenguajes sobre el alfabeto } \Sigma\} = \wp(\Sigma^*).$$

Codificación

Para mostrar que hay más lenguajes que `TMs`, necesitamos contar los posibles lenguajes.

Un lenguaje L es un subconjunto de palabras en Σ^*, y al revés. Es decir:

$$\{\text{lenguajes sobre el alfabeto } \Sigma\} = \wp(\Sigma^*).$$

Entonces, dado que hay $|\Sigma^*|$ máquinas de Turing y $|\wp(\Sigma^*)|$ lenguajes, queremos demostrar que

$$|\Sigma^*| = |\mathbb{N}| < |\wp(\mathbb{N})| = |\wp(\Sigma^*)|.$$

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$.
Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$. Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Supongamos por absurdo que $f : \mathbb{N} \to \wp(\mathbb{N})$ sea inyectiva y sobreyectiva.

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$. Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Supongamos por absurdo que $f : \mathbb{N} \to \wp(\mathbb{N})$ sea inyectiva y sobreyectiva. Consideramos ahora el conjunto

$$B = \{n \in \mathbb{N} : n \notin f(n)\} \subseteq \mathbb{N}.$$

Por construcción $B \in \wp(\mathbb{N})$.

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$. Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Supongamos por absurdo que $f : \mathbb{N} \to \wp(\mathbb{N})$ sea inyectiva y sobreyectiva. Consideramos ahora el conjunto

$$B = \{n \in \mathbb{N} : n \notin f(n)\} \subseteq \mathbb{N}.$$

Por construcción $B \in \wp(\mathbb{N})$. Siendo f sobreyectiva existe un $a \in \mathbb{N}$ tal que $f(a) = B$.

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$. Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Supongamos por absurdo que $f : \mathbb{N} \to \wp(\mathbb{N})$ sea inyectiva y sobreyectiva. Consideramos ahora el conjunto

$$B = \{n \in \mathbb{N} : n \notin f(n)\} \subseteq \mathbb{N}.$$

Por construcción $B \in \wp(\mathbb{N})$. Siendo f sobreyectiva existe un $a \in \mathbb{N}$ tal que $f(a) = B$.

Tenemos dos casos:

1. Si $a \notin B$ tenemos $a \notin f(a) = B$ y, por definición de B, $a \in B$;
2. Si $a \in B$, tenemos $a \in f(a) = B$ y, por definición de B, $a \notin B$.

Demostración de $|\mathbb{N}| < |\wp(\mathbb{N})|$

Claramente existe una función inyectiva entre $\mathbb{N}$ y $\wp(\mathbb{N})$: $n \mapsto \{n\}$. Entonces $|\mathbb{N}| \leq |\wp(\mathbb{N})|$.

Supongamos por absurdo que $f : \mathbb{N} \to \wp(\mathbb{N})$ sea inyectiva y sobreyectiva. Consideramos ahora el conjunto

$$B = \{n \in \mathbb{N} : n \notin f(n)\} \subseteq \mathbb{N}.$$

Por construcción $B \in \wp(\mathbb{N})$. Siendo f sobreyectiva existe un $a \in \mathbb{N}$ tal que $f(a) = B$.

Tenemos dos casos:

1. Si $a \notin B$ tenemos $a \notin f(a) = B$ y, por definición de B, $a \in B$;
2. Si $a \in B$, tenemos $a \in f(a) = B$ y, por definición de B, $a \notin B$.

En ambos casos tenemos una contradicción, lo que implica que $|\mathbb{N}| \neq |\wp(\mathbb{N})|$. $\square$

Muchos lenguajes, pocas máquinas de Turing

En resumen, tenemos tantas máquinas de Turing cuantos elementos de Σ^*, que es un conjunto numerable, i.e., $|\Sigma^*| = |\mathbb{N}|$.

Muchos lenguajes, pocas máquinas de Turing

En resumen, tenemos tantas máquinas de Turing cuantos elementos de Σ^*, que es un conjunto numerable, i.e., $|\Sigma^*| = |\mathbb{N}|$.

A cada máquina de Turing podemos asociar su lenguaje.

Muchos lenguajes, pocas máquinas de Turing

En resumen, tenemos tantas máquinas de Turing cuantos elementos de Σ^*, que es un conjunto numerable, i.e., $|\Sigma^*| = |\mathbb{N}|$.

A cada máquina de Turing podemos asociar su lenguaje.

Dado que hay tantos lenguajes cuantos elementos de $\wp(\Sigma^*)$ y que $|\wp(\Sigma^*)| = |\wp(\mathbb{N})| > |\mathbb{N}|$, hay (muchos) más lenguajes que máquinas de Turing.

Muchos lenguajes, pocas máquinas de Turing

En resumen, tenemos tantas máquinas de Turing cuantos elementos de Σ^*, que es un conjunto numerable, i.e., $|\Sigma^*| = |\mathbb{N}|$.

A cada máquina de Turing podemos asociar su lenguaje.

Dado que hay tantos lenguajes cuantos elementos de $\wp(\Sigma^*)$ y que $|\wp(\Sigma^*)| = |\wp(\mathbb{N})| > |\mathbb{N}|$, hay (muchos) más lenguajes que máquinas de Turing.

Todo esto es un poco abstracto, la próxima clases lo veremos en un ejemplo concreto.

Resumen

Resumen

Hoy aprendimos:

- Cómo relacionar la cardinalidad de dos conjuntos infinitos;
- Que existen lenguajes no Turing reconocibles y no Turing decidibles.