

Indecidibilidad

Pablo Castro - Computabilidad y Complejidad

Problemas Decidibles vs Indecidibles

En general podemos codificar los problemas cómo lenguajes:

$$L = \{ \langle M, w \rangle \mid \text{la MT } M \text{ acepta } w \}$$

Es decir, si podemos decidir este lenguaje, podemos construir una máquina de Turing que decide si M acepta o no la palabra w

Otro ejemplo:

Si podemos decidir este lenguaje, entonces tenemos un programa que dice si una formula es SAT o no

$$SAT = \{ \langle \phi \rangle \mid \text{si } \phi \text{ es satisfacible} \}$$

Problemas Decidibles:

El siguiente lenguaje es decidible:

$$E_{DFA} = \{ \langle A \rangle \mid A \text{ es un AFD y } L(A) = \emptyset \}$$

Es decir, si el lenguaje de un autómata es vacío.

Idea: Construimos M de la siguiente manera, la entrada es $\langle A \rangle$ la representación de un autómata

- Se marca el estado inicial de $\langle A \rangle$
 - Repetir mientras haya nuevos estados marcados:
 - Marcar cualquier estado que este conectado a un estado ya marcado
 - Si todos los estados están marcados se acepta, sino se rechaza

Problemas Decidibles

El siguiente lenguaje es decidible:

$$A_{CFG} = \{ \langle G, w \rangle \mid G \text{ es un gramática libre de contexto que genera } w \}$$

Para ver que el lenguaje es decidible tenemos que usar la siguientes propiedades:

Forma Normal de Chomsky:

$$\left. \begin{array}{l} A \rightarrow BC \\ A \rightarrow a \end{array} \right\} \text{ Solo producciones de esta forma}$$

Propiedad: Si una gramática está en formal normal de Chomsky entonces todas la derivaciones de w tienen $2 * n - 1$ pasos.
Siendo n el tamaño de w .

Problemas Decidibles

Teniendo en cuenta esta propiedad podemos diseñar la siguiente MT M :

Dada una entrada w ,

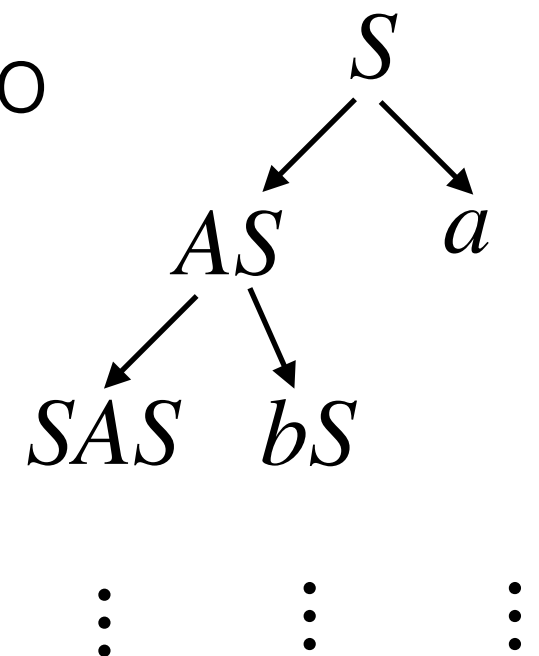
1. Convertir la gramática a Chomsky normal form,
2. Generar todas las derivaciones de $2n-1$ pasos,
3. Si se genera w **aceptar**, en otro caso, **rechazar**:

Podemos hacer un DFS acotado

Gramática:

$$S \rightarrow AS \mid a$$

$$A \rightarrow SA \mid b$$



Problemas indecidibles

Veremos que existen muchos problemas que son indecidibles:

Ejemplo: $A_{TM} = \{\langle M, w \rangle \mid w \text{ es aceptado por } M\}$

Es decir, decir si una máquina de Turing reconoce una palabra

Este lenguaje es reconocible por una MT:

La MT U en la entrada $\langle M, w \rangle$ actúa de la siguiente forma:

1. Simula M en la entrada w
2. Si M acepta, U acepta

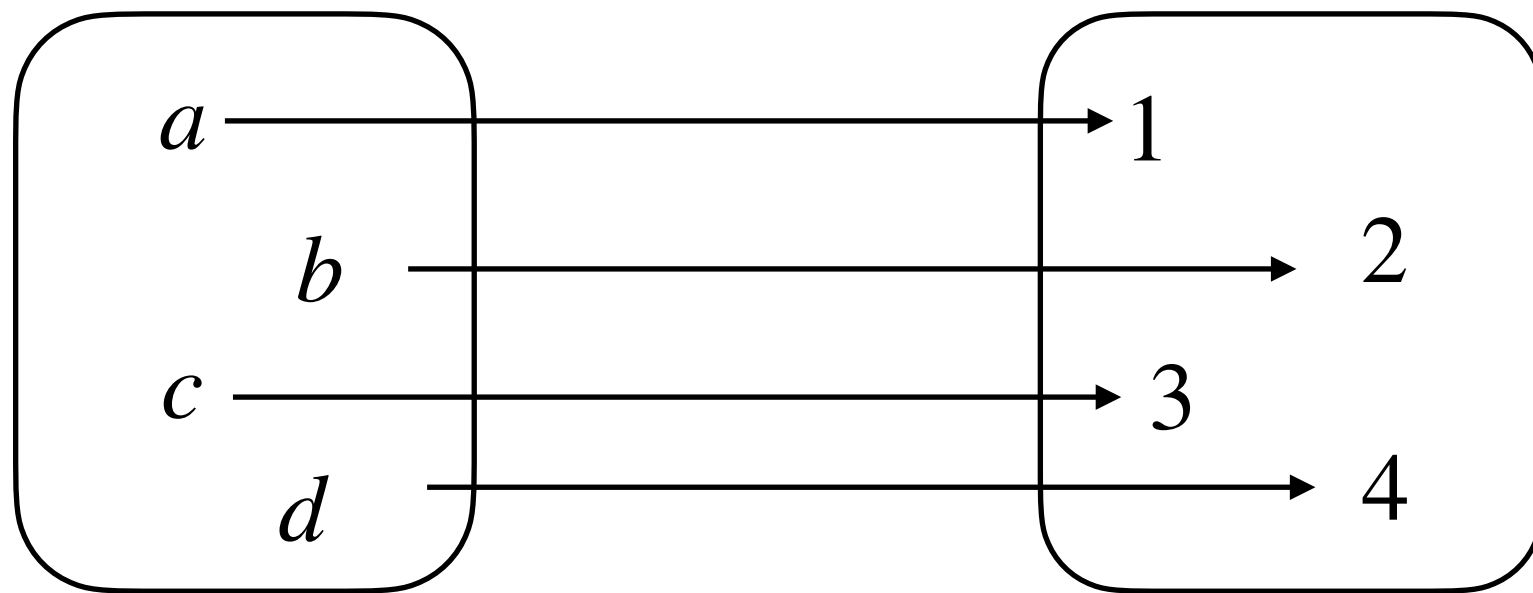
Puede que U no termine

U es una máquina de Turing universal, puede simular cualquier máquina de Turing

Cardinalidad

Cantor introdujo una forma de comparar conjuntos infinitos:

Dos conjuntos tienen la misma cantidad de elementos, si existe una función uno a uno (biyectiva) entre ellos



Esta idea se puede utilizar para
conjuntos infinitos

$$A \cong B$$

Cuando A y B
tienen la misma
cantidad de
elementos

$$A < B$$

Cuando hay una
inyección pero
no biyección

Aritmética Transfinita

Además Cantor demostró que $A < \mathcal{P}(A)$ es decir, que el conjunto partes siempre tiene más elementos que el conjunto original.

La cardinalidad (tamaño) de un conjunto se nota $\#A$

$$\aleph_0 = \#\mathbb{N}$$

$$\aleph_1 = \text{segundo infinito}$$

$$\aleph_2$$

$$\aleph_3$$

\vdots

Existen infinitos \aleph , depende la axiomatización que usemos tenemos

$$\aleph_1 = \mathcal{P}(\aleph_0)$$

Conjunto Contables

Son aquellos que tienen la misma cardinalidad que los \mathbb{N}

Veamos que $\mathbb{N} \cong \mathbb{Z}$, definimos la siguiente biyección:

$$f: \mathbb{N} \rightarrow \mathbb{Z}$$

n	$f(n)$
0	0
1	1
2	-1
3	2
4	-2
5	3
\vdots	\vdots

Es decir:
$$f(n) = \begin{cases} \lceil n \div 2 \rceil & \text{si } n \text{ impar} \\ n \div 2 & \text{si } n \text{ par} \end{cases}$$

Un ejemplo más difícil

Veamos que \mathbb{N} y \mathbb{Q} tienen la misma cardinalidad:

	2	3	4	...	
1	1/1 1	1/2 2	1/3 6	1/4 7	...
2	2/1 3	2/2 5	2/3 8	2/4	
3	3/1 4	3/2 9	3/3	3/4	
4	4/1 10	4/2	4/3	4/4	
⋮					

Podemos definir
un función
biyectiva
recorriendo
diagonalmente

\mathbb{N} vs \mathbb{R}

Habr   m  s reales que naturales? Para ver esto primero comparemos \mathbb{N} con $[0,1)$. Supongamos que hay una biyecci  n:

	1	2	3	4	
1	d_1^1	d_1^1	d_2^1	d_3^1	...
2	d_2^1	d_2^2	d_2^3	d_2^4	
3	d_3^1	d_0^1	d_3^3	d_0^1	
4	d_4^1	d_4^2	d_4^1	d_4^4	
	\vdots				

En donde enumeramos cada n  mero del estilo:

$$0, d_i^1 d_i^2 d_i^3 \dots$$

\mathbb{N} vs \mathbb{R}

Consideremos el siguiente número:

$$0,\bar{d}^1\bar{d}^2\bar{d}^3\bar{d}^4\dots$$

En donde:

$$\bar{d}_j^i = \begin{cases} 1 & \text{si } d_i^j \neq 1 \\ 0 & \text{en otro caso.} \end{cases}$$

Entonces para todo i : $d_i^i \neq \bar{d}_i^i$

Es decir, el número $0,\bar{d}^1\bar{d}^2\bar{d}^3\bar{d}^4\dots$ no está enumerado!

Existencia de Lenguajes no Computables

Veamos que existen lenguajes no computables (no pueden ser reconocidos por MTs):

La cardinalidad del conjunto $\{\langle T \rangle \mid T \text{ es una MT}\}$ es \aleph_0

La cardinalidad de $\{0,1\}^*$ (el lenguaje que contiene todas las palabras) es \aleph_0 también.

Por el teorema de Cantor:

Hay más
lenguajes que
máquinas de
Turing

$$\#\{\langle M \rangle \mid M \text{ es una MT}\} = \#\{0,1\}^* < \#\mathcal{P}(\{0,1\}^*)$$

El Problema de la Terminación

Consideremos el siguiente problema:

$$A_{TM} = \{ \langle M, w \rangle \mid M \text{ acepta } w \}$$

Veamos que es indecidible. Supongamos que existe MT H :

$$H(\langle M, w \rangle) = \begin{cases} \text{acepta} & \text{si } M \text{ acepta } w \\ \text{rechaza} & \text{si } M \text{ no acepta } w \end{cases}$$

El Halting Problem

Podemos construir una máquina D que usa H de la siguiente forma:

Dado un input $\langle M \rangle$, D hace lo siguiente:

1. Simula H con la entrada $\langle M, \langle M \rangle \rangle$
2. Si, H acepta, entonces rechaza
3. Si, H rechaza, entonces acepta.

Se llega a una contradicción, luego H no existe

Corremos D con la entrada $\langle D \rangle$ si D acepta entonces D rechaza la entrada $\langle D \rangle$ (contradicción). Si D rechaza entonces D debería aceptar $\langle D \rangle$ (contradicción.)

El Uso de Diagonalización

Veamos cómo se utilizó la diagonalización:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$...
M_1	A		R	
M_2		R	A	
M_3	A		A	
M_4	R	R		

⋮

En la celda (i,j) tenemos si la MT M_i
acepta o rechaza $\langle M_j \rangle$

Diagonalización (cont)

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$...
M_1	A	R	R	
M_2	R	R	A	
M_3	A	R	A	
M_4	R	R	R	

⋮

H completa la grilla, diciendo en todos los casos si una MT acepta o no cada entrada

Diagonalización (cont)

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	\dots	$\langle D \rangle$
M_1	A	R	R		
M_2	R	R	A		
M_3	A	R	A		
M_4	R	R	R		
\vdots					
D					?

Es A ssi D no acepta $\langle D \rangle$, Es R ssi D acepta $\langle D \rangle$, la MT H no existe.

Un Lenguaje No Reconocible por MT

Usaremos el siguiente teorema:

Teorema: Si un lenguaje L es Turing reconocible, y también \bar{L} es Turing reconocible, entonces L es decidible.

Prueba: Supongamos que M reconoce L y que \bar{M} reconoce \bar{L} . Entonces construimos la MT T de la siguiente forma:

Utiliza dos cintas además de la cinta de entrada. En la cinta 2 simula M , en la cinta 3 simula \bar{M} . Si en algún momento acepta en la cinta 2, entonces se acepta la entrada. Si en algún momento se acepta la de la cinta 3, entonces se rechaza.

Lenguajes no reconocibles

Utilizando el teorema anterior podemos ver que:

$$\overline{A_{TM}} = \{ \langle M, w \rangle \mid M \text{ no reconoce } w \}$$

No es reconocible por una máquina de Turing.

Si fuese reconocible, tendríamos que:

$$\left. \begin{array}{l} A_{TM} \text{ es reconocible} \\ \overline{A_{TM}} \text{ es reconocible} \end{array} \right\} \text{ Por el teorema tendríamos que } A_{TM} \text{ es reconocible}$$

Ejercicios

1. Demostrar que el siguiente lenguaje es decidable:
 $L = \{w \mid w \in \{0,1\}^*\}$
2. Demostrar que el conjunto $\{w \mid w \text{ es una cadena infinita de 0s y 1s}\}$ es no contable
3. Sea C un lenguaje, demostrar que C es reconocible, sii existe un lenguaje decidable D tal que: $\{w \mid \exists y : \langle x, y \rangle \in D\}$
4. Demostrar que el problema de ver si una gramática libre de contexto genera alguna palabra con todos unos es decidable.
5. Escribir en Python una función que enumere todos los pares de naturales.