ÍNDICE

Índice 1

I Clase 1: Conceptos básicos 2
- 1. Esquema de la clase . . . 2
- 2. Parte 1: Alfabetos, Cadenas y Lenguajes . . . 2
- 3. Alfabeto . . . 2
- 4. Concatenación . . . 3
- 5. Lenguaje . . . 3
  - 5.1. Operaciones sobre lenguajes . . . 3
    - 5.1.1. Ejemplos . . . 4
- 6. Especificación Finita de Lenguajes . . . 4
  - 6.1. Los Lenguajes puede representar problemas . . . 4
  - 6.2. Hay Lenguajes imposibles de especificar . . . 5
- 7. Recuerdo: Cardinalidad . . . 5
- 8. Recuerdo: Conjuntos Numerables y No Numerables . . . 5
- 9. Recuerdo: Demostrando Numerabilidad . . . 6

II Parte II 7
- 1. Imposibilidad: Resultados básicos . . . 7
  - 1.1. Existen lenguajes sin especificación finita . . . 7
  - 1.2. Existen lenguajes sin especificación finita (cont. dem.) . . . 7
- 2. Funciones "no programables" . . . 8

I

CLASE 1: CONCEPTOS BÁSICOS

1 Esquema de la clase

1) Conceptos básicos

2) Lenguajes sin representación finita

3) Funciones "no programables"

2 Parte 1: Alfabetos, Cadenas y Lenguajes

Cada rama de la matemática tiene sus objetos de estudio, por ejemplo, conjuntos, números o funciones. En la Teoría de la Computación estudiaremos computadores y computación y por lo tanto necesitaremos un modelo de los objetos que los computadores manipulan:

Objetos básicos en Teoría son: Secuancias de caracteres, caracteres o strings.

3 Alfabeto

Definición 1.1: *Alfabeto*

Un alfabeto es cualquier conjunto finito no vacío. Típicamente denotaremos un alfabeto por Σ. Si $a \in \Sigma$, entonces a es un símbolo o carácter.

Ej: $\Sigma = \{0, 1\}, \Sigma = \{a, ..., z\}$.

Cadenas: (Palabras)

Concatenación de símbolos. Por ej. $aaa, aba, casa$. ¿Definición formal?

Definición 1.2: *Cadenas*

Una cadena, palabra o string s es una secuencia fnita de símbolos en Σ, esto es, s es un elemento de

$$\Sigma^* \stackrel{\text{def}}{=} \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \bigcup_{i=0}^{\infty} \Sigma^i$$

donde

- $\Sigma^0 = \{\varepsilon\}, \Sigma^1 = \Sigma$
- $\Sigma^k = \Sigma \times \Sigma^{k-1}, \forall k > 1$

Σ^* es el conjunto de todas las cadenas de largo finito sobre el alfabeto Σ.

Ej: $\Sigma = \{0, 1\}$,
$\Sigma^* = \{\varepsilon, 0, 1, (0, 0), (0, 1), (1, 0), (1, 1), (0, 0, 0), (0, 0, 1), (0, 1, 0), ...\}$
ε se denomina el **string vacío**.

Pregunta: ¿Una palabra en Σ^*, puede tener largo <u>infinito</u>?

Definición 1.3: ***Largo de una cadena***

Si $s \in \Sigma^\ell$ para un entero positivo ℓ diremos que el largo de s, denotado |s|, es igual a ℓ. Definimos $|\varepsilon| = 0$

Notación:

- Si $\mathrm{s} \in \Sigma^*$, por ejemplo $\mathrm{s} = (a_1, ...a_n)$, simplemente anotaremos $\mathrm{s} = \mathrm{a}_1, ...\mathrm{a}_n$.
- Todos los strings de largo > 0

4 Concatenación

Definición 1.4: ***Concatenación de cadenas***

Si $\mathrm{x} = \mathrm{a}_1 \dots \mathrm{a}_n, \mathrm{y} = \mathrm{b}_1 \dots \mathrm{b}_m \in \Sigma^*$ entonces la concatenación de x e y es $\mathrm{x} \cdot \mathrm{y} = \mathrm{a}_1 \dots \mathrm{a}_n \mathrm{b}_1 \dots \mathrm{b}_m$.

¿Concatenación consigo mismo?:

Para cualquier entero $k > 0$: $\mathrm{x}^k =$ "x concatenado k veces" $= \underbrace{\mathrm{x} \dots \mathrm{x}}_{k}$

Más formalmente,

$$\mathrm{x}^k = \begin{cases} \varepsilon & \text{si } k = 0 \\ \mathrm{x} \cdot \mathrm{x}^{k-1} & \text{si } k > 0 \end{cases}$$

5 Lenguaje

Definición 1.5: ***Lenguaje***

Sea Σ un alfabeto. Un lenguaje sobre Σ es un subconjunto de Σ^*

Ejemplos: $\Sigma = \{\mathrm{a}, \dots, \mathrm{z}\}, L_1 = \{\mathrm{hola}, \mathrm{chao}, \mathrm{xfghj}\}$

5.1 Operaciones sobre lenguajes

Definición 1.6: ***Operaciones***

Sean L_1, L_2, L lenguajes sobre un alfabeto Σ. Definimos las siguientes operaciones sobre lenguajes:

- Unión: $L_1 \cup L_2 = \{\mathrm{w} \in \Sigma^* \mid \mathrm{w} \in L_1 \vee \mathrm{w} \in L_2\}$
- Intersección: $L_1 \cap L_2 = \{\mathrm{w} \in \Sigma^* \mid \mathrm{w} \in L_1 \wedge \mathrm{w} \in L_2\}$
- Diferencia: $L_1 \setminus L_2 = \{\mathrm{w} \in \Sigma^* \mid \mathrm{w} \in L_1 \wedge \mathrm{w} \notin L_2\}$
- Complemento: $L^\complement = \overline{L} = \{\mathrm{w} \in \Sigma^* \mid \mathrm{w} \notin L\} = \Sigma^* \setminus L$
- Concatenación de L_1 con L_2:
$L_1 \circ L_2 = L_1 \parallel L_2 = \{\mathrm{w} = \mathrm{xy} \in \Sigma \mid \mathrm{x} \in L_1 \wedge \mathrm{y} \in L_2\}$
- Potencia: $L^0 = \{\varepsilon\}, L^k = L \parallel L^{k-1}$ para $k > 0$. En otras palabras, $\mathrm{x} \in L^k$ si y solo si $\mathrm{x} = \mathrm{x}_1 \dots \mathrm{x}_k, \mathrm{x}_i \in L$ para $1 \leq i \geq k$.
- Clausura: $L^* = \bigcup_{k \geq 0} L^k$

5.1.1 *Ejemplos*

$$\begin{aligned}\Sigma &= \{0,1,\mathrm{a},\mathrm{b},\mathrm{c},\ldots,\mathrm{z}\},\\ L_1 &= \{\mathrm{a,bb},0\},\\ L_2 &= \{0,1\}^* = \{\varepsilon,0,1,00,01,\ldots\}\end{aligned}$$

$$\begin{aligned}L_1 \cup L_2 &= \{\mathrm{a},\ \mathrm{bb},\varepsilon,0,1,00,01,\ldots\}\\ L_1 \cap L_2 &= \{0\}\\ L_1 \setminus L_2 &= \{\mathrm{a},\ \mathrm{bb}\}\\ \overline{L_1} &= \{\varepsilon,1,\mathrm{b},\ldots,\mathrm{z},00,01,0a,0b,\ldots,\mathrm{a}_0,\mathrm{a}_1,\ldots\}\\ L_1 \parallel L_2 &= \{\mathrm{a},\mathrm{a}0,\mathrm{a}1,\mathrm{a}00,\mathrm{a}01,\ldots,\mathrm{bb},\mathrm{bb}0,\mathrm{bb}1,\mathrm{bb}00,\mathrm{bb}01,\ldots,0,00,01,000,001,\ldots\}\\ L_1^2 &= \{\mathrm{aa},\mathrm{abb},\mathrm{a}0,\mathrm{bba},\mathrm{bbbb},\mathrm{bb}0,0\mathrm{a},0\mathrm{bb},00\}\\ L_1^2 &= \{\mathrm{aaa},\mathrm{aabb},\mathrm{aa}0,\mathrm{abba},\mathrm{abbbb},\ldots,000\}\\ L_1^* &= \{\varepsilon\}\cup\{\mathrm{a},\ \mathrm{bb},0\}\cup\{\mathrm{aa},\ \mathrm{abb},\ \mathrm{a}0,\ \mathrm{bba},\ \mathrm{bbbb},\ \mathrm{bb}0,\ldots\}\cup\ldots\\ L_2^2 &= \{\varepsilon,0,1,00,01,\ldots\}^2 = L_2\end{aligned}$$

6 Especificación Finita de Lenguajes

Veremos ahora cómo definir y usar lenguajes para capturar procesos de cómputo.

Si L es finito, podemos especificarlo por extensión, ej.

$$L_1 = \{\mathrm{aa},\ \mathrm{aba},\ \mathrm{a}\}$$

Si L es infinito, podemos especificarlo en forma finita mediante predicados, por ej.

$$\begin{aligned}L_2 &= \{w \in \{0,1\}^* \mid \text{w comienza con 1 y es número par en binario}\}\\ L_3 &= \{w \in \{\mathrm{a}\}^* \mid \mathrm{w} = \mathrm{a}^p \wedge \text{es un número primo}\}\end{aligned}$$

Si L es especificable en forma finita, ¿son fáciles las siguientes tareas?

- ¿Enumerar todos los elementos en L?
- ¿Saber si x $\in L$?

Por ej.

$$L_4 = \{\mathrm{w} \in \{\mathrm{a}\}^* \mid \mathrm{w} = \mathrm{a}^n \wedge n > 0 \wedge \exists x,y,z \in \mathbb{N}\setminus\{0\}, x^n + y^n = z^n\}$$

está formal y correctamente especificado.

¿Puede saber si $\mathrm{a}^{15} \in L_4$? ¿O si $|L_4| > 2$?

¡Tomó más de 300 años saberlo! (más precisamente, sabe que $L_4 = \{\mathrm{a},\ \mathrm{aa}\}$)

Las dos preguntas de arriba NO son fáciles de resolver.

6.1 Los Lenguajes puede representar problemas

Más aún, si L es

$$L = \{\mathrm{w} \in \{0,1\}^* \mid \text{w es un teorema de teoría de números (escrito en binario)}\}$$

Determinar si un cierto w está o no en L es demostrar un teorema.

Definición 1.7: ***Problema de Decisión, informal***

Diremos que un computador "decide" un lenguaje L si recibiendo como entrada una cadena s puede retornar `'acepta'` si s está en L o `'rechaza'` si no.

Ejemplo: Contestar si "el número 1793773 es primo" es equivalente a determinar si $\mathrm{a}^{1793773} \in L_3$ (usando un computador).

6.2 Hay Lenguajes imposibles de especificar

Demostraremos ahora que hay lenguajes que ningún computador puede "decidir" (esto es, resolver o determinar si un x está en el lenguaje).

Para esto necesitamos recordar un par de cosas de cardinalidad de conjuntos.

7 Recuerdo: Cardinalidad

Definición 1.8: ***Cardinalidad de conjuntos (informal)***

La cardinalidad de un conjunto S, denotado $|S|$ es el número de elementos del conjunto si S es finito.

¿Conjuntos infinitos?

Definición 1.9: ***Comparar cardinalidad entre conjuntos***

- $|A| \leq |B|$ si existe una función $f : A \to B$ inyectiva.
- $|A| \geq |B|$ si existe una función $f : A \to B$ epiyectiva (sobreyectiva).
- $|A| = |B|$ si existe una función $f : A \to B$ biyectiva.
- $|A| < |B|$ si $|A| \leq |B|$ pero $|A| \neq |B|$.
- $|A| > |B|$ si $|A| \geq |B|$ pero $|A| \neq |B|$.

8 Recuerdo: Conjuntos Numerables y No Numerables

Definición 1.10: ***Conjuntos numerables***

La cardinalidad del conjunto de enteros positivos $\mathbb{N}$ es $|N| = \aleph_0$. Todo conjunto S tal que $|S| \leq \aleph_0$ se dice numerable.

Teorema 1.1

- Todo conjunto finito es numerable
- El conjunto potencia de $\mathbb{N}$, $\mathcal{P}(\mathbb{N})$ es no numerable, esto es, $|\mathcal{P}(\mathbb{N})| = \aleph_1 > \aleph_0$.
- El conjunto de números reales $\mathbb{R}$, tiene cardinalidad $\aleph_1$.
- El intervalo de números reales entre 0 y 1, tiene cardinalidad $\aleph_1$.

Demostración omitida

9 Recuerdo: Demostrando Numerabilidad

Para demostrar que un conjunto A es numerable basta mostrar una biyección f entre A y $\mathbb{N}$: listar los elementos de A vía la secuencia $f(0), f(1), f(2), \ldots$ donde cada elemento de A aparece al menos una vez.

Teorema 1.2

Sean A y B dos conjuntos numerables. Entoncces, los siguientes conjuntos son numerables:

- $A \cup B$
- $A \times B$
- A^k para $k \in \mathbb{N}$
- $\bigcup A_i$ donde cada A_i es numerable
- A^+

Demostración en apunte

II

PARTE II

1 Imposibilidad: Resultados básicos

1.1 Existen lenguajes sin especificación finita

Como vimos, un lenguaje puede ser especificado por, un predicado, por ej. En otras palabras, una cadena, define un lenguaje. Otra manera de especificar un lenguaje con una cadena es vía un programa (en `Python`, por ej.) que decide el lenguaje.

En ambos casos, decimos que el lenguaje es representado por un formalismo finito.

Teorema 2.1

Existen lenguajes que no se pueden reperesentar por ningún formalismo finito (ninguna cadena).

Demostración: Intuitivamente, hay más lenguajes distintos que cadenas finitas que pudieran representar el lenguaje. Luego, hay más lenguajes que no poseen representación finita.

¿Conjunto de todas las representaciones finitas? Cada representación es codificable como una cadena binaria finita w, luego es simplemente el conjunto de todas las cadenas binarias

$$\{\mathrm{w} \in \{0,1\}^* \mid \mathrm{w} \text{ es una cadena}\} = \{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, \dots\}$$

¡Este conjunto es numerable!

1.2 Existen lenguajes sin especificación finita (cont. dem.)

¿Conjunto de todos los lenguajes?. Un lenguaje L es por definición un subconjunto de Σ^*. Así, sea $\mathcal{L} = \mathcal{P}(\Sigma^*)$ el conjunto de todos los lenguajes sobre $\Sigma = \{0,1\}$.

$$\begin{array}{rcccccccccccccccccc}
\Sigma^* = & \{ & \varepsilon & , & 0 & , & 1 & , & 00 & , & 01 & , & 10 & , & 11 & , & 000 & , & \dots & \} \\
L = & \{ & & & 0 & , & & & 00 & , & 01 & , & & & & & 000 & , & \dots & \} \\
\chi_L = & & 0 & & 1 & & 0 & & 1 & & 1 & & 0 & & 0 & & 1 & & \dots &
\end{array}$$

Cualquier $L \in \mathcal{L}$ puede ser representado por una cadena binaria infinita χ_L llamada su función característica.

Si $\mathcal{B} = \{\mathrm{w} = \mathrm{w}_1\mathrm{w}_2\dots, \text{donde } \mathrm{w}_i \in \{0,1\}\} = \{0,1\}^\infty$ el conjunto de secuencias binarias infinitas, entonces contiene todas las χ_L para todos los L distintos.

¿Y cuál es la cardinalidad de $\mathcal{B}$?

$\mathcal{B}$ tiene el mismo cardinal que $\mathbb{R}$: ¡Es no numerable!

$\square$

(La demostración es similar a no numerabilidad de intervalo real $[0,1)$.) Por lo tanto, existen lenguajes para los cuales NO podemos dar (no existe) una especificación finita.

Este es un resultado existencial: existen lenguajes que no tienen especificación finita, pero no dice cuáles.

Sin embargo, es posible exhibir ejemplos de lenguajes, los cuales pese a tener especificación finita, no son computables. Los veremos luego en el curso.

2 Funciones "no programables"

Corolario 2.1: ***Existen funciones "no programables"***

Dado cualquier lenguaje de programación, existen funciones de los enteros que no se pueden calcular con ningún programa escrito en este lenguaje.

Dem. (en ucursos)

□