ANÁLISIS COMPARATIVO DE LENGUAJES

Guía de estudio correspondiente a la Teoría sobre: Aspectos Formales de los Lenguajes de Programación

Depto. de Informática

Universidad Nacional de San Luis - Año 2022

ASPECTOS FORMALES DE LOS LENGUAJES DE PROGRAMACIÓN PRIMERA PARTE

Notas de Clase

Capítulos III – Programming Languages – Design and Implementation – Terrence Pratt

Capítulos III Concepts of Programming Languages – Robert Sebesta

Apunte de la cátedra.

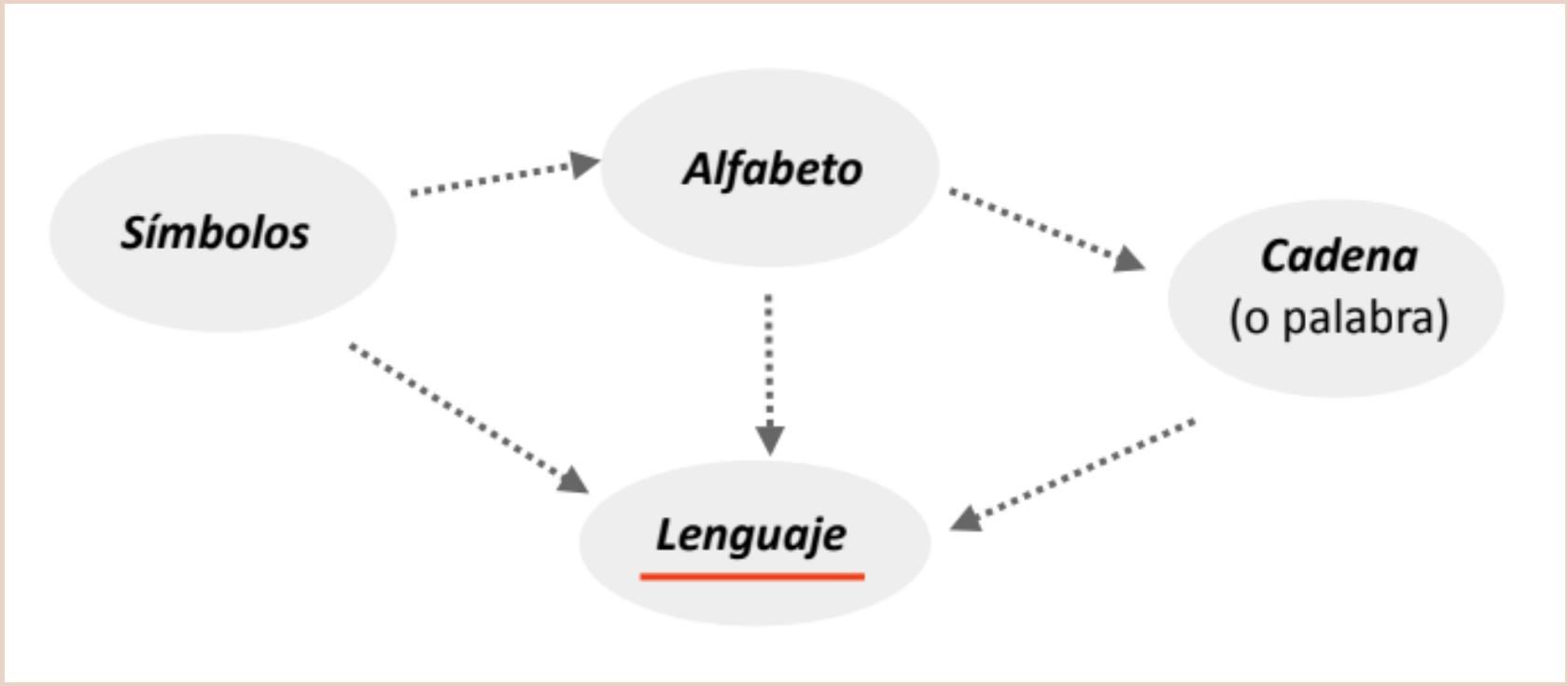
OBJETIVOS DE ESTA TEORÍA

- Introducir métodos formales para la especificación de lenguajes programación.
- Definiciones necesarias para comenzar el estudio de modelos formales.
- Clasificación de lenguajes: Jerarquía de Chomsky.
- Expresiones Regulares.

MODELOS PARA LA DESCRIPCIÓN FORMAL DE LENGUAJES DE PROGRAMACIÓN

- A partir de ahora es nuestro objetivo estudiar métodos que permitan la descripción formal de lenguajes, es decir el cómo describir sintaxis.
- Chomsky (1950, 1959). Lingüista, estudió la naturaleza teórica de los lenguajes naturales.
- En particular nuestro interés es estudiar cómo se definen los lenguajes de programación.
- Para comenzar debemos dar algunas definiciones previas necesarias para poder definir modelos formales:

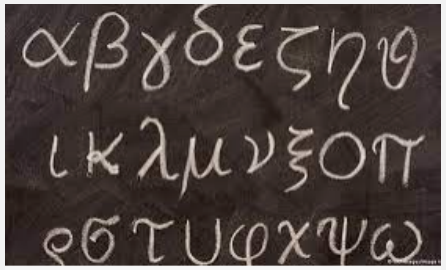
DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS, ALFABETOS Y LENGUAJES



DEFINICIONES Y CONCEPTOS INTRODUCTORIOS

SÍMBOLO

Un **símbolo** es una entidad abstracta a la que no definiremos formalmente, como el punto en geometría. Letras y dígitos son ejemplos de símbolos frecuentemente usados.

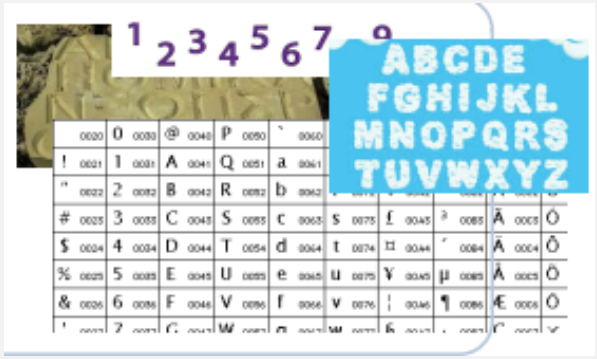


DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
ALFABETOS

Un **alfabeto** es un conjunto finito, no vacío de símbolos. Se utiliza comúnmente el símbolo Σ, para denotarlo.

$\Sigma_1 = \{a,b\}$

$\Sigma_2 = \{0,1\}$



$\Sigma_3 = \{a,b,c,d,\ldots,z\}$

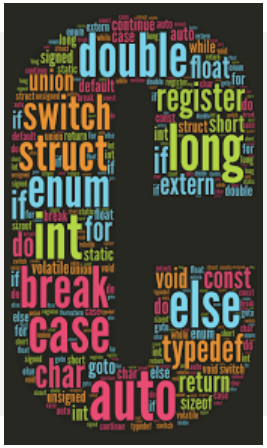
$\Sigma_4 = \{0,1,2,\ldots,9\}$

UNICODE

ASCII

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Una **cadena** o **palabra** es una secuencia finita de símbolos. Por ejemplo, *a*, *b* y *c* son símbolos pertenecientes a un alfabeto y *cccab* es una cadena.



Ejemplos

Σ_1 = {a,b}

x_1 = aabbbaa; x_2 = bbbaa ; ....

Σ_2 = {0,1}

x_1 = 1011011; x_2 = 1100 ; ....

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS

CADENAS

Otros ejemplos de cadenas

```
#include <stdio.h>
#include <stdlib.h>

//declaramos la funcion Suma, que recibe dos enteros
void Suma(int n1, int n2)
{
    int resultado; //para guardar el resultado de la suma
    resultado = n1 + n2; //asigno el resultado de la suma a la var
    printf("La suma de ambos numeros es: %i \n", resultado);
}

int main()
{
    int num1, num2; //creamos dos enteros, num1 y

    //damos valores a estas variables
    num1 = 3;
    num2 = 7;

    //llamada a la función "Suma"
    Suma(num1, num2);

    system("PAUSE");

}
```

```
package ifelse;

import java.util.Scanner;

public class cifra {
    public static void main(String[] args) {
        int num;
        Scanner sc=new Scanner(System.in);
        System.out.println("Ingrese numero");
        num=sc.nextInt();
        if(num>0&&num<10){
            System.out.println("El numero es de un
        }else{
            if(num>=10&&num<100){
                System.out.println("El numero es de
            }else{
                if(num>=100&&num<1000){
                    System.out.println("El numero es
                }else{
                    if(num>=1000&&num<10000){
                    System.out.println("El numero es d
                    }
                }
            }
        }
    }
```

$\Sigma =$

	0020	0	0030	@	0040	P	0050	`	0060	p	0070		00A0	°	00B0	Á	00C0	Đ
!	0021	1	0031	A	0041	Q	0051	a	0061	q	0071	¡	00A1	±	00B1	Á	00C1	Ñ
"	0022	2	0032	B	0042	R	0052	b	0062	r	0072	¢	00A2	²	00B2	Â	00C2	Ò
#	0023	3	0033	C	0043	S	0053	c	0063	s	0073	£	00A3	³	00B3	Ã	00C3	Ó
$	0024	4	0034	D	0044	T	0054	d	0064	t	0074	¤	00A4	´	00B4	Ä	00C4	Ô
%	0025	5	0035	E	0045	U	0055	e	0065	u	0075	¥	00A5	µ	00B5	Å	00C5	Õ
&	0026	6	0036	F	0046	V	0056	f	0066	v	0076	¦	00A6	¶	00B6	Æ	00C6	Ö
'	0027	7	0037	G	0047	W	0057	g	0067	w	0077	§	00A7	·	00B7	Ç	00C7	×

```
<html>
<head>
<title>contact us
<meta http-equiv="
<link rel="short
<script type="t
</head>
class
```

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

- ***Cadena vacía***

  Es la cadena con 0 ocurrencias de símbolos y denotada por la letra griega λ

  (sin importar el alfabeto)

Ejemplos

$\Sigma_1 = \{a,b\}$

$x = \lambda$

$\Sigma_2 = \{0,1\}$

$x = \lambda$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Operaciones sobre cadenas

Longitud de una cadena denotada por $|x|$

Cantidad de símbolos de la cadena x

$x = a_1\ a_2\ a_3 \ldots a_n$ ➡ $|x| = |\ a_1\ a_2\ a_3 \ldots a_n\ | = n$

Ejemplo

$|x| = |1011011| = 7$

$|x| = |123+32*339+57/65| = 16$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Operaciones sobre cadenas

Longitud de una cadena: cadena nula

$x = \lambda \Rightarrow |x| = |\lambda| = 0$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Operaciones sobre cadenas

Concatenación de cadenas:

Es una operación binaria que permite concatenar cadenas

(se colocan los símbolos de una cadena y a continuación de los símbolos de la cadena x)

$x = a_1\ a_2\ a_3\ \dots\ a_n$

$y = b_1\ b_2\ b_3\ \dots\ b_m$

$\Rightarrow$

$x\ y = a_1\ a_2\ a_3\ \dots\ a_n\ b_1\ b_2\ b_3\ \dots\ b_m$

$|x\ y| = n + m$

La cadena vacía (λ) es el neutro de la concatenación

$$\lambda x = x\lambda = x$$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Operaciones sobre cadenas

Potencia sobre cadenas:

Potencia sobre Cadenas ⇨ x^n

$x^0 = \lambda$

$x^n = x^{n-1}\, x$

Ejemplos:

Si x= abaa entonces:

x^2= abaaabaa

x^3= abaaabaaabaa

¿Cómo se obtiene x^3 según la definición?

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
CADENAS

Operaciones sobre cadenas

Potencia sobre Alfabetos Σ^n

Es el conjunto de todas las cadenas de longitud n cuyos símbolos están en Σ.

$\Sigma^0 = \{ \lambda \}$

$\Sigma^n = \{ x_1, x_2, ..., x_i, ... \}$ donde $\forall x_i \ |x_i| = n$

Ejemplos:

1. Notar que $\Sigma^0 = \{\lambda\}$, independientemente de lo que sea Σ. Dado que λ es la única cadena cuya longitud es 0.
2. Si $\Sigma = \{0, 1\}$, entonces $\Sigma^1 = \{0, 1\}$,
3. $\Sigma^2 = \{00, 01, 10, 11\}$,
4. $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$ y así siguiendo.

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Un lenguaje L es un conjunto (finito o infinito) de cadenas sobre algún alfabeto Σ

Ejemplos

$\Sigma_1 = \{a,b\}$

$L_1 = \varnothing$; $L_2 = \{aabbbaa, bbaa, bb, b, bbbaaaabbb, ...\}$; ....

$\Sigma_2 = \{0,1\}$

$L_1 = \{0,1\}$; $L_2 = \{1001, 100, 1, 000, 11111, ... \}$; ...

Observación: el lenguaje L no necesariamente debe incluir todos los símbolos del alfabeto

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Otros ejemplos de Lenguajes

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Lenguaje Referencial

Es el conjunto de todas las cadenas sobre algún alfabeto Σ y es denotado por Σ^*

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots = \{\lambda\} \cup \Sigma^1 \cup \Sigma^2 \cup \ldots = \bigcup_{n \in (\mathbb{N} \cup \{0\})} \Sigma^n$$

Ejemplo:

$$\Sigma = \{0,1\} \quad \Sigma^* = \{\lambda, 0, 1, 00, 11, 010, 000, \ldots\}$$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

- Σ^* es un conjunto infinito de cadenas de longitud finita, ya que Σ es un conjunto finto no vacío de símbolos.

- Para cualquier lenguaje L se cumple que $L \subseteq \Sigma^*$

- Y además:

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup .... = \bigcup_{n \in \mathbb{N}} \Sigma^n$$

$$\Sigma^+ = \Sigma^* - \Sigma 0$$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Formas de describirlos

Primera opción: descripción del conjunto por extensión

$L = \{x_1, x_2, x_3, x_4, \ldots\}$

Ejemplo:

Σ = {b, c} L = {c, cc, ccc, cccc}

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Formas de describirlos

Segunda opción: descripción del conjunto por comprensión

$$L = \{x \in \Sigma^* / \mathcal{P}(x)\}$$

Ejemplo:

$\Sigma = \{a, b\}$

$$L = \{x \in \Sigma^* / \overbrace{x \text{ contiene una cantidad par del símbolo a y no contiene el símbolo b}}^{\mathcal{P}(x)}\}$$

DEFINICIONES Y CONCEPTOS INTRODUCTORIOS
LENGUAJES

Formas de describirlos

Otra opción: remplazar x por una expresión con parámetros y describir el lenguaje estableciendo las condiciones sobre los parámetros.

Ejemplo

$\Sigma = \{a, b\}$

$L = \{a^n / n \in \mathbb{N} \text{ y } n \text{ es par}\}$

(expresión, donde n es el parámetro)

(condición sobre el parámetro n)

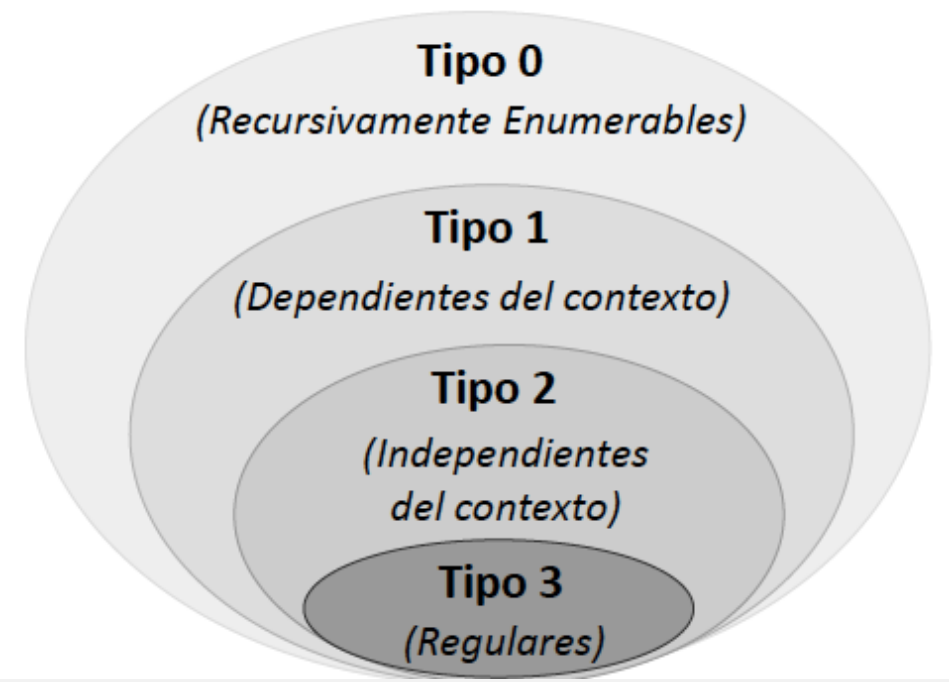
Observación: también es posible describir un lenguaje, utilizando lenguaje natural.

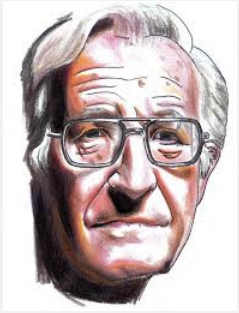
LENGUAJES FORMALES

Modelos descriptores de Lenguajes:

- ✓ Dispositivos reconocedores: Autómatas o Máquinas abstractas o Máquinas de Estados
- ✓ Dispositivos generadores: Gramáticas

Jerarquía de Chomsky





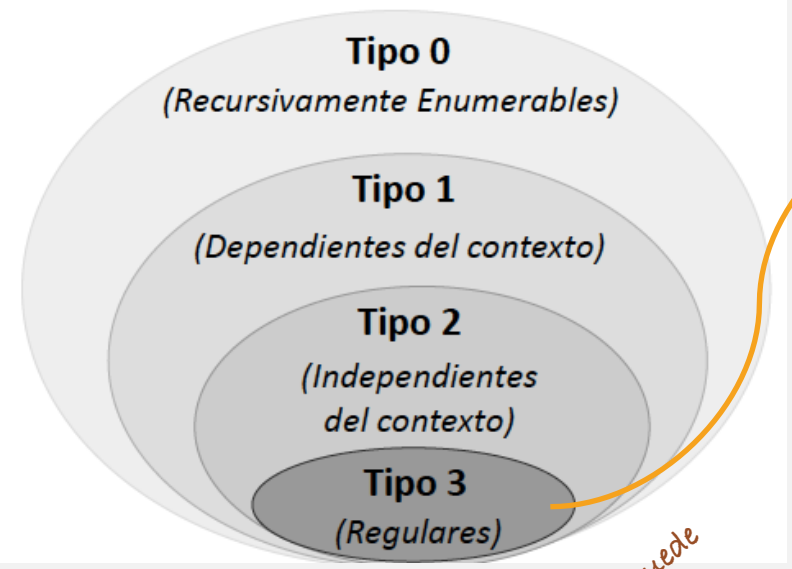
Noam Chomsky

DISPOSITIVOS DESCRIPTORES DE LENGUAJES

Gramáticas: es un modelo matemático que permite generar a través de reglas sintácticas o gramaticales, cadenas miembros de un leguaje específico.

Autómatas: es un modelo matemático que representa la idea de computación o manipulación de cadenas vía la aplicación de acciones preestablecidas. Tiene como objetivo (en general), determinar la pertenencia de una cadena a un lenguaje específico.

LENGUAJES REGULARES



Lenguajes Regulares

$\Sigma = \{0, 1\}$

$L_1 = \{\lambda, 00, 11\}$

$L_2 = \{0^i 1 / 1 \leq i \leq 5\}$

Finitos

También se puede escribir como

$L_3 = \{x \in \Sigma^* / x \text{ comienza con } 0\}$

$L_4 = \{1^k / k > 0\}$

$L_5 = \{0^i 1^k / i \geq 1 \text{ y } k > 0\}$

Infinitos

$L = \{x \in \{0,1\}^* / x \text{ comienza con } 0\}$

LENGUAJES REGULARES

- Otros ejemplos de lenguajes regulares son:

Los nombres de los identificadores, palabras claves, números, etc., en los lenguajes de programación

EXPRESIONES REGULARES (ER)

- Las **ER** son un tipo de notación particular que permite describir **lenguajes regulares**.
- Ofrecen una forma declarativa de expresar las cadenas.
- Se construyen utilizando los siguientes elementos:

- **operandos:**

  los símbolos σ en Σ

  la palabra vacía (λ)

  el conjunto vacío (∅)

  expresiones regulares

- **operadores:**

  unión (+)

  concatenación (.)

  clausura de Kleene (*)

EXPRESIONES REGULARES (ER)

- La ER se definen recursivamente de la siguiente manera:

$\emptyset$ es una ER y denota el conjunto vacío, $L(\emptyset) = \emptyset$.

λ es una ER y denota el conjunto $\{\lambda\}$. $L(\lambda) = \{\lambda\}$.

$\forall a \in \Sigma$, a es una ER y denota el conjunto $\{a\}$. $L(a) = \{a\}$.

Si α y β son ER, entonces:

$\alpha + \beta$ es una ER y denota el conjunto $L(\alpha + \beta) = L(\alpha) \cup L(\beta)$

$\alpha\beta$ es una ER y denota el conjunto $L(\alpha\beta) = L(\alpha)L(\beta)$

α^* es una ER y denota el conjunto $L(\alpha^*) = (L(\alpha))^*$

(α) es una ER y denota el conjunto $L((\alpha)) = L(\alpha)$

Ninguna otra cosa es una ER.

EXPRESIONES REGULARES (ER)

La **Precedencia** de los operadores (de mayor a menor): clausura, concatenación y unión. La precedencia puede alterarse usando paréntesis.
La **Asociatividad**: de izquierda a derecha.

Ejemplos:

Considerando $\Sigma = \{a, b\}$

- a es una ER que denota el lenguaje $L = \{a\}$
- b es una ER que denota el lenguaje $L = \{b\}$
- $a + b$ es una ER que denota el lenguaje $L = \{a\} \cup \{b\} = \{a, b\}$
- ab es una ER que denota el lenguaje $L = \{a\}.\{b\} = \{ab\}$
- $(ab)^*$ es una ER que denota el lenguaje $L = \{ab\}^*$
- $(a + b)(ab)^*$ es una ER que denota el lenguaje $L = \{a, b\}\{ab\}^*$

EXPRESIONES REGULARES (ER)

Construir una ER para cada uno de los siguientes lenguajes, Considerando $\Sigma = \{a, b\}$:

- $\{\lambda, aab, baa\}$
- $L = \{x \in \Sigma^* / x \text{ comienza con b y termina con a}\}$
- $L = \{x \in \Sigma^* / x \text{ contiene la subcadena ab}\}$
- El lenguaje de todas las cadenas que tienen número par de símbolos.

EXPRESIONES REGULARES (ER)

- **$(\lambda + aab + baa)$**
- **$b\ (a + b)^*\ a$**
- **$(a + b)^*\ ab\ (a + b)^*$**
- **$(aa + ab + ba + bb)^*$**

Dar ejemplos de cadenas pertenecientes al lenguaje denotado por cada una de las ER. (desarrollado en clase)

OTROS USOS DE LAS ER

- Describir patrones para búsqueda en textos (palabras terminadas en aba, números telefónicos, direcciones de mail, etc., etc.).
- Generar Analizadores Lexicográficos.
- Comandos de búsqueda en SO como Linux (ejemplo: **$ ls -l *.ps | grep May**).
- Etc.

¿DUDAS?