

## Práctico 2 - Lenguajes y Expresiones Regulares

### Ejercicio 0:

- Demostrar el siguiente teorema: "Todo lenguaje finito es regular".
- Demostrar el siguiente teorema: "Dado un alfabeto  $\Sigma$ ,  $\Sigma^*$  es regular".
- Demostrar que los siguientes lenguajes son regulares utilizando la definición:
  - $L = \{0^{2i}, i \geq 1\}$  con  $\Sigma = \{0, 1\}$ .
  - $L = \{x / \text{la longitud de } x \text{ es par}\}$  con  $\Sigma = \{0, 1\}$ .

a) Teóricamente:

b)

c)

i.  $L = \{0^{2i}, i \geq 1\}$  con  $\Sigma = \{0, 1\}$ :

$i \geq 1$ , caso base 1 cumplido.

ii.  $L = \{x / \text{la longitud de } x \text{ es par}\}$  con  $\Sigma = \{0, 1\}$ :

### Ejercicio 1:

Escribir expresiones regulares que definan los siguientes lenguajes:

- $L = \{x / x = a^i b^j \vee x = (cd)^{2n+1}; i \geq 0; j, n \geq 1\}$ .
- $L = \{xy / x = a^{2p+1} \wedge (y = c^i d^j \vee y = a^n b^m); p \geq 1; i, j > 0; n, m \geq 1\}$ .
- $L = \{xy / x \in (0, 1)^+ \wedge (y = c^n b^m \vee y = c^n d^i); n \geq 0; i, m \geq 1\}$ .
- $L = \{x / x = a^m b^n c d^p; m, n > 0; p \geq 0\}$ .
- $L = \{x / x = a w c^n; n > 1; w \in \{a, b\}^*\}$ .
- $L = \{0^{2i}, i \geq 1\}$  con  $\Sigma = \{0, 1\}$ .
- El conjunto de todas las cadenas pertenecientes a  $\Sigma = \{0, 1\}$  que tienen tres ceros consecutivos.
- $L_1 = \{a^{2k} b^{3n} / k \geq 1, n \geq 0\}$ .
- $L_2 = \{(ab)^n c (ba)^{2m+1} / n \geq 1, m > 0\}$ .

a)  $L = \{x / x = a^i b^j \vee x = (cd)^{2n+1}; i \geq 0; j, n \geq 1\}$

$a^i b^j$  con  $i \geq 0$  y  $j \geq 1$ . Todas las potencias empezando con  $a^0$  y  $b^1$   $\Rightarrow a^*b^+$   
 $(cd)^{2n+1}$  Todas las potencias impares empezando con  $cd^3$   $\Rightarrow cd(cdcd)^+$

Resultado =  $a^*b^+ \mid cd(cdcd)^+$ .

Ejemplos = b, ab, abb, cdcdcd, cdcdcdcdcd, aab.

b)  $L = \{xy / x = a^{2p+1} \wedge (y = c^i d^j \vee y = a^n b^m); p \geq 1; i, j \geq 0; n, m \geq 1\}$ .

$a^{2p+1}$  con  $p \geq 1$ . Todas las potencias impares arrancando con  $a^3$   $\Rightarrow a(aa)^+$ .

$c^i d^j$  con  $i, j > 0$ . Todas las potencias arrancando en  $c^1 d^1$   $\Rightarrow (c^*d^*)$ .  
 $a^n b^m$  con  $n, m \geq 1$ . Todas las potencias arrancando con  $a^1 b^1$   $\Rightarrow (ab)^+$ .

Resultado =  $a(aa)^+ \cdot ((c^*d^*) \mid (ab)^+)$ .

Ejemplo: aaa, aaaab, aaac, aaad, aaacd, aaaaaab, aaaaa, aaaaacd.

c)  $L = \{xy \mid x \in (0, 1)^+ \wedge (y = c^n b^m \vee y = c^n d^i); n \geq 0; i, m \geq 1\}$ .

$x \in (0, 1)^+$ . Conjunto positivo de  $\{0, 1\}$   $\Rightarrow (01)^+$ .

$c^n b^m$  con  $n \geq 0$  y  $m \geq 1$ . Todas las potencias arrancando en  $c^0 d^1$   $\Rightarrow (c^*d^+)$ .

$c^n d^i$  con  $n \geq 0$  e  $i \geq 1$ . Todas las potencias arrancando en  $c^0 d^1$   $\Rightarrow (c^*d^+)$ .

Como  $(c^*d^+) \mid (c^*d^+) = (c^*d^+) \Rightarrow$  Resultado =  $(01)^+ \cdot (c^*d^+)$ .

Ejemplo: 01d, 01cd, 0101dd.

d)  $L = \{x \mid x = a^m b^n c d^p; m, n > 0; p \geq 0\}$ .

$a^m b^n c d^p$  con  $m, n \geq 1, p \geq 0$ .

Resultado =  $(ab)^+ \cdot c \cdot d^*$ .

Ejemplo: abc, ababcd, abcd, abccdd.

e)  $L = \{x \mid x = a w c^n; n > 1; w \in \{a, b\}^*\}$ .

Resultado =  $a \cdot a^* \cdot b^* \cdot c^+$ .

Ejemplo: ac, acc, aac, abc, aabcc.

f)  $L = \{0^{2i} \mid i \geq 1\}$  con  $\Sigma = \{0, 1\}$

$0^{2i}$  con  $i \geq 1$ . Todas las potencias pares arrancando con  $i = 2$   $\Rightarrow 00(00)^*$ .

Resultado =  $0(0)^+ \cdot 1$ .

Ejemplo: 001, 00001, 0000001.

g) El conjunto de todas las cadenas pertenecientes a  $\Sigma = \{0, 1\}$  que tienen tres ceros consecutivos.

Resultado =  $(000 \cdot 1^* \cdot 0^*)^+$

Ejemplo = 000, 0000, 00010, 000111, 0001000.

h)  $L_1 = \{a^{2k} b^{3n} \mid k \geq 1, n \geq 0\}$ .

$a^{2k}$  Todas las potencias pares arrancando en  $k = 2$   $\Rightarrow aa(aa)^*$

$b^{3n}$  Todas las potencias multiples de 3, empezando en  $n = 0$   $\Rightarrow (bbb)^*$

Resultado =  $aa(aa)^* \cdot (bbb)^*$ .

Ejemplo: aa, aabbbb, aaaabbbb, aaaa, aaaabbbbbbb.

i)  $L_2 = \{(ab)^n c (ba)^{2m+1} \mid n \geq 1, m > 0\}$ .

$(ab)^n$  Todas las potencias empezando con  $n = 1$ .  $\Rightarrow (ab)^+$

c

$\Rightarrow c$

$(ba)^{2m+1}$  Todas las potencias impares empezando con  $ba^1$   $\Rightarrow ba(baba)^*$

Resultado =  $(ab)^+ \cdot c \cdot ba(baba)^*$

Ejemplo: abcba, abcbababa, ababcba, ababcbababa.

### Ejercicio 2:

Considerando las siguientes definiciones regulares definidas sobre  $\Sigma = \{-, ., \_, a-z, A-Z, 0-9\}$ :

- Dígito =  $1 / 2 / \dots / 9$ .
- Cero =  $0$ .
- Número = cero / dígito.
- Punto =  $.$
- Guión =  $-$ .
- Subrayado =  $\_$ .
- Letra =  $a / b / \dots / z / A / B / \dots / Z$ .

Indicar cuál es el lenguaje denotado por cada una de las siguientes expresiones regulares (definirlo conceptualmente):

- a) Número número\* punto número\*.
- b) Número\* (punto /  $\lambda$ ).
- c) (Cero / dígito número\*) punto (cero / número\* dígito).
- d) Letra (letra / número).
- e) Letra (letra / número / guión (letra / número))\*.
- f) (letra / subrayado) (letra / número / subrayado)\*.

- a)  $L = \{x / x \text{ es un número real positivo con punto decimal}\}$ .
- b)  $L = \{x / x \text{ es un número entero}\}$ .
- c)  $L = \{x / x \text{ es un número flotante con punto decimal}\}$ .
- d)  $L = \{x / x \text{ es un número binario, octal o hexadecimal en formato C}\}$ .
- e)  $L = \{x / x \text{ es una cadena de caracteres con letras, números, o un guión seguido de una letra o un número y que siempre comienza con una letra}\}$ .
- f)  $L = \{x / x \text{ es el nombre de una variable o palabra reservada}\}$ .

### Ejercicio 3:

Dar, cuando sea posible, una expresión regular que describa los siguientes conjuntos de cadenas:

- a)  $\{w / w \in \{a, b, c\}^* \text{ y } w \text{ comienza con } a\}$ .
- b)  $\{a^{3i} / i \geq 0\}$ .
- c)  $(\{b\} \cup \{c\}) \cdot (\{a\} \cup \{b\} \cup \{c\})^*$ .
- d)  $\{a^{3i} b^i / i \geq 0\}$ .

- a)  $\{w / w \in \{a, b, c\}^* \text{ y } w \text{ comienza con } a\} \Rightarrow (a(bc)^*)^+$ .
- b)  $\{a^{3i} / i \geq 0\} \Rightarrow (aaa)^*$ .
- c)  $(\{b\} \cup \{c\}) \cdot (\{a\} \cup \{b\} \cup \{c\})^* \Rightarrow (b|c) \cdot (a|b|c)^*$ .
- d)  $\{a^{3i} b^i / i \geq 0\} \Rightarrow (aaab)^*$ .

### Ejercicio 4:

Dadas las siguientes expresiones regulares, generar cadenas para cada una de ellas:

- a)  $0 \cdot 0 \cdot (0 / 1)^*$ .
- b)  $((0 \cdot 1) / 1)^*$ .
- c)  $(a / b)^* \cdot (a \cdot a / b \cdot b)$ .
- d)  $a \cdot a / (a \cdot a)^*$ .

- a)  $0 \cdot 0 \cdot (0 / 1)^*$ :  
 $w_s = 00, 000, 0010, 001001$ .
- b)  $((0 \cdot 1) / 1)^*$ :  
 $w_s = \lambda, 01, 1, 011, 01101$ .
- c)  $(a / b)^* \cdot (a \cdot a / b \cdot b)$ :  
 $w_s = aa, bb, aaa, bbb, aaabbbabb$ .

- d)  $a . a / (a . a)^*$ :  
 $w_s = aa, \lambda, aaaa, aaaaaa$ .

### Ejercicio 5:

Generar una expresión regular para cada uno de los siguientes lenguajes. Indicar cuáles son los criterios tomados. Por ej., para el punto a) se puede tomar como fecha válida los formatos mm/dd/yyyy o dd-mm-yyyy o dd-mm-yy, etc.

- a) Cadenas que representen una fecha válida.
- b) Cadenas que representen una hora válida.

- a) Cadenas que representen una fecha válida:

dd-mm-yyyy  $\Rightarrow ((0-2)(1-9)) | 3 (0-1) \text{ guión } 0(1-9) | 1(0-2) \text{ guión } (0-9)(0-9)(0-9)(0-9)$

- b) Cadenas que representen una hora válida:

hh-mm-ss  $\Rightarrow (0|1)(0-9) | 2(0-3) \text{ guión } (0-5)(0-9) \text{ guión } (0-5)(0-9)$

### Ejercicio 6:

Escribir expresiones regulares para cada uno de los siguientes lenguajes:

- a) Códigos postales. (Por ej: 1234, 8366, pero no 422 o 0227).
- b) Números de patentes Mercosur.
- c) Comentarios acotados por /\* y \*/.  $\Sigma = \{\text{letras}, *, /\}$ .
- d) Identificadores de cualquier longitud que empiecen con una letra y contengan letras, dígitos o guiones. No pueden terminar con guión.
- e) Ídem anterior pero que no contengan dos guiones seguidos.
- f) Constantes en otras bases como las del lenguaje C.
- g) Constantes aritméticas enteras. Controlar el rango permitido.
- h) Constantes reales con formato xx.xx. Controlar el rango permitido.
- i) Constantes string de la forma "texto".
- j) Palabras reservadas (IF-WHILE-DECVAR-ENDDEC-INTEGER-FLOAT-WRITE).
- k) Operadores lógicos y operadores aritméticos básicos.

- a) Códigos postales. (Por ej: 1234, 8366, pero no 422 o 0227)  
 $(1-9)(0-9)(0-9)(0-9)$

- b) Números de patentes Mercosur:  
 $(A-Z)(A-Z) (0-9)(0-9)(0-9) (A-Z)(A-Z)$

- c) Comentarios acotados por /\* y \*/.  $\Sigma = \{\text{letras}, *, /\}$ :  
 $/* (\text{letras})^* */$

- d) Identificadores de cualquier longitud que empiecen con una letra y contengan letras, dígitos o guiones. No pueden terminar con guión:  
 $(\text{letra}) (\text{dígito}|\text{letra}|\text{guión})^* (\text{dígito}|\text{letra})$

- e) Ídem anterior pero que no contengan dos guiones seguidos:  
 $\text{letra} ((\text{dígito}|\text{letra})^* \text{guión} (\text{dígito}|\text{letra})^*)^* (\text{dígito}|\text{letra})$

- f) Constantes en otras bases como las del lenguaje C:  
 $0(b|0|1)^+ | x(0 | 1 | 2 | \dots | A | \dots | F)^+ | (0 | 1 | \dots | 7)^+$

- g) Constantes aritméticas enteras. Controlar el rango permitido:  
 $\text{formato } xxxx \Rightarrow (- | \lambda) (\text{dígito} . \text{numero}^*) | 0$

- h) Constantes reales con formato xx.xx. Controlar el rango permitido:

Numero = 0 | 1 | ... | 9

Formato xx.xx => (- | λ) (numero.numero) punto (numero.numero)

i) Constantes string de la forma "texto":

letra = (a-z|A-Z)\*

Dígito = (0-9)\*

" (letra | dígito)\* "

j) Palabras reservadas (IF-WHILE-DECVAR-ENDDEC-INTEGGER-FLOAT-WRITE):

IF | WHILE | DECVAR | ENDDEC | INTEGER | FLOAT | WRITE

k) Operadores lógicos y operadores aritméticos básicos:

AND | OR | NOT | XOR | + | - | / | \* | > | >= | < | <= | =