

AUTÓMATAS Y GRAMÁTICAS

TRABAJO PRÁCTICO N° 1

**IMPLEMENTACIÓN CON PYTHON PARA VALIDACIÓN DE DATOS
MEDIANTE EXPRESIONES REGULARES**

Contenido Conceptual

Conceptos fundamentales: cadenas, lenguajes, operaciones con los lenguajes, expresiones regulares, definiciones regulares.

Objetivos

- Identificar cadenas y lenguajes.
- Aplicar diferentes operaciones a cadenas y lenguajes.
- Adquirir la habilidad para identificar y crear expresiones regulares que definan cadenas.

PARTE A: Conceptos fundamentales: cadenas, lenguajes, operaciones con los lenguajes.

Definiciones

Alfabeto: conjunto de símbolos finito y no vacío, por ejemplo, las letras y los caracteres.

Cadena: es una secuencia finita de símbolos seleccionados de algún alfabeto. La longitud de una cadena s , $|s|$, es el número de apariciones de símbolos en s .

Cadena vacía: Representa cero apariciones de símbolos. Esta cadena ϵ es una cadena que puede construirse en cualquier alfabeto.

Operaciones aplicadas a cadenas:

- **Concatenación:** de x e y , xy , es la cadena que resulta de agregar y a x .
- **Potencia:** cuando se considera la concatenación como un “producto”.
- **Prefijo:** una cadena que se obtiene eliminando cero o más símbolos desde la derecha otra cadena.

AUTÓMATAS Y GRAMÁTICAS

- **Sufijo:** una cadena que se forma suprimiendo cero o más símbolos desde la izquierda de otra cadena.
- **Subcadena:** una cadena que se forma suprimiendo un prefijo y un sufijo de otra cadena.

Lenguaje: conjunto de cadenas, todas ellas seleccionadas de un Σ^* , donde Σ es un determinado alfabeto se denomina lenguaje.

Operaciones con Lenguajes:

Sean A y B dos lenguajes se pueden definir las siguientes operaciones:

- **Unión:** $A \cup B = \{ x \mid x \text{ está en } A \text{ o } x \text{ está en } B \}$
- **Intersección:** $A \cap B = \{ x \mid x \text{ está en } A \text{ y } x \text{ está en } B \text{ simultáneamente} \}$
- **Concatenación:** $A.B = \{ w.x \mid w \text{ está en } A \text{ y } x \text{ está en } B \}$
- **Potencia:** Si $A = \{ab\}$, $A^0 = \varepsilon$, $A^1 = A = \{ab\}$, $A^2 = A.A = \{abab\}$, $A^3 = A.A.A = \{ababab\}$
- **Cerradura de Kleene:** Es el lenguaje A compuesto por todas las cadenas sobre un alfabeto dado Σ . Si $\Sigma = \{1\}$, $A^* = \{ \varepsilon, 1, 11, 111, 1111, \dots \}$, A^* es infinito.

Ejercicio 1

Dado el alfabeto $\Sigma = \{a,b,c,d,0,1,2,3,4\}$, obtenga dos cadenas, x e y, del alfabeto dado, indique la longitud de cada cadena y aplique las siguientes operaciones: concatenación de x e y, las potencias x^0 , x^1 , y^2 , y^3 .

Ejemplo: si $x = abc123$ e $y = d4$, $|x| = 6$, $xy = abc123 d4$, $x^0 = \varepsilon$, $y^3 = d4 d4 d4$

Ejercicio 2

Dados los siguientes lenguajes, A, el conjunto de letras y B, el conjunto de dígitos, realice las siguientes operaciones $A \cup B$, $A \cap B$, $A.B$, A^3 , B^2 , B^0 , A^* , $A(A \cup B)^*$.

Ejemplo:

- AA es el conjunto de cadenas que consta de una letra seguida de otra letra.
- A^2 es el conjunto de todas las cadenas de dos letras.

AUTÓMATAS Y GRAMÁTICAS

- B^* es el conjunto de todas las cadenas de números, incluyendo ε , la cadena vacía.
- $D(LUD)^*$ es el conjunto de todas las cadenas de dígitos y letras que comienzan con un dígito.

Ejercicio 3

Dado el lenguaje $N = \{1, 2, 3\}$. Indique las cinco cadenas más cortas pertenecientes a N^* .

PARTE B: Expresiones Regulares

Definiciones

Expresiones regulares: representan patrones de cadenas de caracteres. Una expresión regular r se encuentra completamente definida mediante el conjunto de cadenas con las que concuerda.

Expresiones regulares básicas:

- Son los caracteres simples del alfabeto.
- Dado cualquier carácter a del alfabeto Σ , indicamos que la expresión regular a corresponde al carácter a escribiendo $L(a) = \{a\}$.
- El símbolo ε denota la cadena vacía, $L(\varepsilon) = \{\varepsilon\}$.
- El símbolo $\{\}$ corresponde a la ausencia de cadenas, cuyo lenguaje sea el conjunto vacío ϕ , $L(\phi) = \{\}$.
- El conjunto $\{\}$ no contiene ninguna cadena, y el conjunto $\{\varepsilon\}$ contiene la cadena simple que no se compone de ningún carácter.

Operaciones de expresiones regulares:

- **Selección entre alternativas:** se indica mediante el metacarácter $|$ (barra vertical). Si r y s son expresiones regulares, entonces $r | s$ es una expresión regular que define cualquier cadena que concuerda con r o con s .
- **Concatenación:** se indica mediante yuxtaposición, sin un metacarácter. La concatenación de r y s , rs , corresponde a cualquier cadena que sea la concatenación de dos cadenas, con la primera r y la segunda s .

AUTÓMATAS Y GRAMÁTICAS

- **Repetición:** se indica mediante el metacarácter *. La expresión regular r^* corresponde a cualquier concatenación finita de cadenas, cada una de las cuales corresponde a r .

Ejercicio 1

Escriba las descripciones para los lenguajes generados por las siguientes expresiones regulares:

1. $(ba|b)^*(a|bb)^*$
2. $(0|1|..|9|A|B|C|D|E|F)(w|W)$
3. $(A|B|...|Z)(a|b|...z)^*$

Ejemplo: dada la siguiente expresión regular $(a|b)^*a(a|b|\epsilon)$, el lenguaje generado por la misma es: Todas las cadenas de letras minúsculas que comienzan con una o más a o una o con una o más b.

Ejercicio 2

Escriba las expresiones regulares para validar:

1. Seguridad de una contraseña, que incluya:
 - a. Que contengan al menos una letra mayúscula.
 - b. Que contengan al menos una letra minúscula.
 - c. Que contengan al menos un número o carácter especial.
 - d. Cuya longitud sea como mínimo 8 caracteres.
 - a. Cuya longitud máxima no debe ser arbitrariamente limitada.
2. Cuenta de twitter
3. URL
4. Email
5. ID de un vídeo de Youtube
6. Fecha con formato dd/mm/yyyy o dd-mm-yyyy

AUTÓMATAS Y GRAMÁTICAS

7. Número de teléfono móvil de Argentina, que incluya código de país, de provincia, y el 15.
8. Código postal.
9. Número de teléfono fijo, que incluya código de país, de provincia.

Ejercicio 3:

Utilizando el lenguaje de programación Python, implementar la validación de las expresiones regulares del ejercicio anterior (Ejercicio 2)

Ejemplo: validación de extensiones para imágenes.

```
import re

regex = re.compile(r"jpg|png|gif|bmp|svg")

img_ext = input("Ingrese una extensión de una imagen: ")

if regex.match(img_ext):

    print('La extensión ', img_ext, 'se corresponde con la extensión de una imagen')

else:

    print('La extensión ', img_ext, 'no se corresponde con la extensión de una imagen')
```