
Grupo F: Bujaldon Juan Pablo, Herrero Tomas

AUTÓMATAS Y GRAMÁTICAS

TRABAJO PRÁCTICO N° 1

EXPRESIONES REGULARES

CONTENIDOS CONCEPTUALES

Expresiones regulares, lenguajes generados por las expresiones regulares, operaciones con expresiones regulares.

MODALIDAD DE TRABAJO

Trabajo de manera grupal, resolución de expresiones regulares y conjuntos regulares, utilización de expresiones regulares para validación de patrones.

OBJETIVOS DEL TP

- Comprender el concepto de expresiones regulares y su relación con los lenguajes formales.
- Aplicar expresiones regulares para la validación y búsqueda de patrones en cadenas de texto.
- Utilizar herramientas y lenguajes de programación para implementar y probar expresiones regulares.

RESULTADO/S DE APRENDIZAJE A LOS CUALES APORTA:

Al finalizar el trabajo práctico, el estudiante será capaz de explicar la sintaxis y semántica de las expresiones regulares, así como construir expresiones que representen distintos lenguajes formales. Además, podrá implementarlas en herramientas o lenguajes de programación como Python, evaluando su eficiencia y corrección en la resolución de problemas.

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

EJERCICIO N° 1

Dados los conjuntos $\alpha=\{e\}$, $\beta=\{a,b,c\}$, $\delta=\{c,d,e\}$ y $\lambda=\{\epsilon\}$, obtener los conjuntos resultantes:

- a) $\lambda \cup \delta = \{\epsilon, c, d, e\}$
- b) $\delta \cup \alpha = \{c, d, e\}$
- c) $\beta \lambda = \{a, b, c\}$
- d) $\beta \cup \delta = \{a, b, c, d, e\}$
- e) $\delta \beta = \{ca, cb, cc, da, db, dc, ea, eb, ec\}$
- f) $\beta \alpha \delta = \{aec, aed, aee, bec, bed, bee, cec, ced, cee\}$

EJERCICIO N° 2

Dados los conjuntos $\rho = \{0-3\}$ y $\chi = \{H-J\}$, obtener los conjuntos

- resultantes: a) $\rho \cup \chi$
- b) $\rho \chi$
- c) ρ^*
- d) $\rho +$

Respuestas:

- a) $\{0, 1, 2, 3, H, I, J\}$
- b) $\{0H, 1H, 2H, 3H, 0I, 1I, 2I, 3I, 0J, 1J, 2J, 3J\}$
- c) $\{\epsilon, 0, 1, 2, 3, 00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33, \dots\}$
- d) $\{0, 1, 2, 3, 00, 01, \dots\}$

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

EJERCICIO N° 3

Escriba una expresión regular que representa el siguiente lenguaje “Todas las palabras que comienzan con una a y que es seguida por 2 ó 3 b”. En caso de no ser posible justifique por qué.

{ abb(b) } o también podría ser { a(bb | bbb) }

EJERCICIO N° 4

Escriba una expresión para los siguientes conjuntos de caracteres:

- Todas las palabras de letras minúsculas que comiencen con la letra “hr”.
- Todas las cadenas de dígitos que no comiencen con el número “3”.
-

Escriba una expresión regular que contenga todas las cadenas que cumplan con las siguientes condiciones:

- Sus dos primeros caracteres son: “b” y “n”, en ese orden.
- Sus dos últimos caracteres son: “o” y “p”, en ese orden.

Respuestas:

- a) $hr(a-z)^*$
- b) $[0-24-9][0-9]^*$
- c) $bn[a-z]^*op$

EJERCICIO N° 5

Sea $L=\{a,b\}^*$ ¿Cuántas palabras de longitud 3 tiene este lenguajes?
¿Y de longitud 4? Escribálas.

Longitud 3: $2^3 = 8$ palabras

Longitud 4: $2^4 = 16$ palabras

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

EJERCICIO N° 6

Consideremos las cadenas en el alfabeto $\Sigma = \{a, b, c\}$, escriba la expresión regular que no contenga dos b consecutivas. En caso de no ser posible justifique por qué.

$(a|c)^*(b(a|c)^*)^*$

$(a|c)^*$ → Cualquier secuencia de 'a' y 'c' antes de una posible 'b'.

$b(a|c)^*$ → Si aparece una 'b', debe ir seguida solo por 'a' o 'c'.

$(b(a|c)^*)^*$ → Permite múltiples 'b', asegurando que cada una esté separada por al menos una 'a' o 'c'.

EJERCICIO N° 7

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

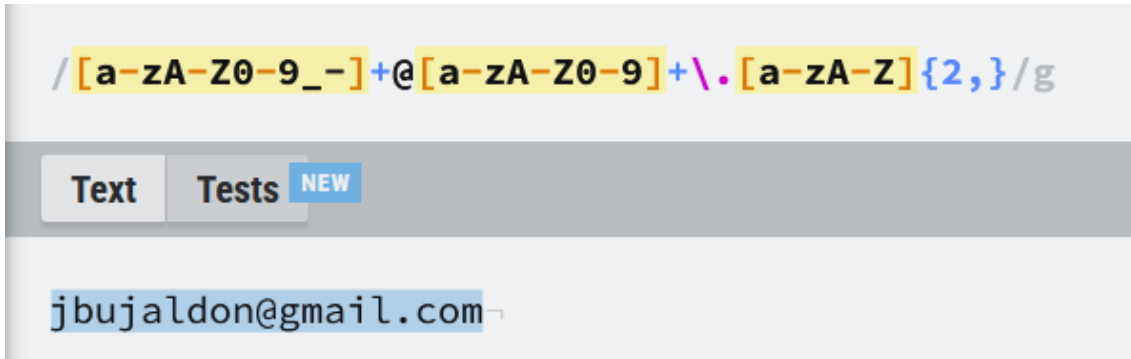
- $a(a|b)^*a$ = empieza con "a", puede contener cualquier combinación de "a" o "b" en el medio y debe terminar con "a"
- $(a|b)^*a(a|b|e)$ = puede empezar con cualquier combinación de "a" y "b", le sigue una "a" y puede contener una "a", "b" al final.
- $(A|B|...|Z)(a|b|...|z)^*$ = la primer letra tiene que ser mayúscula y seguido puede haber una minúscula o nada.

EJERCICIO N° 8

Escriba una expresión regular que valide una dirección de mail. El nombre de usuario puede contener letras mayúsculas, minúsculas, números y solo los caracteres especiales guion medio y bajo. El dominio puede contener solo letras mayúsculas, minúsculas y números. Luego validarla y presentar un ejemplo.

$[a-zA-Z0-9_-]^+@[a-zA-Z0-9]+\.[a-zA-Z]{2,}$

Grupo F: Bujaldon Juan Pablo, Herrero Tomas



EJERCICIO N° 9

Escriba una expresión regular que valide una URL.

`(https?:\V)(([a-zA-Z0-9-]+\.)+[a-zA-Z]{2,6})(\V[a-zA-Z0-9\-.~:\/?#\[\]@!$&'()*+,;=]*)?`

EJERCICIO N° 10

Construye una expresión regular que valide fechas en formato DD/MM/AAAA, asegurando que los días sean entre 01 y 31, los meses entre 01 y 12, y los años de cuatro dígitos.

`(0[1-9]|[12][0-9]|3[01])\V(0[1-9]|1[0-2])\V\d{4}`

PARTE 2 - EJERCICIOS PYTHON

EJERCICIO N° 1

Escribir una función en Python llamada `validate_string` que espere un único argumento de tipo string. La función deberá validar si el string cumple las siguientes condiciones:

- Contiene al menos un carácter alfanumérico.

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

- Contiene al menos una letra.
- Contiene al menos una letra mayúscula.
- Contiene al menos una letra minúscula.
- Contiene al menos un dígito.
- Contiene 8 o más caracteres.

El output esperado de la función es un boolean (True o False) por cada condición que se evalúa, en el orden especificado en este ejercicio.

Por ejemplo:

Dado el string xYz8, el output esperado es:

```
```\nTrue\nTrue\nTrue\nTrue\nFalse\n`\n`\n`
```

Dado el string xy@z!, el output esperado es:

```
```\nTrue\nTrue\nFalse\nTrue\nFalse\nFalse\n```\n
```

Se pueden utilizar funciones auxiliares si se deseara, pero recordar que el resultado se evaluará llamando a la función `validate_string`

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

EJERCICIO N° 2

Escribir una función en Python llamada solve que espere un único argumento de tipo string. Este string representará una operación matemática entre 2 o más números. Las posibles operaciones son suma (+) y multiplicación (*).

Se espera que la función resuelva la operación y devuelva el resultado como entero (int).

Por ejemplo:

Dado el string "2 + 7 * 2 + 1", Se espera que la función devuelva 17

Dado el string "2 * 2 * 2 + 32 * 2", Se espera que la función devuelva 72

Tener en cuenta que el string de entrada siempre será válido, no contendrá paréntesis y los números serán siempre enteros y positivos.

Recordar hacer return del valor correspondiente. Es opcional imprimirlo en pantalla

EJERCICIO N° 3

Utilizando el lenguaje de programación Python, construir un analizador para cada una de las siguientes expresiones, el analizador debe leer su entrada de un archivo de texto existente. El archivo de

Grupo F: Bujaldon Juan Pablo, Herrero Tomas

texto contendrá varias líneas, donde cada una representa una cadena a evaluar de manera individual.

Se espera que el programa lea el archivo correspondiente a cada caso y analice línea por línea, si la cadena es reconocida como correcta o no. Los archivos de texto correspondientes a cada ejercicio deben estar commiteados en el repositorio, y deben incluir al menos 5 cadenas cada uno.

Email (nombre@dominio.com) (el nombre debe comenzar con letra y puede tener números, _, puntos y -) (definir 5 dominios y 5 países)

a) URL (http-https://www.....com) (puede o no figurar el protocolo, y el www) puede terminar con la / o ? (query param "?clave=valor&clave=valor ")

b) Dirección IPv4 (000.000.000.000-255.255.255.255)

c) Contar la cantidad de palabras y determinar cuál es la palabra más repetida (y la cantidad de veces que aparece) del texto asignado a cada grupo.