



**Universidad Autónoma de Chiapas UNACH**

**Facultad:**

Contaduría y Administración, Campus I



**Licenciatura:**

Ingeniería en Desarrollo y Tecnologías de Software

**Unidad de Aprendizaje:**

Compiladores

**Docente:**

D.s.c. Luis Gutiérrez Alfaro

**Alumno:**

Lozano Monjarás David José

**Grado y Grupo:**

6° "M"

**Matricula:**

A210116

**Actividad:**

Define los siguientes conceptos y realizar los ejercicios.- Actividad I, II.- 5%

**Tuxtla Gutiérrez, Chiapas**

**Fecha: 25 – Enero – 2024**

## **Contenido**

<b>1. Definir el concepto de expresión regular.....</b>	<b>3</b>
<b>2. I.- Explicar los tipos de operadores de expresiones regulares. ....</b>	<b>3</b>
<b>Caracteres Especiales: Los caracteres especiales son metacaracteres que tienen significados especiales en las expresiones regulares. Algunos ejemplos comunes son:.....</b>	
	<b>3</b>
<b>Cuantificadores Los cuantificadores definen cuántas veces debe aparecer un carácter o un grupo en la cadena. Algunos ejemplos son: .....</b>	
	<b>3</b>
<b>3. II.- Explicar el proceso de conversión de DFA a expresiones regulares. ....</b>	<b>4</b>
<b>4. III.- Explicar leyes algebraicas de expresiones regulares. ....</b>	<b>4</b>
<b>5. Bibliografía formato APA.....</b>	<b>5</b>

## **1. Definir el concepto de expresión regular.**

Una expresión regular (también conocida como regex o regexp) es una secuencia de caracteres que define un patrón de búsqueda. Se utilizan para buscar, extraer o manipular texto, basándose en ciertos patrones predefinidos. Las expresiones regulares son herramientas poderosas y flexibles que se utilizan en procesamiento de texto, manipulación de cadenas, validación de entradas y otras tareas relacionadas con el manejo de patrones en texto.

En una expresión regular, puedes incluir caracteres literales y caracteres especiales que representan clases de caracteres, repeticiones, opciones y otros patrones.

## **2. Explicar los tipos de operadores de expresiones regulares.**

- La coincidencia literal: es el tipo más básico de expresión regular. Se compone de caracteres literales que deben coincidir exactamente con la cadena de texto. Por ejemplo, la expresión regular "hola" coincidiría solo con la cadena "hola".
- Operadores regulares comunes: Concatenación (sin operador específico): Ejemplo: 'A|B' coincide con cualquier cadena que tenga primero la letra 'A' y luego la letra 'B'.
- Caracteres Especiales: Los caracteres especiales son metacaracteres que tienen significados especiales en las expresiones regulares. Algunos ejemplos comunes son:
  - . (punto): Este símbolo coincide con cualquier carácter excepto el salto de línea.
  - \* (asterisco): En este caso, coincide con cero o más repeticiones del carácter anterior.
  - + (más): Coincide con una o más repeticiones del carácter anterior.
  - ? (signo de interrogación): Coincide con cero o una repetición del carácter anterior.
  - [] (corchetes): Define un conjunto de caracteres posibles para una posición en la cadena.
  - \ (barra invertida): Escapa un carácter especial para tratarlo como literal.

**Cuantificadores** Los cuantificadores definen cuántas veces debe aparecer un carácter o un grupo en la cadena. Algunos ejemplos son:

- {n}: Coincide con exactamente n repeticiones.
- {n, m}: Coincide con al menos n y como máximo m repeticiones.

**Operador de rango (-):** Se usa dentro de clases de caracteres para definir rangos.

Ejemplo: [A-Z] coincide con cualquier letra mayúscula.

Anclajes (^ y \$): ^ coincide con el inicio de una cadena. \$ coincide con el final de una cadena.

### 3. Explicar el proceso de conversión de DFA a expresiones regulares.

Supongamos que tenemos un DFA con un conjunto finito de estados  $Q$ , un conjunto de símbolos de entrada  $\Sigma$ , una función de transición  $\delta$ , un estado inicial  $q_0$ , un conjunto de estados de aceptación  $F$  y un conjunto de transiciones  $\epsilon$  (transiciones con el símbolo vacío).

- Eliminar estados inalcanzables: Identifica y elimina cualquier estado que no sea alcanzable desde el estado inicial.
- Eliminar estados no alcanzables: Identifica y elimina cualquier estado que no pueda alcanzar un estado de aceptación.
- Manejar transiciones con el símbolo vacío ( $\epsilon$ -transiciones): Utiliza técnicas para eliminar las transiciones con el símbolo vacío. Puedes aplicar el algoritmo de eliminación de  $\epsilon$ -transiciones.
- Asignar expresiones regulares a las transiciones: Asigna una expresión regular a cada transición entre estados del DFA. Puedes hacer esto de manera recursiva, utilizando las expresiones regulares de los estados alcanzados.
- Formar un conjunto de ecuaciones: Crea un conjunto de ecuaciones que representen las expresiones regulares para cada par de estados. Estas ecuaciones se derivan de las transiciones del DFA.
- Resolver las ecuaciones: Resuelve las ecuaciones del conjunto utilizando técnicas algebraicas para obtener las expresiones regulares definitivas para cada par de estados.
- Expresar la expresión regular para el estado inicial y final: Utiliza la expresión regular obtenida para el estado inicial y final para representar la expresión regular que describe todo el lenguaje aceptado por el DFA.

### 4. Explicar leyes algebraicas de expresiones regulares.

Las leyes algebraicas de expresiones regulares son reglas y propiedades que rigen la manipulación algebraica de expresiones regulares. Estas leyes facilitan la simplificación y manipulación de expresiones regulares, y son útiles en la teoría de lenguajes formales y autómatas.

**Ley de la Identidad (Idempotencia):**  $R + \emptyset = R$   $R \cdot \emptyset = \emptyset$

- La unión con el conjunto vacío ( $\emptyset$ ) no afecta la expresión regular  $R$ .
- La concatenación con el conjunto vacío ( $\emptyset$ ) siempre resulta en el conjunto vacío.

Ley del Elemento Neutro:  $R + \epsilon = R$   $\epsilon \cdot R = R$

La unión con el símbolo vacío ( $\epsilon$ ) no afecta la expresión regular  $R$ .

La concatenación con el símbolo vacío ( $\epsilon$ ) no afecta la expresión regular  $R$ .

**Ley de Anulación:**  $R \cdot \emptyset = \emptyset$   $\emptyset \cdot R = \emptyset$

- La concatenación con el conjunto vacío ( $\emptyset$ ) siempre resulta en el conjunto vacío.

**Ley de De Morgan:**  $(R + S)' = R' \cdot S'$   $(R \cdot S)' = R' + S'$

- La negación de la unión de dos expresiones regulares es equivalente a la concatenación de sus negaciones.
- La negación de la concatenación de dos expresiones regulares es equivalente a la unión de sus negaciones.

Estas leyes proporcionan reglas algebraicas que permiten simplificar y manipular expresiones regulares de manera eficiente. Al aplicar estas leyes de manera sistemática, se pueden reducir expresiones regulares a formas más simples y equivalentes.

## 5. Bibliografía formato APA

IBM Documentation. (2023, octubre 10). lbm.com.

<https://www.ibm.com/docs/es/i/7.3?topic=expressions-regular>

de Formación Profesional Superior, E. C. (2023, agosto 29). *¿Qué es una expresión regular y qué tipos existen?* Esic.edu; ESIC.

<https://www.esic.edu/rethink/tecnologia/que-es-una-expresion-regular-que-tipos-existen-c>

*Expresiones Regulares.* (s/f). MDN Web Docs. Recuperado el 28 de enero de 2024,

de

[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular\\_expressions](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_expressions)

(S/f). Uji.es. Recuperado el 28 de enero de 2024, de  
<https://repositori.uji.es/xmlui/bitstream/handle/10234/5875/bolAuto4.pdf?sequence=5>

*DFA A Expresion Regular*. (s/f). Scribd. Recuperado el 28 de enero de 2024, de  
<https://es.scribd.com/doc/12929632/DFA-a-Expresion-Regular>

(S/f). Edu.ar. Recuperado el 28 de enero de 2024, de  
<https://wiki.cs.famaf.unc.edu.ar/lib/exe/fetch.php?media=intrologica:2015:clas-3-handout-2015.pdf>