



UNIVERSIDAD AUTÓNOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN CAMPUS 1

ACTIVIDAD *Investigación*.

ING. EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

ALUMNO: *Gabriel Omar Fuentes Chacon*.

DOCENTE: Mtro. Luis Gutiérrez Alfaro

Fecha de entrega: *12/08/2023*





Definición de una expresión regular:

Una expresión regular es una cadena que es utilizada para describir o encontrar patrones dentro de otros strings. En base del uso de delimitadores y ciertas reglas de sintaxis.

Estas se componen por caracteres literales y meta caracteres que representan clases de caracteres, repeticiones, alternancias y otros elementos que definen el patrón de búsqueda.

Las expresiones regulares son herramientas flexibles utilizadas en programación, procedimientos de texto, análisis de datos y muchas áreas.

Explicar los tipos de operadores de expresiones regulares.

- 1 **.**: Punto coincide con cualquier carácter excepto un salto de línea.
- 2 **\***: Asterisco coincide con cero o más repeticiones del elemento anterior
- 3 **+**: mas coincide con una o más repeticiones del elemento anterior
- 4 **?**: Interrogación coincide con cero o una repetición del elemento anterior o lo hace opcional
- 5 **^**: Circunflejo: Coincide con el final de una línea o una cadena.
- 6 **\$**: Peso coincide con el final de una línea o cadena.

7 `[]` (Corchetes): Define una clase de caracteres, coincide con cualquier caracter dentro de la clase.

8 `[]^`: Corchetes negados. Coinciden con cualquier caracter que no este en clase.

9 `|`: Barra vertical actua como un operador OR, coincide con una expresion o la otra.

10 `()`: Parantesis agrupa los elementos juntos y captura subexpresiones.

11 `\`: Barra invertida Escapan un caracter especial para que sea interpretado literalmente.

12 `\d`: Coinciden con un dígito del (0-9)

13 `\w`: Coinciden con un caracter alfanumerico o letra o dígito.

14 `\s`: Coincide con un espacio en blanco (espacio, tabulacion, salto de linea, etc)

15 `\b`: Coincide con un limite de palabra (inicio o fin de una palabra).

16 `\B`: Coincide con cualquier posicion que no sea un limite de palabra.

17 `n`: Coincide con "n" repeticiones del elemento anterior.

18 `n,m`: Coincide con al menos "n" y como maximo "m" repeticiones del elemento anterior.



## Proceso de conversión de DFA a expresiones regulares

1. **Identificar el proceso inicial y los estados finales**  
Comienza identificando los estados iniciales y los estados finales del DFA, siendo elementos claves en los elementos y en la creación de la expresión regular resultante.
2. **Agregar un nuevo estado** Agrega un nuevo estado al DFA y conecta todas las transiciones de estados iniciales a este nuevo estado.
3. **Eliminación de estados**: para cada par de estados " $q$ " y " $p$ " en el DFA original intenta encontrar una secuencia de estados intermedios que conecten " $q$ " con " $p$ " a través del nuevo estado agregado en el paso anterior.
4. **Expresiones regulares en transición**: Después de eliminar todos los estados excepto la " $p$ " el estado inicial y los estados finales, las transiciones restantes en el "DFA" deben representar directamente las partes de la expresión regular que estamos buscando.
5. **Resolver las expresiones regulares**: Apartir de las transiciones y estados restantes, construye la expresión regular para cada transición, pudiendo hacer esto iterativamente resolviendo las expresiones para cada estado.
6. **Expresión regular final**: Una vez que hayas calculado las expresiones regulares para todas las transiciones y ajustado las ocurrencias de estados intermedios, deberías de obtener una única expresión regular que represente el lenguaje aceptado por el DFA original.

8 **Lex de idempotencia:** La union de una expresion regular con ella misma es igual a la expresion regular original es decir  $A \cup A$  es igual a  $A$

9 **Lex de complemento de cierre:** El complemento de cierre de una expresion regular  $A^*$  es igual a  $(A)^*$

10 **Lex de cierre Opcional:**  $A?$  es equivalente a  $A \cup \epsilon$

Ejemplo

La expresion regular E.R del lenguaje cuyas palabras son  $V = (a, b)$  y terminan B son:

$$(a + b)^* b$$

La expresion regular cuyas palabras estan formadas por letras del alfabeto  $V = (a, b)$  que tienen longitud mayor a 2 la segunda letra es a y la antenultima e b

$$(a + b) a (a + b)^* b (a + b)$$

### ¿Que es un token?

Es una expresion regular en una secuencia de caracteres que actua como una unidad individual dentro del patron, puede representarse un elemento en particular en una cadena estas tratando de coincidir o analizar

### ¿Que es un lexema?

Se conoce como secuencia de especificas de caracteres en codigo fuente que empareja con reglas definidas por expresiones regulares. Los tokens seria categoria abstracta numero entero. Los Lexemas son secuencias especificas de caracteres que coinciden con el patron en la cadena.

### ¿Que es un patron?

Un patron es el contexto de expresiones regulares es una descripcion formal de una regla que define secuencias de caracteres. Siendo la plantilla con la cual se compara las cadenas de texto y se utilizan para buscar, extraer, remplazar, manipular, etc.