

## ***I.- Explicar los tipos de operadores de expresiones regulares.***

### **1. Operadores de concordancia:**

- ``literal``: encuentra una coincidencia exacta del texto especificado.
- ``.` (punto): encuentra cualquier carácter, excepto un salto de línea.
- ``\`` (barra invertida): anula el significado especial de los caracteres especiales.

### **2. Operadores de cantidad:**

- ``*`` (asterisco): encuentra cero o más repeticiones del carácter o grupo anterior.
- ``+`` (más): encuentra una o más repeticiones del carácter o grupo anterior.
- ``?`` (signo de interrogación): encuentra cero o una repetición del carácter o grupo anterior.
- ``{n}``: encuentra exactamente *n* repeticiones del carácter o grupo anterior.
- ``{n,}``: encuentra al menos *n* repeticiones del carácter o grupo anterior.
- ``{n, m}``: encuentra al menos *n* y como máximo *m* repeticiones del carácter o grupo anterior.

### **3. Operadores de clase de caracteres:**

- ``[]`` (corchetes): encuentra cualquier carácter especificado dentro de los corchetes.
- ``[^ ]`` (corchetes negados): encuentra cualquier carácter que NO esté especificado dentro de los corchetes.
- ``-`` (guion): define un rango de caracteres dentro de los corchetes.

### **4. Operadores de agrupación y alternancia:**

- ``()`` (paréntesis): agrupa caracteres o sobrexpresiones en una sola unidad.
- ``|`` (barra vertical): permite alternar entre múltiples opciones de coincidencia.

### **5. Operadores de anclaje:**

- ``^`` (circunflejo): coincide con el inicio de una línea o cadena.
- ``$`` (signo de dólar): coincide con el final de una línea o cadena.
- ``\b``: coincide con los límites de una palabra.
- ``\B``: coincide con cualquier posición que no sea límite de palabra.

## **II.- Explicar el proceso de conversión de DFA a expresiones regulares.**

Básicamente este método consiste en seleccionar tres estados:  $q_r$ , el cual no deberá ser ni el estado inicial, ni ninguno de los estados finales o de aceptación, también se deberá seleccionar un estado  $q_x$  y  $q_y$ , de manera que  $q_x$  pueda llegar (por medio de transiciones) a  $q_y$  utilizando a  $q$ , como estado intermedio entre estos. Después de haber seleccionado estos estados, se debe proceder a eliminar el estado  $q$ , haciendo una transición que vaya de  $q_x$  a  $q_y$  y que por medio de la concatenación de las transiciones que llegan de  $q_x$  a  $q$ , y salen de  $q$ , a  $q_y$  (incluyendo las que hacen un bucle en  $q$ ). En caso de que ya exista una transición que va de  $q_x$  a  $q_y$ , se hace la unión de la Expresión Regular de dicha transición con la Expresión Regular de la nueva transición antes creada. Esto se repite hasta que solo existan estados iniciales y finales en el DFA. Luego de tener la máquina de esta forma se debe generar la Expresión Regular a partir de ella.

## **III.- Explicar leyes algebraicas de expresiones regulares**

Existen un conjunto de leyes algebraicas que se pueden utilizar para las expresiones regulares:

- Ley conmutativa para la union:  $L + M = M + L$
- Ley asociativa para la union:  $(L + M) + N = L + (M + N)$
- Ley asociativa para la concatenacion:  $(LM)N = L(MN)$

NOTA: La concatenación no es conmutativa, es decir

$$LM \neq ML$$

Una identidad para un operador es un valor tal que cuando el operador se aplica a la identidad y a algun otro valor, el resultado es el otro valor.

- 0 es la identidad para la adición:  $0 + x = x + 0 = x$ .
- 1 es la identidad para la multiplicación:

$1 \times x = x \times 1 = x$  Un aniquilador para un operador es un valor tal que cuando el operador se aplica al aniquilador y algun otro valor, el resultado es el aniquilador.

- 0 es el aniquilador para la multiplicación:

$$0 \times x = x \times 0 = 0$$

## Ejercicios

1. Realice una expresión regular de todas las cadenas con símbolos a y b, que terminan con el sufijo abb. Ejemplo de estas cadenas son:

abb, aabb, babb, aaabb, ababb, baabb, bbabb, ...

$$R=[ab]^*abb \quad R=(a|b)^*abb$$

2. Realice una expresión regular de todas las cadenas de con símbolos 0 y 1, que primero tengan los símbolos 1 's con longitud impar y después aparezcan los 0 's con longitud par. Ejemplo de estas cadenas son:

100, 10000, 1000000, 11100, 1110000, 111110000, ...

$$R=1(11)^*(00)^*00 \quad R=1(11)^*(00)^+$$

3. Para la expresión regular (+|-)?d + .d + indique las cadenas correctas de los siguientes incisos. (Nota. En esta expresión él es un símbolo no el operador concatenación y d representa los dígitos del 0 al 9).

a) -20.43

b) 0.3216

c) 329.

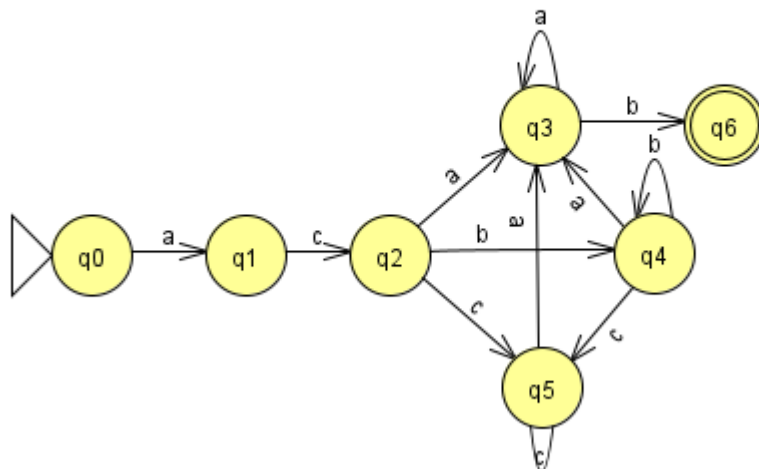
d) 217.92

e) +2019

f) +.762

g) -.4555

4.- Obtenga un AFD dado el siguiente lenguaje definido en el alfabeto  $\Sigma=\{a, b, c\}$ . El conjunto de cadenas que inician en la sub-cadena "ac" y terminan en la sub-cadena "ab".



5.- Obtenga un AFND dado el siguiente lenguaje definido en el alfabeto  $\Sigma=\{a, b, c\}$ . El conjunto de cadenas que no inician en la sub-cadena "ac" o no terminan en la sub-cadena "ab".

