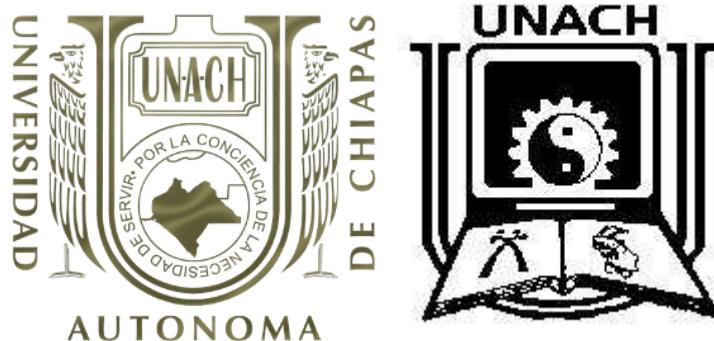


# **UNIVERSIDAD AUTÓNOMA DE CHIAPAS**



**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN  
CAMPUS 1**

**Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software**

Define los siguientes conceptos y realizar los ejercicios.-

Actividad I, II

6 “M”

Materia: Compiladores

Docente: Luis Gutiérrez Alfaro

**ALUMNO:  
A 211387**

**Steven de Dios Montoya Hernández**

**TUXTLA GUTIÉRREZ, CHIAPAS  
Domingo 13 de Agosto 2023**

Steven de Dios Montoya Hernández

TEMA

FECHA

08/08/2023

Definir el concepto de expresión regular.

Una expresión regular, es una expresión que describe un conjunto de cadenas sin enumerar sus elementos.

Ejemplo:

El grupo formado por las cadenas Handel, Händel y Haendel se describe mediante el patrón:

"H(Calälæ)ndel". La mayoría de las formalizaciones proporcionan estos constructores: Una expresión regular es una forma de representar si los lenguajes regulares (finitos o infinitos) y se construye utilizando caracteres del alfabeto sobre los cuales se define el lenguaje

Explicar los tipos de operadores de expresiones regulares.

1- Operadores de caracteres literales: Por ejemplo la expresión regular 'abc' buscará la secuencia de caracteres "abc"

2- Punto (.): Representa cualquier carácter excepto el salto de línea. Por ejemplo "a.b" coincidirá con "aab", "axb", "alb", etc pero no con "a\nb".

3- Asterisco (\*): Representa cero o más repeticiones del elemento anterior

4- Signo mas (+): Representa una o más repeticiones del elemento anterior. Por ejemplo, "ab+c" coincidirá con abc, abbc, "abbbc" pero no con ac

5- Interrogación (?): Representa cero o una repetición del elemento anterior. "Colou?r" coincide con "color" o "colour"

6.- Claves ( $\{n\}$  y  $\{n,m\}$ ): Representa una cantidad específica de repeticiones.  $\{n\}$  indica exactamente  $n$  repeticiones, mientras que  $\{n,m\}$  indica entre  $n$  y  $m$  repeticiones. Por ejemplo,  $a\{2\}$  coincide con "aa" y  $a\{2,4\}$  coincide con aa,aaa o aaaa.

7.  $[C]$ : Definen un conjunto de caracteres posibles para una posición en la expresión. Ejemplo  $[aeiou]$  coincide con cualquier vocal.

8.- Negación de clase de caracteres  $[^C]$ : Coincide con cualquier carácter que no esté en el conjunto definido. Por ejemplo  $[^0-9]$  coincide con cualquier carácter que no sea un dígito numérico.

9.- Carácter de escape (\): Se utiliza para tratar un carácter especial como un carácter literal. Por ejemplo,  $\backslash\backslash$ , coincide con el carácter punto literal.

10.- Andajes ( $^A$  y  $^{\$}$ ):  $^A$  representa el inicio de una línea y  $^{\$}$  representa el final de una línea. Por ejemplo, " $^A$ Inicio" coincide solo si "Inicio"

11.- Operadores de alternancia (|): Se utiliza para alternar entre dos o más patrones posibles.  $a|b = a \cup b$

12.- ( ): Se utiliza para agrupar partes de una expresión para aplicar operadores a ellas. Ejemplo  $(abc)^+$  coincide con "abc".

TEMA

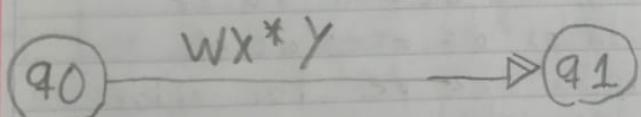
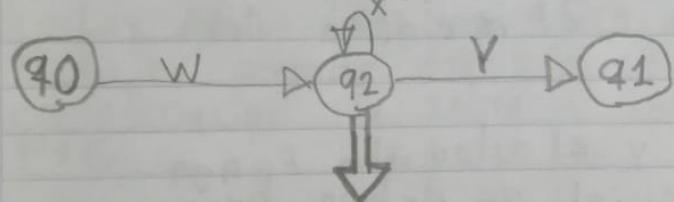
FECHA

Explicar el proceso de conversión de DFA a expresiones regulares.

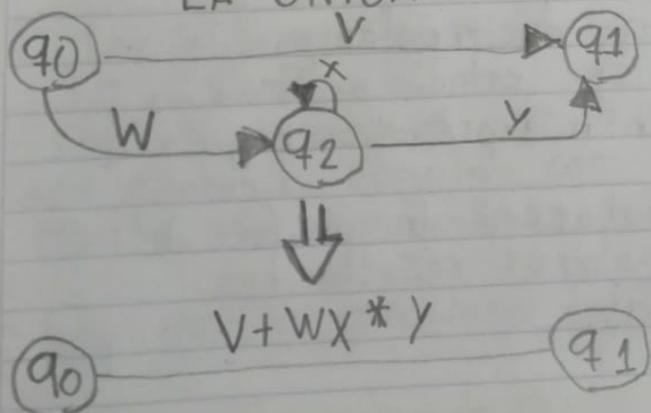
Uno de los métodos que se usan para transformar autómatas finitos deterministas en expresiones regulares, es el método de eliminación de estados.

Cuando eliminamos un estado, tenemos que remplazar todos los caminos que pasaban a través de él como transiciones directas que ahora se realizan con el ingreso de expresiones regulares, en vez de con símbolos.

Concatenación

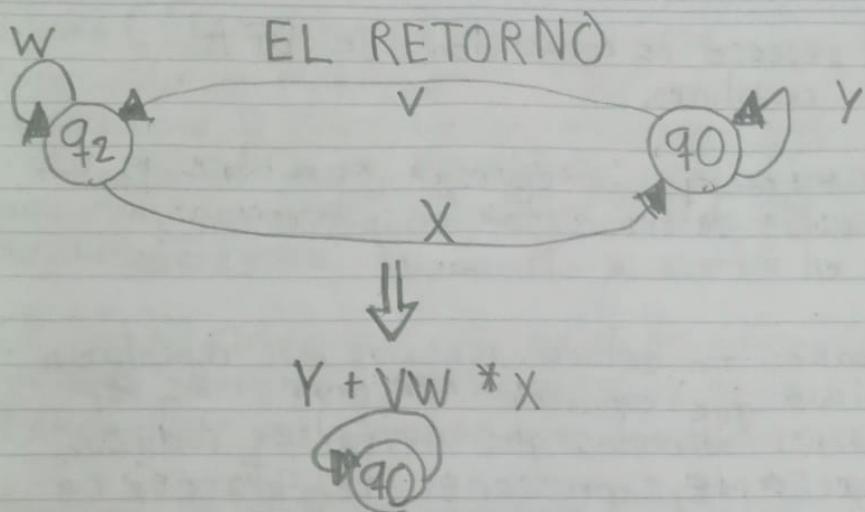


LA UNION



TEMA

FECHA



Se recomienda eliminar primero todos los estados que no sean ni el inicial ni las finales

Una vez eliminados y el automata tenga mas de un estado inicial, se deben hacer tantas copias como estados de aceptación tenga el automata. En cada una de las copias, se debe elegir uno de los estados de aceptación diferentes.

Todos los demás estados de aceptación de esta copia pasarán a ser estados ordinarios, los cuales se reducen a expresiones regulares

La expresión regular final será la unión de todas las expresiones regulares resultantes de cada una de las copias.

TEMA

FECHA

Un patrón explica lo que puede ser un token, y estos patrones se definen mediante expresiones regulares. En el lenguaje de programación, las palabras clave, las constantes, los identificadores, las cadenas, los números, los operadores y los símbolos de puntuación pueden considerarse tokens.

Los tokens en expresiones regulares son partes de texto que se capturan entre parentesis y que se usan para hacer coincidir otras partes del mismo.

Por ejemplo la expresión regular  $C.*\n$  tiene un token  $C.*$  que coincide con cualquier cadena de caracteres seguida de un salto de linea \n,

Un patrón en expresión regular es una secuencia de caracteres que se usa para buscar o reconocer otras cadenas de texto que coinciden con ese patrón. Ejemplo: /a\* b/ coincide con cualquier cadena que empiece con cero o más a y termine con una b.

Un lexema es una secuencia de caracteres en el programa fuente con lo que concuerda el patrón para un componente léxico. Regla que describe el conjunto de lexemas que puede representar a un determinado componente léxico en los programas fuente.

TEMA

FECHA

7- Ley de la cerradura de Kleene:  
 $E^* = E + E \cdot E^*$

Esta ley define la cerradura de Kleene de una expresión regular  $E$  como la unión de la cadena vacía y la concatenación de  $E$  con su cerradura de Kleene.

8- Ley del complemento:  $E + \bar{E} = 1$

Esta ley establece que la unión de una expresión regular con su complemento resulta en una expresión regular que acepta cualquier cadena posible.

TEMA

FECHA

Un patrón explica lo que puede ser un token, y estos patrones se definen mediante expresiones regulares. En el lenguaje de programación, las palabras clave, las constantes, los identificadores, las cadenas, los números, los operadores y los símbolos de puntuación pueden considerarse tokens.

Los tokens en expresiones regulares son partes de texto que se capturan entre parentesis y que se usan para hacer coincidir otras partes del mismo.

Por ejemplo la expresión regular  $C.*\n$  tiene un token  $C.*$  que coincide con cualquier cadena de caracteres seguida de un salto de linea \n,

Un patrón en expresión regular es una secuencia de caracteres que se usa para buscar o reconocer otras cadenas de texto que coinciden con ese patrón. Ejemplo: /a\* b/ coincide con cualquier cadena que empiece con cero o más a y termine con una b.

Un lexema es una secuencia de caracteres en el programa fuente con lo que concuerda el patrón para un componente léxico. Regla que describe el conjunto de lexemas que puede representar a un determinado componente léxico en los programas fuente.

**1.- Realice una expresión regular de todas las cadenas con símbolos a y b, que terminan con el sufijo abb, Ejemplo de éstas cadenas son: abb, aabb, babb, aaabb, ababb, baabb, bbabb,...**

**Expresión regular:**  $[a|b]^*abb$

**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,.....**

**Expresión Regular:**  $1(11)^*(00)^+$

**3.- Para la expresión regular  $(+|-)?d+d+d$  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.

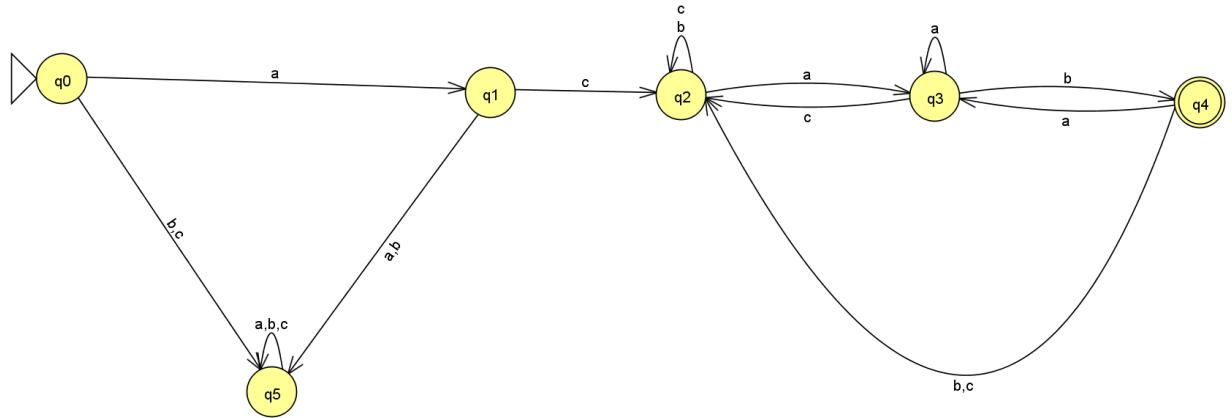
**d) 217.92**

e)+2019

f)+.762

g)-.4555

**4.- Obtenga a 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”.**

