



**UNIVERSIDAD AUTONOMA DE CHIAPAS
FACULTAD DE CONTADURIA Y ADMINISTRACION
CAMPUS I TUXTLA GUTIERREZ.**

**LICENCIATURA EN INGENIERÍA EN DESARROLLO Y
TECNOLOGÍAS DE SOFTWARE.**

ASIGNATURA:

COMPILADORES.

ACTIVIDAD:

ACT 1. INVESTIGACION Y EJEMPLOS

ALUMNO:

GARCIA MONTEJO JESUS ALBERTO.

DOCENTE:

DR. GUTIERREZ LUIS ALFARO.

SEMESTRE: 6°

GRUPO: N

LUGAR: TUXTLA GTZ

FECHA: 27/01/2024

EXPRESIÓN REGULAR.

En el ámbito de la programación, las expresiones regulares representan una herramienta sumamente poderosa y flexible para la búsqueda y manipulación eficiente de patrones de texto. En esencia, una expresión regular, también conocida como regex o regexp, es una secuencia de caracteres que define un patrón de búsqueda. Para los programadores y desarrolladores que trabajan con manipulación de texto, las expresiones regulares son esenciales, ya que permiten realizar diversas tareas como validar datos, buscar y reemplazar texto, así como extraer información específica de conjuntos de datos.

En términos más simples, una expresión regular se resume como una secuencia de caracteres que establece un patrón textual. Este patrón se utiliza para buscar coincidencias dentro de cadenas de caracteres. Para visualizarlas de manera más concreta, puedes concebir las expresiones regulares como "máscaras de búsqueda" que posibilitan la identificación de patrones específicos en un texto, sin importar cuán complejos o variados sean dichos patrones.

Ejemplo:

| Expresión regular | Descripción |
|---------------------------------|---|
| [xyz] | x, y, o z |
| (información advertencia error) | información, advertencia o error |
| [a-z] | Una letra minúscula |
| [^a-z] | No una letra minúscula |
| [a-z] + | Una o más letras minúsculas |
| [a-z] * | Cero o más letras minúsculas |
| [a-z] ? | Cero o una letra minúscula |
| [a-z] {3} | Exactamente tres letras minúsculas |
| [d] | Un dígito |
| \d+\$ | Uno o más dígitos seguidos por fin del mensaje |
| [0-5] | Un número de 0 a 5 |
| \w | Un carácter de palabra (letra, dígito o guion bajo) |
| \s | Espacio en blanco |
| \S | Cualquier carácter, excepto espacio en blanco |
| [a-zA-Z0-9] + | Uno o más caracteres alfanuméricos |
| ([a-z] {2,} [0-9] {3,5}) | Dos o más letras seguidas por tres a cinco números |

I.- EXPLICAR LOS TIPOS DE OPERADORES DE EXPRESIONES REGULARES.

Las expresiones regulares se pueden construir utilizando una variedad de operadores. Los operadores básicos son los siguientes:

| Operador | Descripción | Ejemplo | Devuelve |
|--------------|--|------------------|---|
| ^ | Coincide con el principio de una cadena | ^abc | abc, abcdef..., abc123 |
| \$ | Coincide con el final de una cadena | abc\$ | mi:abc, 123abc, theabc |
| . | Coincide con cualquier carácter como comodín | a.c | abc, asc, a123c |
| | Un carácter O | abc xyz | abc o xyz |
| (...) | Captura los valores entre paréntesis. | (a)b(c) | a y c |
| [...] | Coincide con todo lo que esté entre corchetes | [abc] | a, b, o c |
| [a-z] | Coincide con los caracteres en minúscula entre a y z | [b-z] | bc, mente, xyz |
| [0-9] | Coincide con cualquier valor numérico entre 0 y 9. | [0-3] | 3201 |
| {x} | El número exacto de veces que debe coincidir | (abc){2} | abcabc |
| {x,} | El número mínimo de veces que debe coincidir | (abc){2,} | abcabcabc |
| * | Coincide con cualquier cosa en lugar de *, o una coincidencia "codiciosa". | ab*c | abc, abbcc, abcdc |
| + | Coincide con el carácter anterior al + una o más veces | a+c | ac, aac, aaac |
| ? | Coincide con el carácter anterior a ? cero o una vez, o una coincidencia "no codiciosa". | ab?c | ac, abc |
| / | Escapa el carácter después de /, O crea una secuencia de escape | a/bc | a c, con el espacio correspondiente a /b |

II.- EXPLICAR EL PROCESO DE CONVERSIÓN DE DFA A EXPRESIONES REGULARES.

El proceso de conversión de un DFA a una expresión regular se puede dividir en los siguientes pasos:

Identifique los estados finales del DFA. Los estados finales representan las cadenas de texto que el DFA reconoce.

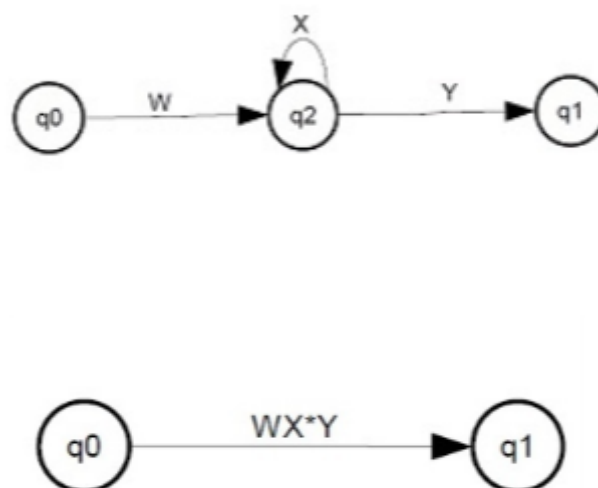
Asigne a cada estado no final una expresión regular. La expresión regular para un estado no final debe coincidir con todas las cadenas de texto que conducen a ese estado.

Combine las expresiones regulares de los estados no finales para obtener una expresión regular que coincida con todas las cadenas de texto que el DFA reconoce.

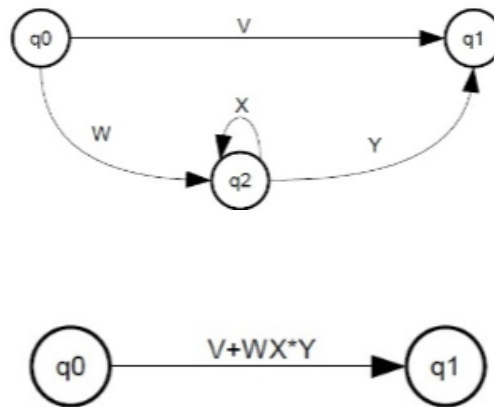
Uno de los métodos que se usan para transformar autómatas finitos deterministas en expresiones regulares es el método de eliminación de estados.

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

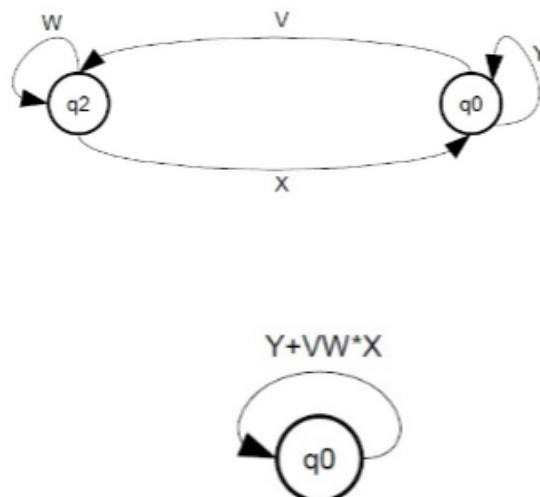
La concatenación.



La unión.



El retorno.



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

Una vez eliminados y el autómata tenga más de un estado inicial, se deben hacer tantas copias como estados de aceptación tenga el autómata. 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 pasaran 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.

III.- EXPLICAR LEYES ALGEBRAICAS DE EXPRESIONES REGULARES.

Conmutativas.

Se dice que un lenguaje L es conmutativo si se cumple que un operador pueda cambiar el orden de sus operadores y aun así obtener el mismo resultado.

$L+M = M+L$. Esta ley, la ley conmutativa de la unión, establece que podemos efectuar la unión de dos lenguajes en cualquier orden.

Asociativas.

El asociativo es la propiedad de un operador que nos permite reagrupar los operando cuando el operador se aplica dos veces.

$(L+M)+N = L+(M+N)$. Esta ley, la ley asociativa para la unión, establece que podemos efectuar la unión de tres lenguajes bien calculando primero la unión de los dos primeros, o bien la unión de los dos últimos.

Elemento Identidad.

El elemento identidad de un operador es un valor que operado con cualquier otro número no lo altera.

Ejemplo: 0 es el elemento identidad para la suma, ya que $0+X = X+0 = X$, Y 1 es el elemento identidad de la multiplicación, puesto que $1 \times X = X \times 1 = X$.

$0+L=L+0= L$. Esta ley establece que 0 es el elemento identidad para la unión.

Elemento Nulo.

Es un valor tal que cuando el operador se aplica al propio elemento nulo y a algún otro valor, el resultado es el elemento nulo.

Ejemplo: 0 es el elemento nulo de la multiplicación, ya que $0 \times x = x \times 0 = 0$.

Leyes distributivas.

Esta implica a dos operadores y establece que un operador puede aplicarse por separado a cada argumento del otro operador. Existe una ley análoga para las expresiones regulares, que tenemos que establecer de dos formas

$L(M + N) = LM + LN$. Ésta es la ley distributiva por la izquierda de la concatenación respecto de la unión.

$(M + N)L = ML + NL$. Ésta es la ley distributiva por la derecha de la concatenación respecto de la unión.

Ley de idempotencia.

Se dice que un operador es idempotente si el resultado de aplicarlo a dos valores iguales es dicho valor. Los operadores aritméticos habituales no son idempotentes.

$L + L = L$. Ésta es la ley de idempotencia para la unión, que establece que si tomamos la unión de dos expresiones idénticas, podemos reemplazarla por una copia de la de la expresión.

Las leyes algebraicas de las expresiones regulares son un conjunto de reglas que establecen la equivalencia entre diferentes formas de expresiones regulares.

Algunas de estas leyes son:

- Ley conmutativa para la unión: $E + F = F + E$
- Ley asociativa para la unión: $(E + F) + G = E + (F + G)$
- Ley asociativa para la unión: $(E + F) + G = E + (F + G)$
- Ley de absorción: $E + (EF) = E$
- Ley de De Morgan: $(E + F)^* = (E^*F^*)^*$

Estas leyes son útiles para manipular y transformar expresiones regulares, lo que puede ser beneficioso en el contexto de la teoría de autómatas y la construcción de lenguajes formales.

BIBLIOGRAFIA.

de Formación Profesional Superior, E. C. (2023, August 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>

Ejemplos de expresiones regulares. (n.d.). VMware.com. Retrieved January 27, 2024, from <https://docs.vmware.com/es/vRealize-Log-Insight/8.10/com.vmware.log-insight.user.doc/GUID-88B2952D-3112-46BC-B126-84C9BF38B6D2.html>

(N.d.). Workiva.com. Retrieved January 27, 2024, from <https://support.workiva.com/hc/es-419/articles/4407304269204-Operadores-de-expresiones-regulares>

TRANSFORMAR UN DFA EN UNA EXPRESION REGULAR. (n.d.). prezi.com. Retrieved January 27, 2024, from <https://prezi.com/fzrkq7tpeo9k/transformar-un-dfa-en-una-expresion-regular/>

Algebra de las expresiones regulares. (n.d.). Autómatas Wiki; Fandom, Inc. Retrieved January 27, 2024, from https://automatas.fandom.com/es/wiki/Algebra_de_las_expresiones_regulares

(N.d.). Edu.Ar. Retrieved January 27, 2024, from <https://wiki.cs.famaf.unc.edu.ar/lib/exe/fetch.php?media=intrologica%3A2015%3Aclass-3-handout-2015.pdf>