UNIVERSIDAD AUTONOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN, C-I

LIC.ING.EN DESARROLLO Y TECNOLOGIAS DE SOFTWARE

ACTIVIDAD 1

PROFESOR:
LUIS GUTIERREZ ALFARO

ALUMNA:
RAMIREZ SANCHEZ ADELENE SARAI-A210329

MATERIA:
COMPILADORES

SEMESTRE Y GRUPO: 6°M

28-ENERO-2024

Índice

Índice *1*

Introducción *2*

Definir los siguientes Conceptos y de ejemplo de cada uno de los Incicios de I, II, III **3**

I.- Explicar los tipos de operadores de expresiones regulares......................... 3

II.- Explicar el proceso de conversión de DFA a expresiones regulares 4

Anexo.. 4

III.- Explicar leyes algebraicas de expresiones regulares 7

Conclusión *8*

Bibliografía *9*

Introducción

Estos temas son interesantes porque no solo es matemática también trata de usar la lógica para poder realizar cualquier ejercicio que nos propongan, en esta investigación aprendí algunos conceptos, aunque en los ejercicios se me hacen aún un poco difícil porque a veces no logro encontrar la lógica.

Si hablamos de expresión regular hay muchos que lo definen como un método flexible y eficaz para procesar textos. Por lo cual te dejo esta información sacada de fuentes de internet esperando que sea buena información y clara.

Definir los siguientes Conceptos y de ejemplo de cada uno de los Incicios de I, II, III

Las expresiones regulares proporcionan un método eficaz y flexible para procesar texto. La notación extensa de coincidencia de patrones de expresiones regulares permite analizar rápidamente grandes cantidades de texto para lo siguiente:

- Buscar patrones concretos de caracteres.
- Validar el texto para garantizar que coincide con un patrón predefinido (como una dirección de correo electrónico).
- Extraer, editar, reemplazar o eliminar subcadenas de texto.
- Agregar cadenas extraídas en una colección para generar un informe.

Para muchas aplicaciones que usan cadenas o analizan grandes bloques de texto, las expresiones regulares son una herramienta indispensable. (Learn.Microsoft, s.f.)

I.- Explicar los tipos de operadores de expresiones regulares

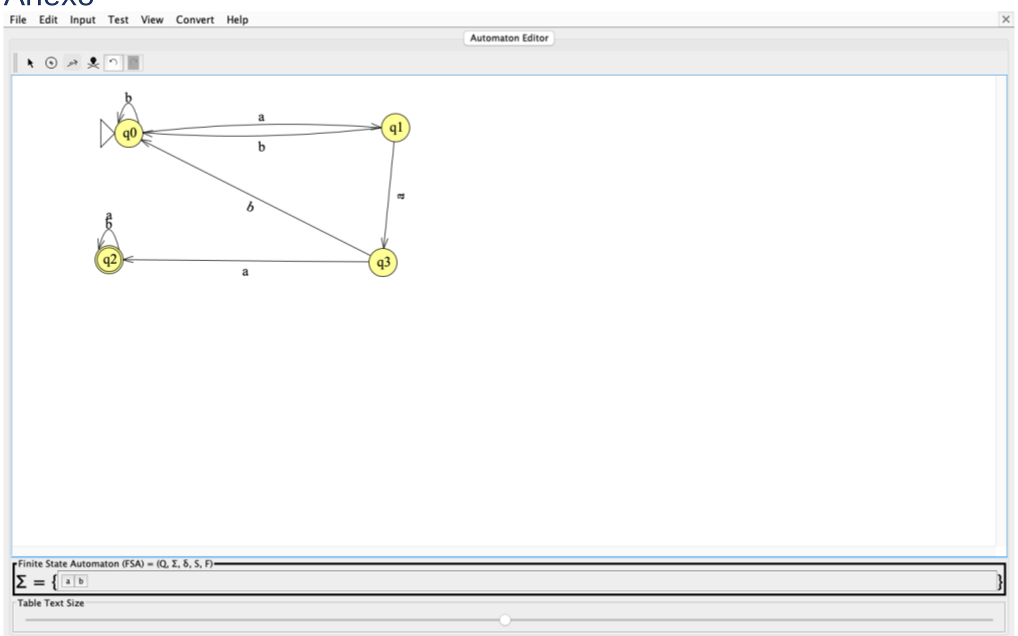
Comunmente existen tres operadores de las expresiones regulares:

- Unión, concatenaci´on y cerradura. Si L y M son dos lenguajes, su unión se denota por L ∪ M e.g. L = {001,10,111}, M ={ε,001}, entonces la unión será L ∪ M = {ε,10,001,111}.
- La concatenación de lenguajes se denota como LM o L.M e.g. L = {001,10,111}, M ={ε,001}, entonces la concatenación será LM ={001,10,111,001001,10001,111001}.
- Finalmente, la cerradura (o cerradura de Kleene) de un lenguaje L se denota como $L*$. Representa el conjunto de cadenas que puede que pueden formarse tomando cualquier número de cadenas de L, posiblemente con repeticiones y concatenando todas ellas e.g. si L = {0, 1}, $L*$ son todas las cadenas con 0’s y 1’s. Si L = {0, 11}, entonces $L*$ son todas las cadenas de 0’s y 1’s tal que los 1’s están en pareja. (INAOE, s.f.)

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

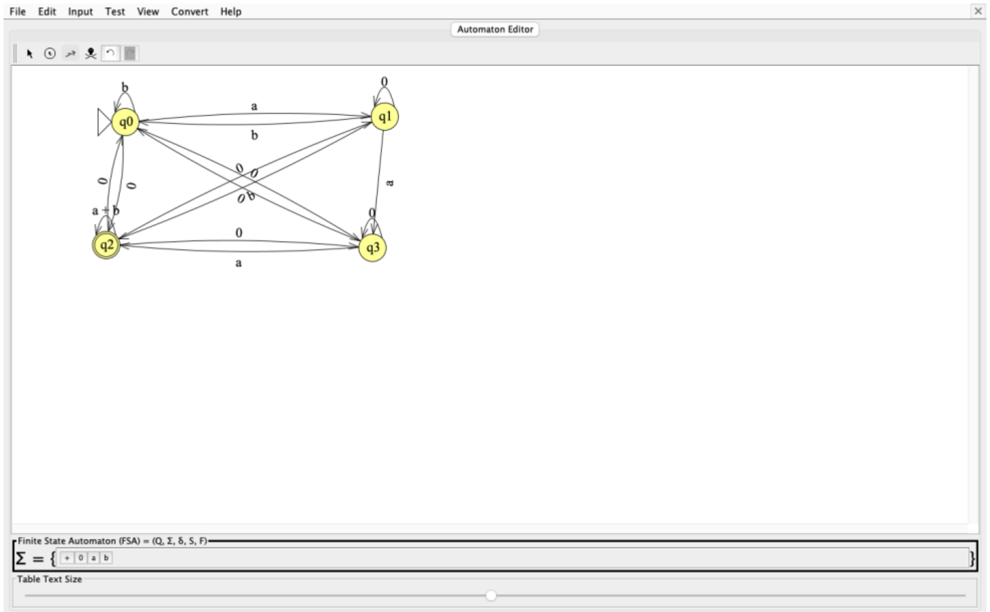
Básicamente este método consiste en seleccionar tres estados, qr, el cual no debera ser ni el estado inicial, ni ninguno de los estados finales o de aceptación, también se debería seleccionar un estado qx y qy, de manera que qx pueda llegar (por medio de transiciones) a qy utiliando a qr como estado intermedio entre estos. Después de haber selccionado estos estados, se debe proceder a eliminar el estado qr, haciendo una transición que vaya de qx a qy y que por medio de la concatenación de las transiciones que llegan de qx a qr y salen de qr a qy (incluyendo las que hacen bucle en qr). En caso de que ya exista una transición que va de qx a qy, se hace la unión de la Expreción Regular de dicha transición con la Expresión Regular 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. (Escher, s.f.)

Anexo

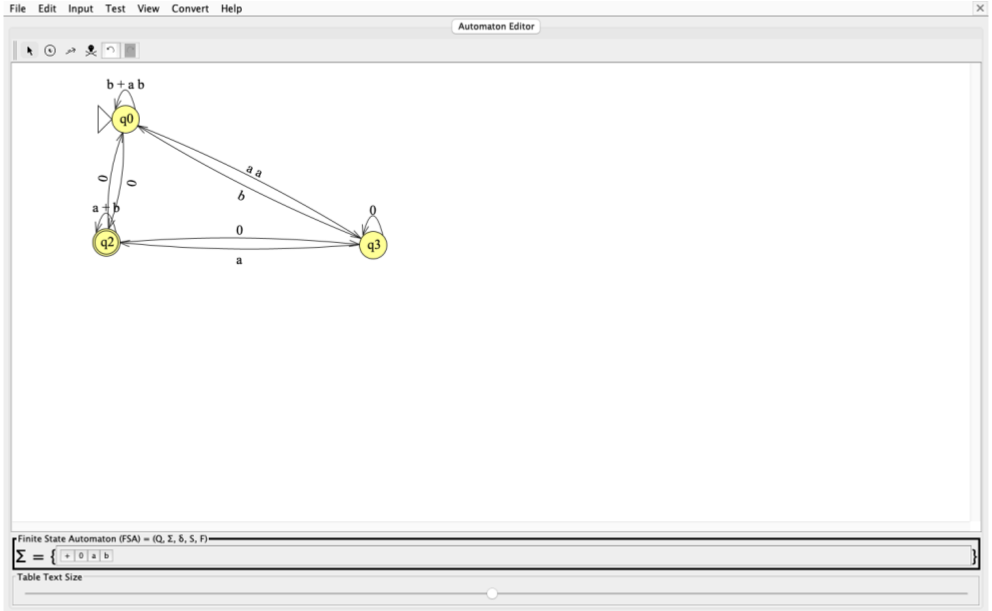


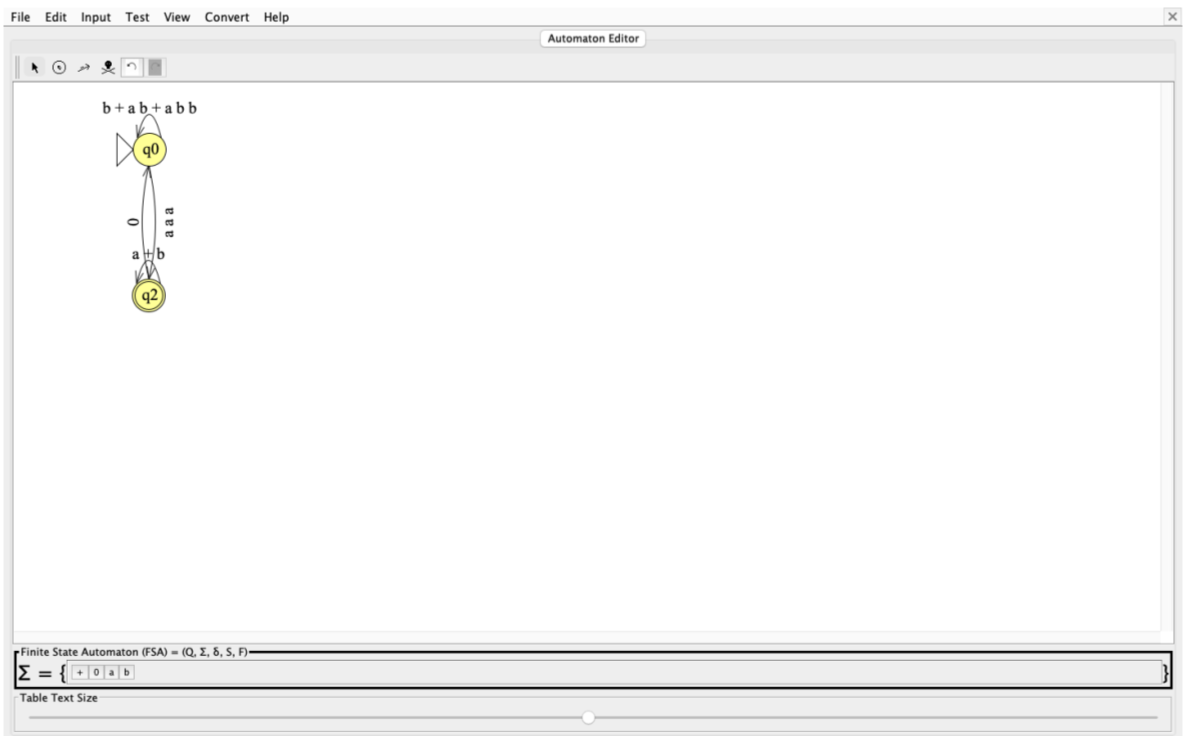
Paso 1: Por cada transición Qi^3 que pueda ser recorrida con múltiples símbolos, se hará una transición Qj (siendo esta una transición que contiene una Expresión Regular) que contenga los símbolos de dicha transición Qi representados como una Expresión Regular, específicamente como una unión.

Paso2: Por cada estado qi, se debe verificar si hay una transición Qj que llegue a cada estado qn (donde qn= qi) de la máquina. En caso de no existir esta transición se deberá agregar una transición que va desde qi hasta qn con el valor 0.



Paso 3: Seleccionar un estado qr, tal que qr no sea un estado inicial y/o final. Luego, por cada estado qx se selecciona un camino, pasando por qr, hacia cada estado qy del DFA, tal que qx ≠ qr y qy ≠ qr. Ahora se crea una transición Qj que tenga como Expresión Regular el símbolo (o Expresión Regular) de la transición que va de qx a qr concatenando con el símbolo de la transición que va de qr a qy. Al bucle que se hace en qr se le aplicará la operación de clausura (o clausura Kleene) y se concatenará con la Expresión Regular antes encontrada. A esta nueva transición Qj se le aplica una operación de unión con el símbolo de la transición que va de qx a qy. La transición Qj deberá quedar de la forma TiS*Tj+ Tk. Esta nueva transición Qj transitará del estado qx al estado qy.





Por lo cual hemos terminado de eliminar los estados no iniciales y no finales de nuestro DFA.

III.- Explicar leyes algebraicas de expresiones regulares

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

• Ley conmutativaparalaunión:L+M=M+L

• Ley asociativa para la unión: (L+M)+N = L+(M +N) • Ley asociativa para la concatenación: (LM)N = L(MN)

La concatenación no es conmutativa, es decir LM ≠ ML.

Leyes Distributivas

• Como la concatenación no es conmutativa, tenemos dos formas de la ley distributiva para la concatenación:

• Ley Distributiva Izquierda para la concatenación sobre unión: L(M + N) = LM + LN

• Ley Distributiva Derecha para la concatenación sobre unión: (M + N)L = ML + NL

Ley de Impotencia

• Se dice que un operador es idempotente (idempotent) si el resultado de aplicarlo a dos argumentos con el mismo valor es el mismo valor

• En general la suma no es idempotente: x + x ≠ x (aunque para algunos valores sí aplica como 0 + 0 = 0)

• En general la multiplicación tampoco es idempotente: x × x ≠ x

• La unión e intersección son ejemplos comunes de operadores idempotentes. Ley idempotente para la unión: L + L = L

Ley que involucran la cerradura

• $(L*)* = L*$ (Idempotencia para la cerradura)

• $\emptyset* = \varepsilon$

• $\varepsilon* = \varepsilon$

• L+ =LL∗ =L∗L,L+ sedefinecomoL+LL+LLL+... • L∗ =ε+L+LL+LLL+...

• LL∗ =Lε+LL+LLL+LLLL+... • L∗ = L+ + ε

• L?=ε+L

(INAOE, INAOEP, s.f.)

Conclusión

Mi conclusión para este tema de investigación es realmente interesante para mí, aunque siento que me falla aún, puesto a que las expresiones regulares requieren de conocimientos algebraicos como lógicos, por lo cual aún no logro comprender al cien por ciento, pero se me hace interesante poder aprenderlo y más aún que en el desarrollo de software se utiliza regularmente.

Para ello espero estudiarlo mucho para poder comprenderlo al cien por ciento, y así poder sentirlo fácilmente y realizarlo con facilidad.

Bibliografía

Learn.Microsoft. (s.f.). Obtenido de https://learn.microsoft.com/es-es/dotnet/standard/base-types/regular-expression-language-quick-reference

CS.FAMAF. (s.f.). Obtenido de https://cs.famaf.unc.edu.ar/~hoffmann/md18/10.html

Escher, F. D. (s.f.). *SCRIBD*. Obtenido de https://es.scribd.com/doc/12929632/DFA-a-Expresion-Regular

INAOE. (s.f.). Obtenido de https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Automatas/03_Automatas_ExpresionesRegularesLenguajes/CAPTUL1.PDF

INAOE. (s.f.). *INAOEP*. Obtenido de https://ccc.inaoep.mx/~emorales/Cursos/Automatas/ExpRegulares.pdf