UNIVERSIDAD AUTÓNOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

TITULO DE ACTIVIDAD

Actividad 1. Investigación y ejemplos.

NOMBRE, GRADO Y GRUPO DEL ALUMNO

Mónica Isabel López Flores del 6°M

NUMERO DE CONTROL DEL ALUMNO

A210525.

MATERIA

Compiladores.

DOCENTE

Luis Gutiérrez Alfaro.

Sábado 27 de enero de 2024.

Tuxtla Gutiérrez Chiapas, México.

Contenido

Definir El Concepto De Expresión Regular.

Instrucciones: Definir los siguientes Conceptos y de ejemplo de cada uno de los Incisos de I, II, III.

I. Tipos de Operadores de las Expresiones Regulares

Las expresiones regulares son un modelo capaz y flexible para buscar y procesar segmentos similares en un texto de entrada. Dicho modelo se compone de uno o más caracteres, operadores o estructuras en una cadena, estos caracteres se pueden clasificar en ocho categorías: operadores de carácter o escapes de carácter, operadores cuantificadores, operadores de combinación, operadores lógicos o delimitadores, clases de carácter, construcción de referencia inversa, construcción de alternancia y sustituciones.

Operadores de carácter o Escapes de carácter.

Este carácter se refiere al uso del carácter de barra diagonal inversa (\) en una cadena, que se usa para especificar que lo que sigue es un carácter especial. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.1

Escapes de carácter

carácter	Descripción
\	Es el carácter de escape para los caracteres especiales.
\b	Límite de palabra
\B	No es límite de palabra
\d	Un dígito
\D	Un carácter que no es dígito
\n	Nueva línea
\r	Carácter de retorno
\s	Un espacio

\S	Cualquier carácter, excepto espacio en blanco
\t	Tabulador
\w	Un carácter alfanumérico o guion bajo
\W	Un carácter no alfanumérico o guion bajo

Nota. Esta tabla muestra los caracteres de la categoría de escapes de operador con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Operadores cuantificadores

Este tipo de carácter se usa para indicar cuantas instancias del elemento anterior (siendo un carácter, grupo o clase de caracteres) debe haber para que se busquen las coincidencias en una cadena de entrada. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.2

Operadores cuantificadores

carácter	Descripción
.	Cualquier carácter, excepto una línea nueva
*	Cero o más con la máxima extensión posible
?	Cero o un carácter OR lo más corto posible
+	Uno o más
{<n>}	Exactamente <n> veces
{*n*,}	Coincide con el elemento anterior al menos *n* veces.
{<n>,<m>}	<n> a <m> veces
*?	Coincide con el elemento anterior cero o más veces, pero el menor número de veces que sea posible.
+?	Coincide con el elemento anterior una o más veces, pero el menor número de veces que sea posible.

??	Coincide con el elemento anterior cero o una vez, pero el menor número de veces que sea posible.
{*n*,}?	Coincide con el elemento anterior al menos *n* veces, pero el menor número de veces posible.
{*n*,*m*}?	Coincide con el elemento anterior entre *n* y *m* veces, pero el menor número de veces posible.

Nota. Esta tabla muestra los caracteres de la categoría de operadores cuantificadores con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Operadores de combinación

Este tipo de carácter se usa para indicar cuantas instancias del elemento anterior (siendo un carácter, grupo o clase de caracteres) debe haber para que se busquen las coincidencias en una cadena de entrada. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.3

Operadores de combinación

carácter	Descripción
.*	Cualquier cosa
.*?	Cualquier cosa lo más breve posible antes de

Nota. Esta tabla muestra los caracteres de la categoría de operadores de combinación con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Operadores lógicos o Delimitadores

Los delimitadores hacen que una coincidencia tenga éxito o no dependiendo de la posición en que se encuentre el elemento en la cadena sin consumir los caracteres A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.4

Delimitadores

carácter	Descripción
^	Comienzo de una línea OR no si está entre corchetes
$	Fin de una línea
()	Encapsulación
[]	Un carácter entre corchetes
|	OR
-	Intervalo
\A	Comienzo de una cadena
\Z	Fin de una cadena
\z	La coincidencia se debe producir al final de la cadena.
\G	La coincidencia debe ocurrir donde finalizó la coincidencia anterior, o si no había ninguna coincidencia anterior, en la posición de la cadena donde se inició la búsqueda de coincidencias.
\b	La coincidencia se debe producir en un límite entre un carácter \w (alfanumérico) y un carácter \W (no alfanumérico).
\B	La coincidencia no se debe producir en un límite \b.

Nota. Esta tabla muestra los caracteres de la categoría de delimitadores con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Clases de carácter

Este tipo de grupos de clase de carácter se usa para indicar que cualquier carácter es un elemento existente en la cadena de un juego de caracteres. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.5

Clases de carácter

carácter	Descripción
[*character_group*]	Coincide con cualquier carácter individual de *character_group*. De forma predeterminada, la coincidencia distingue entre mayúsculas y minúsculas.
[^*character_group*]	Negación: coincide con cualquier carácter individual que no esté en *character_group*. De forma predeterminada, los caracteres de *grupo_caracteres* distinguen entre mayúsculas y minúsculas.
[*first*-*last*]	El intervalo de caracteres: coincide con cualquier carácter individual en el intervalo de *primero* a *último*.
.	Carácter comodín: coincide con cualquier carácter individual a excepción de \n. Para que coincida un carácter de punto literal (o \u002E), debe ir precedido de un carácter de escape (\.).
\p{*name*}	Coincide con cualquier carácter individual que pertenezca a la categoría general Unicode o al bloque con nombre especificado por *nombre*.
\w	Coincide con cualquier carácter de una palabra.
\W	Coincide con cualquier carácter que no pertenezca a una palabra.
\s	Coincide con cualquier carácter que sea un espacio en blanco.
\S	Coincide con cualquier carácter que no sea un espacio en blanco.
\d	Coincide con cualquier dígito decimal.
\D	Coincide con cualquier carácter que no sea un dígito decimal.

Nota. Esta tabla muestra los caracteres de la categoría de clases de carácter con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Construcción de referencia inversa

Este tipo de caracteres se usan para identificar una subexpresión que existe más adelante en la misma cadena. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.6

Construcción de referencia inversa

carácter	Descripción
number	Referencia inversa Coincide con el valor de una subexpresión numerada.
\k<*name*>	Referencia inversa con nombre Coincide con el valor de una expresión con nombre

Nota. Esta tabla muestra los caracteres de la categoría de construcción de referencia inversa con una breve descripción para lo que sirven o especifican en una cadena u expresión regular.

Construcción de alternancia

Este tipo de caracteres se usan cambiar una cadena que habilita o no las coincidencias, es decir si es a o es b o es c. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.7

Construcción de alternancia

carácter	Descripción
|	Coincide con cualquier elemento separado por el carácter de barra vertical (|).
(?(*expression*)*yes*|*no*) o (?(*expression*)*yes*)	Coincide con *sí* si el patrón de expresión regular designado por *expresión* coincide; de lo contrario, coincide con la parte opcional *no*. *expresión* se interpreta como una aserción de ancho cero.

	Para evitar ambigüedades con un grupo de captura con nombre o numerado, puede usar una aserción explícita, como la siguiente: (?((?=*expression*))*yes*|*no*)
(?(*name*)*yes*|*no*) o **(?(*name*)*yes*)**	Coincide con *sí* si *nombre*, un grupo de captura con nombre o numerado, tiene una coincidencia; de lo contrario, coincide con la parte opcional *no*.

Nota. Esta tabla muestra los caracteres de la categoría de construcción de referencia inversa con una breve descripción para lo que sirven o especifican en una cadena u expresión regular

Sustituciones

Este tipo de caracteres se usan para cambiar una cadena que contiene un elemento por otro, es decir, si la cadena tiene una a, lo sustituirá por una d en donde estaba la a. A continuación, los caracteres pertenecientes a esta categoría se muestran en la siguiente tabla:

Tabla 1.8

Sustituciones

carácter	Descripción
$*number*	Sustituye la subcadena que coincide con el grupo *número*.
${*name*}	Sustituye la subcadena que coincide con el grupo con nombre *nombre*.
$$	Sustituye un "$" literal.
$&	Sustituye una copia de toda la coincidencia.
$`	Sustituye todo el texto de la cadena de entrada delante de la coincidencia.
$'	Sustituye todo el texto de la cadena de entrada detrás de la coincidencia.
$+	Sustituye el último grupo capturado.
$_	Sustituye toda la cadena de entrada.

Nota. Esta tabla muestra los caracteres de la categoría de construcción sustituciones con una breve descripción para lo que sirven o especifican en una cadena u expresión regular

II. Proceso de conversión de DFA a expresiones

Cuando se tiene un Autómata Finito Determinista (AFD o DFA) y se quiere convertir a una expresión regular que sea capaz de leer una cadena se deben de tener ciertas reglas a consideración y x número de pasos.

Reglas a Considerar para el Proceso de Conversión de un AFD a una ER

Para comenzar con el proceso de conversión de AFD a RE se deben identificar que tipo de regla se encuentra en la parte del autómata observado, estas reglas se clasifican en cinco tipos de reglas las cuales se presentan a continuación.

Regla 1

Si se tienen dos estados *q0* y *q1*, donde cuya transición para pasar de *q0* a *q1* se debe introducir una *a* entonces *q0 = a q1*.

Regla 2

Si se tienen tres estados *q0*, *q1 y q2*, cuya transición para pasar de *q0* a *q1* se debe introducir una *a*, y otra transición para pasar de *q0* a *q2* se debe introducir una *b* entonces q0 = a q1 + b q2

Regla 3

Si se tiene un estado *q0*, donde existe o cuya transición es un retorno/bucle a ese mismo estado *q0* al introducir una a entonces *q0 = a**

Regla 4

Si se tienen dos estados *q0* y *q1*, donde existe un retorno/bucle a ese mismo estado *q0* al introducir una *a* y otra transición para pasar de *q0* a *q1* se debe introducir una *b* entonces *q0 = a** *. (b q1), donde el* **.** *es usado como concatenación.*

Regla 5

Si se tiene un estado *q0* como estado final, entonces *q0 = λ*

Pasos para la Conversión de un AFD a una RE

El proceso de conversión de una AFD a una RE consta de n número de pasos.

Paso 1. Identificación de elementos

Observar el Autómata en cuestión e identificar el numero de estados presentes en el Autómata, así como la identificación de estado inicial y estado final.

Paso 2. Identificación de reglas.

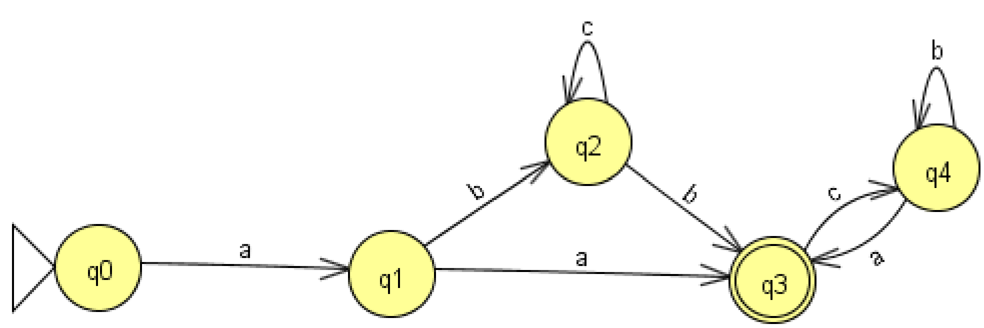
Una vez observado el autómata e identificado los números de estados, inicial y final, se comienza a identificar que tipo de regla o reglas se aplican en cada transición, empezando con el estado inicial hasta el estado final, anotando las ecuaciones pertenecientes a los movimiento o transiciones de los estados, yendo de uno en uno.

Paso 3. Unión o reducción de las ecuaciones.

Cuando se obtienes todas las ecuaciones de los estados, estos se comienzan a reducir o concatenar entre ellos haciendo uso nuevamente de las cinco reglas, esta reducción se comienza con el ultimo estado del autómata, sin importar si el estado final no es el ultimo en la línea, es decir, si se tienen tres estados q0, q1 y q2 distribuidos bajo un orden lineal del 0 al 1, pero el estado final es q1, igualmente se comienza a reducir los estados con el estado al final de la línea del autómata, q2, y así sucesivamente hasta llegar a la ecuación del estado inicial.

Ejemplo de Conversión de un AFD a una ER

Determinar la ER del siguiente autómata



Paso 1: El siguiente autómata consta de 4 estados: q1, q2, q3 y q4, siendo q0 el estado inicial y q3 el estado final.

Paso 2: Se aplican las reglas de las transiciones o movimientos de los estados, quedando de la siguiente manera: ***q0***= a q1, ***q1***= b q2 | a q3, ***q2***= c q2 | b q3, ***q3***= c q4 | λ, ***q4***= b q4 | a q3

Paso 3: Como la ecuación de q4 es la ultima en la línea, se empieza a reducir las ecuaciones de q4 para luego combinarlo con las ecuaciones de q3 y así sucesivamente, haciendo uso de las reglas dadas anteriormente quedando de la siguiente manera:

q0= a q1 *q1*= b q2 | a q3, *q2*= c q2 | b q3, *q3*= c q4 | λ, *q4*= b q4 | a q3	→	*q4*= b* . a q3 q3= (cb*a)q3 | λ q3= (cb*a)* q2= c* b (cb*a)* q1= (bc*b | a). (cb*a)*	→	Quedando finalmente la expresión regular como: *q0= a(bc*b | a). (cb*a)**

III. Leyes algebraicas de las expresiones regulares

Las expresiones regulares se relacionan con algunas leyes algebraicas comunes que reflejan propiedades de los operadores en las operaciones regulares y pueden ayudar a simplificar y comprender la manipulación de dichas expresiones, Las leyes algebraicas relacionadas son: Ley Conmutativa, Ley Asociativa, Ley de Identidad, Ley de anulación o de Elemento Nulo, Ley Distributiva y Ley de Idempotencia.

Ley Conmutativa

Esta ley se encuentra presente cuando un lenguaje L es conmutativo solo si se cumple que un operador pueda cambiar el orden de sus operadores y aun así obtener el mismo resultado

Por lo tanto, la ley conmutativa para la unión establece que se puede unir dos lenguajes en cualquier orden.

Ejemplo para la Ley Conmutativa para la Unión: $\boldsymbol{L + M = M + L}$

Ley Asociativa

Esta ley se aplica o se encuentra presente cuando en un autómata se encuentran varios operandos que se aplican dos veces y se necesitan reagrupar para simplificar y sacar su expresión regular. En esta Ley sirve para realizar dos cosas unir operandos (Ley asociativa para la unión) o concatenar operandos (Ley asociativa para la concatenación).

Ejemplo para la Ley Asociativa de la Unión: **(L + M) + N = L + (M + N)**

Ejemplo para la Ley Asociativa para la Concatenación: **(LM) N = L (MN)**

Ley de Identidad

Esta ley se encuentra presente cuando un valor que operado con cualquier otro número no lo altera.

Ejemplo para la ley de Identidad: **0 + L = L + /0 = L**

Donde */0* es el elemento identidad para la unión

Ley de Anulación o de Elemento Nulo

Esta ley se encuentra presente cuando se tiene un elemento vacío, que actúa como el elemento neutro y se tiene que concatenar con otro elemento, la expresión original se anula.

Ejemplo: **0 x n = n x 0 = 0**

Ley Distributiva

Esta ley se encuentra presente cuando se tienen dos operadores y uno de ellos puede aplicarse por separado a cada argumento del otro operador, esta ley puede realizarse de dos formas: por la izquierda y por la derecha, de la concatenación respecto a la unión.

Ejemplo para la Ley Distributiva por la izquierda: **L (M + N) = LM + LN**

Ejemplo para la Ley Distributiva por la Derecha: **(M +N) L = ML+NL**

Ley de Idempotencia

Esta ley esta presente cuando el resultado, al haberlo aplicado a dos valores iguales es dicho valor, es decir, que si se toma la unión de dos expresiones idénticas se puede cambiar por una copia de la expresión o valor.

Ejemplo: **L + L = L**

Referencias bibliográficas.

Ejemplos de expresiones regulares. (s/f). Vmware.com. Recuperado el 26 de enero de 2024, de https://docs.vmware.com/es/vRealize-Log-Insight/8.10/com.vmware.log-insight.user.doc/GUID-88B2952D-3112-46BC-B126-84C9BF38B6D2.html

Lenguaje de expresiones regulares - Referencia rápida. (s/f). Microsoft.com. Recuperado el 26 de enero de 2024, de https://learn.microsoft.com/es-es/dotnet/standard/base-types/regular-expression-language-quick-reference

Piñeiro, M. A. [@matematicamaravillosa]. (2020, junio 27). *Autómatas Finitos*. Youtube. https://www.youtube.com/watch?v=ScR1T1DME14

(S/f). Inaoep.mx. Recuperado el 26 de enero de 2024, de https://ccc.inaoep.mx/ingreso/automatas/expresionesRegulares.pdf