UNIVERSIDAD AUTÓNOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN “CAMPUS I”

LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

COMPILADORES

6° “M”

ALUMNO:

JORGE DAVID LEPE HERNÁNDEZ

DOCENTE: D.S.C LUIS GUTIERREZ ALFARO

ACTIVIDAD I

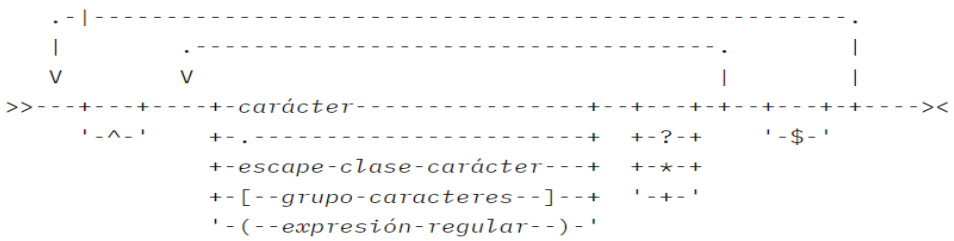
TUXTLA GUTIÉRREZ, CHIAPAS A 28 DE ENERO DE 2024

Contenido

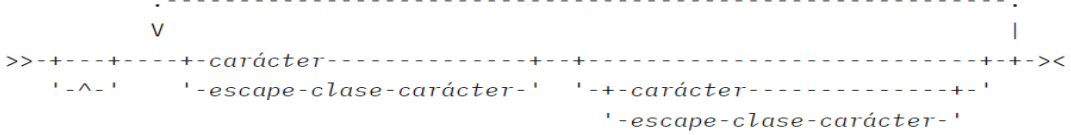
I.- Explicar los tipos de operadores de expresiones regulares

Comúnmente existen tres operadores de las expresiones regulares: Unión, concatenación y cerradura. Si L y M son dos lenguajes, su unión se denota por L ∪ M e.g. L = {001,10,111}, M ={,001}, entonces la unión será L ∪ M = {,10,001,111}. La concatenación de lenguajes se denota como LM o L.M e.g. L = {001,10,111}, M ={,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. Para calcular L ∗ se debe calcular L i para cada i y tomar la unión de todos estos lenguajes. L i tiene 2i elementos. Aunque cada L i es finito, la unión del número de términos de L i es en general un conjunto infinito e.g. ∅ ∗ = {} o ∅ 0 = {}. Generalizando, para todo i mayor o igual que uno, ∅ i es el conjunto vacío, no se puede seleccionar ninguna cadena del conjunto vacío.

Sintaxis



grupo-caracteres



Carácter

En una expresión regular, un carácter es un carácter XML normal que no es un metacarácter.

Metacaracteres

Los metacaracteres son caracteres de control en las expresiones regulares. Los metacaracteres de expresiones regulares que están soportados actualmente son:

barra inclinada invertida (\)

Inicia un escape de clase de carácter. Un escape de clase de carácter indica que el metacarácter siguiente debe utilizarse como carácter, en lugar de un metacarácter.

punto (.)

Coincide con cualquier carácter individual, excepto con el carácter de nueva línea (\n).

signo de intercalación (^)

Si el carácter de intercalación aparece fuera de una clase de carácter, los caracteres que le siguen coinciden con el inicio de la serie de entrada o, para series de entrada de varias líneas, con el inicio de una línea. Una serie de entrada se considera como una serie de entrada de varias líneas si la función que utiliza la serie de entrada incluye el distintivo m.

Si el carácter de intercalación aparece como primer carácter en una clase de carácter, el carácter de intercalación actúa como un signo de negación. Se produce una coincidencia si ninguno de los caracteres del grupo de caracteres aparece en la serie que se compara con la expresión regular.

signo de dólar ($)

Coincide con el final de la serie de entrada o, para series de entrada de varias líneas, con el final de una línea. Una serie de entrada se considera como una serie de

entrada de varias líneas si la función que utiliza la serie de entrada incluye el distintivo m.

signo de interrogación (?)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular cero o una vez.

asterisco (*)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular de cero o más veces.

signo Más (+)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular una o más veces.

barra vertical (|)

Coincide con el carácter (o grupo de caracteres) anterior o el carácter (o grupo de caracteres) siguiente.

corchete de apertura ([) y corchete de cierre (])

Los corchetes de apertura y cierre y el grupo de caracteres que incluyen definen una clase de carácter. Por ejemplo, la clase de carácter [aeiou] coincide con cualquier vocal. Las clases de carácter también admiten rangos de caracteres. Por ejemplo:

[a-z] significa cualquier letra minúscula.

[a-p] significa cualquier letra minúscula de la a a la p.

[0-9] significa cualquier dígito único.

paréntesis de apertura (() y paréntesis de cierre ())

Un paréntesis de apertura y uno de cierre indican una agrupación de algunos caracteres dentro de una expresión regular. A continuación, puede aplicar un operador, como por ejemplo un operador de repetición, a todo el grupo.

La llave de apertura ({) y la llave de cierre (}) también son metacaracteres, pero actualmente no están soportados.

Escape-clase-carácter

Un escape de clase de carácter especifica que determinados caracteres especiales deben tratarse como caracteres, en lugar de realizar alguna función. Un escape de clase de carácter consta de una barra inclinada invertida (\), seguida de un único metacarácter, carácter de nueva línea, carácter de retorno o carácter de tabulación. La tabla siguiente lista los escapes de clase de carácter.

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_r como estado intermedio entre estos. Después de haber seleccionado estos estados, se debe proceder a eliminar el estado q_r, 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_r y salen de q_r a q_y (incluyendo las que hacen un bucle en q_r). 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.

En el párrafo anterior se explico a groso modo (sin ningún detalle) como se debe reducir la máquina de estados, y también vimos que al reducirla la debemos transformar a una Expresión Regular. Ahora vamos a definir paso por paso y con detalle cómo se debe hacer la reducción. Veremos que al hacer la reducción se pueden dar tres casos en los que tendremos que generar la Expresión Regular de una manera ligeramente diferente.

Para un mejor entendimiento de la eliminación de los estados utilizaremos el siguiente ejemplo:

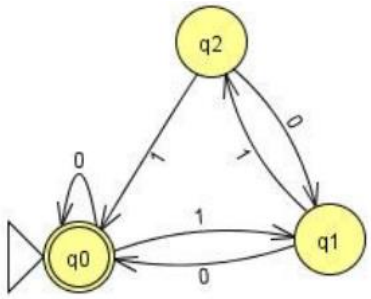
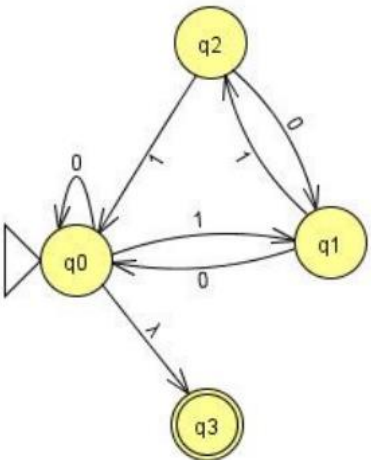


ILUSTRACIÓN 6: DFA ORIGINAL

Antes de comenzar a reducir el DFA, debemos llevar a cabo un paso alternativo que nos permita separar el estado inicial y el estado final en dos estados distintos.

PASO A1: Si existe un estado q_x talque q_x sea el estado inicial y un estado final, haremos un nuevo estado q_y que pasara a ser el estado final y q_x permanecerá solo como estado inicial. Se debe crear una transición Q_j que tendrá como símbolo ε[5].

APLICACIÓN: Creamos un nuevo estado q_3 y ponemos una transición que vaya de q_0 a q_3 con el símbolo ε.



Una vez hecha la separación de los estados podemos proseguir con los pasos 1-4. Como podemos ver, cada transición de nuestro DFA tiene un símbolo único que le permite ser transitada, por lo tanto el paso 1 nos generará en mismo DFA de la ilustración 7. Aplicamos el paso 2 para poner todas las transiciones que nos hacen falta. Con esto obtenemos:

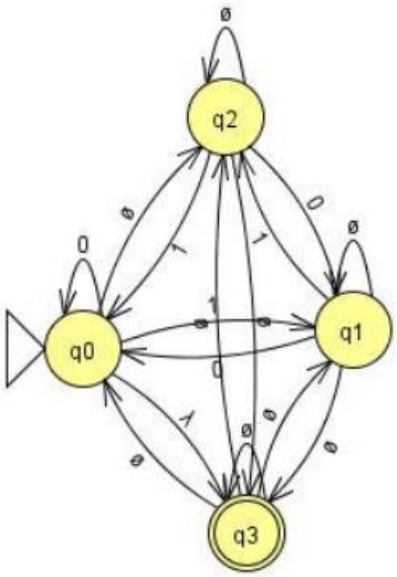


ILUSTRACIÓN 8: PASO 2

Esta es la tabla con las expresiones regulares ya simplificadas que nos resulta después de haber hecho el paso tres para el estado q_1:

q_x	q_y	Q_j
0	**0**	$0 + 01$
0	**2**	11
0	**3**	λ
2	**0**	$1 + 00$
2	**2**	01
2	**3**	$\emptyset$
3	**0**	$\emptyset$
3	**2**	$\emptyset$
3	**3**	$\emptyset$

A continuación el DFA que nos resulta al haber eliminado q_1:

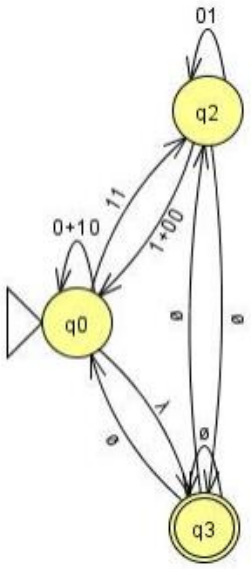


ILUSTRACIÓN 9: PASO 3

Hacemos lo mismo para q_2, con esto tendremos solo estados iniciales y finales en nuestro DFA.
La tabla de transiciones es:

q_x	q_y	Q_j
0	**0**	$0 + 10 + 11(01)^*(1 + 00)$
0	**3**	λ
3	**0**	Ø
3	**3**	Ø

Al aplicar el paso tres a todos los estados no iniciales y no finales, obtenemos el siguiente DFA:

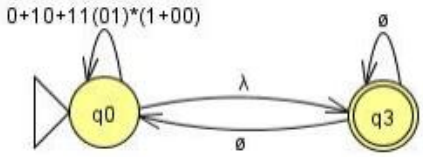


ILUSTRACIÓN 10: PASO 3

PASO A2: Si existe un estado q_x inicial y q_y final talque hay una transición λ entre estos estados, se deberá hacer una concatenación de las cerradura Kleene de las Expresiones Regulares que tienen la transición que hace bucle en q_x, la transición que va de q_y a q_x y la transición que hace bucle en q_y. A este conjunto se le aplicará la cerradura Kleeme y se le concatenará la cerradura Kleene de la transición que hace bucle en q_x y la transición que hace bucle en q_y.

APLICACIÓN: Al hacer la concatenación de las cerraduras Kleene de el bucle en q_0, el bucle en q_3 y la transición que va de q_3 a q_0 nos queda le Expresión Regular:

$$(0 + 10 + 11(01)^*(1 + 00))^* \emptyset^*\emptyset^*$$

Ahora le aplicamos la cerradura Kleene a este conjunto:

$$((0 + 10 + 11(01)^*(1 + 00))^* \emptyset^*\emptyset^*)^*$$

Proseguimos a concatenarlo con el bucle en q_0 y en q_3:

$$((0 + 10 + 11(01)^*(1 + 00))^* \emptyset^*\emptyset^*)^*(0 + 10 + 11(01)^*(1 + 00)^*)^*\emptyset^*$$

Al simplificar esta Expresion Regular nos queda:

$$(\mathbf{0 + 10 + 11(01)^*(1 + 00)})^*$$

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 unión: ´ L + M = M + L

• Ley asociativa para la unión: ´ (L + M) + N = L + (M + N)

• Ley asociativa para la concatenación: ´ (LM)N = L(MN)

NOTA: La concatenación no es conmutativa, es decir ´

LM 6= ML

• Una identidad para un operador es un valor tal que cuando el operador se aplica a la identidad y a algún 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 × x = x × 1 = x

• Un aniquilador para un operador es un valor tal que cuando el operador se aplica al aniquilador y algún otro valor, el resultado es el aniquilador.

• 0 es el aniquilador para la multiplicacion: 0 × x = x × 0 = 0

• No hay aniquilador para la suma

• ∅ es la identidad para la union: ´ ∅ + L = L + ∅ = L

• e es la identidad para la concatenacion: ´eL = Le = L

• ∅ es el aniquilador para la concatenacion: ´ ∅L = L∅ = ∅

NOTA: Estas leyes las utilizamos para hacer simplificaciones

• 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

• 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 6= x (aunque para algunos valores si aplica como 0 + 0 = 0)

• En general la multiplicación tampoco es idempotente: ´

x × x 6= x

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

BIBLIOGRAFIA

IBM documentation. (2023, October 10).

https://www.ibm.com/docs/es/i/7.3?topic=expressions-regular

FernandoEscher. (n.d.). *DFA a expresion regular*. Scribd.

https://es.scribd.com/doc/12929632/DFA-a-Expresion-Regular

https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Automatas/03_Automatas_ExpresionesRegularesLenguajes/CAPTUL1.PDF

file:///C:/Users/David%20LH/Downloads/ilide.info-dfa-a-expresion-regular-pr_bb6aa88a494603b4545cc67faf758315.pdf

https://ccc.inaoep.mx/ingreso/automatas/expresionesRegulares.pdf