



**UNIVERSIDAD AUTÓNOMA DE CHIAPAS**



**FACULTAD: CONTADURÍA Y ADMINISTRACIÓN, CAMPUS I**

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

**UNIDAD DE APRENDIZAJE: COMPILADORES**

**DOCENTE: D.S.C LUIS GUTIÉRREZ ALFARO**

**ALUMNO: EDSSON DANIEL LÓPEZ MENDOZA**

**NUMERO DE CONTROL: A210089**

**GRADO Y GRUPO: 6° "M"**

**ACTIVIDAD I.- INVESTIGACIÓN Y EJEMPLOS**

**TUXTLA GUTIÉRREZ, CHIAPAS**

**26 DE ENERO DE 2024**

## Contenido

<b>I. Explicar los tipos de operadores de expresiones regulares. ....</b>	<b>1</b>
Operadores de caracteres .....	1
Operadores cuantificadores .....	2
Operadores de combinación .....	2
Operadores lógicos.....	3
Operadores de lectura previa.....	3
Otros ejemplos de expresiones regulares .....	4
<b>II. Explicar el proceso de conversión de DFA a expresiones regulares.....</b>	<b>5</b>
<b>III. Explicar leyes algebraicas de expresiones regulares. ....</b>	<b>7</b>
Asociatividad y Conmutatividad.....	7
Identidades y Aniquiladores.....	7
Leyes Distributivas.....	8
Ley de Idempotencia .....	8
Leyes que involucran la cerradura .....	8
Descubriendo leyes para RE .....	9
<b>Bibliografía.....</b>	<b>10</b>

## I. Explicar los tipos de operadores de expresiones regulares.

### Operadores de caracteres

Expresión regular	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

Por ejemplo, si tiene la cadena 1234-5678 y aplica las siguientes expresiones regulares, obtendrá:

Expresión regular	Resultado
\d	1
\d+	1234
\w+	1234
\S	1234-5678

### Operadores cuantificadores

Expresión regular	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>, <m>}	<n> a <m> veces

Por ejemplo, si tiene la cadena aaaaaa y aplica las siguientes expresiones regulares

Expresión regular	Resultado
.	a
*	aaaaaa
.*?	aaaaaa
{1}	a
{1,2}	aa

### Operadores de combinación

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

Por ejemplo, si tiene la cadena a b 3 hi d hi y aplica las siguientes expresiones regulares

Expresión regular	Resultado
a.* hi	b 3 hi d
a.*? hi	b 3

### Operadores lógicos

Expresión regular	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

Por ejemplo, si aplica las siguientes expresiones regulares

Expresión regular	Resultado
(hola)?	Contiene hola OR no contiene hola
(a b c)	a OR b OR c
[a-cp]	a OR b OR c OR p
palabra \$	Finaliza con palabra seguido por nada más

### Operadores de lectura previa

Expresión regular	Descripción
?=	Lectura previa positiva (incluye)
?!=	Lectura previa negativa (no incluye)

Por ejemplo, si aplica las siguientes expresiones regulares

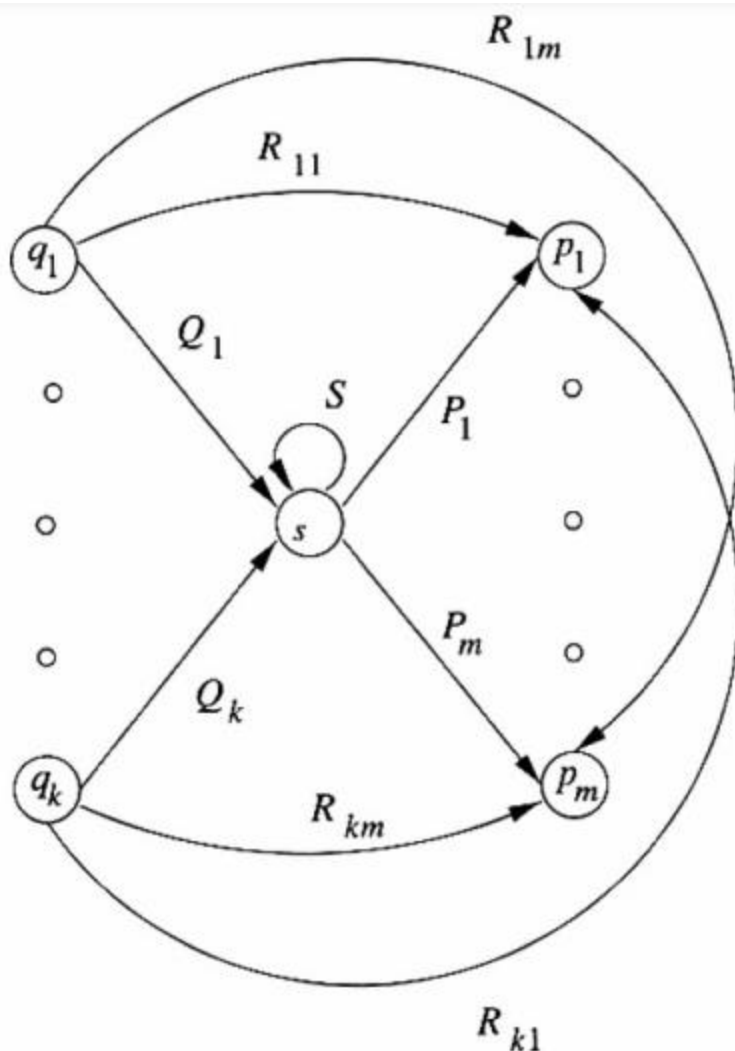
Expresión regular	Resultado
is (?=w+)\w{2} primario	es FT primario? falso
opid=(?!WFU-1fecf8f9)\S+	WFU-3c9bb994

### Otros ejemplos de expresiones regulares

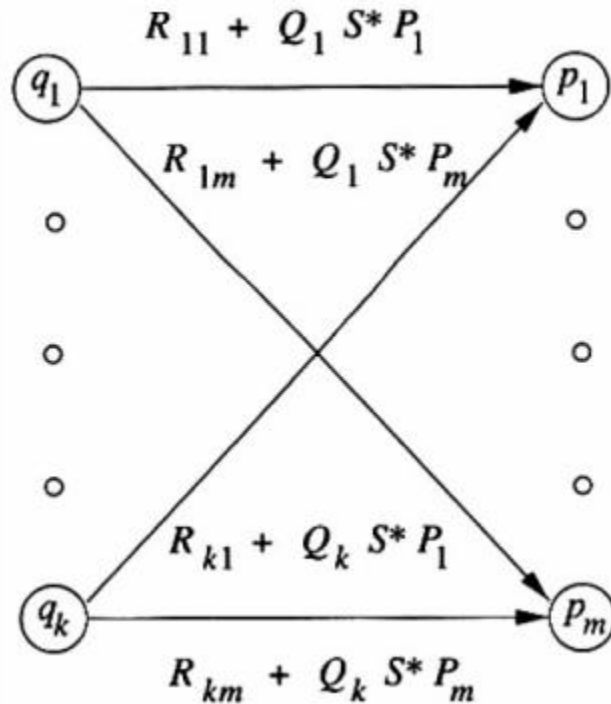
Expresión regular	Descripción
[xyz]	x, y, o z
(información advertencia error)	información, advertencia o error
[a-z]	Una letra minúscula
[^a-z]	No una letra minúscula
[a-z] +	Una o más letras minúsculas
[a-z] *	Cero o más letras minúsculas
[a-z] ?	Cero o una letra minúscula
[a-z] {3}	Exactamente tres 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

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

- Evita duplicar trabajo en algunos puntos del teorema anterior
- Ahora utilizaremos autómatas que podr'an tener RE como etiquetas.
- El lenguaje del autómata es la unión de todas las rutas que van del estado inicial a un estado de aceptación.
  - Concatenando los lenguajes de las RE que van a través de la ruta.
  - En la siguiente figura se muestra un autómata al cual se va a eliminar el estado "s".



- Se eliminan todos los arcos que incluyen a “s”
- Se introduce, para cada predecesor  $q_i$  de s y cada sucesor  $p_j$  de s, una RE que representa todas las rutas que inician en  $q_i$ , van a s, quizás hacen un loop en s cero o más veces, y finalmente van a  $p_j$ .
- La expresión para estas rutas es  $Q_i S^* P_j$ .
- Esta expresión se suma (con el operador unión) al arco que va de  $q_i$  a  $p_j$ .
- Si este arco no existe, se añade primero uno con la RE  $\emptyset$
- El autómata resultante después de la eliminación de “s” es el siguiente:





### III. Explicar leyes algebraicas de expresiones regulares.

#### Asociatividad y Conmutatividad

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:  $LM \neq ML$

#### Identidades y Aniquiladores

- 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:  $1 \times x = x \times 1 = x$
- 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 multiplicación:  $0 \times x = x \times 0 = 0$
- No hay aniquilador para la suma
- $\emptyset$  es la identidad para la unión:  $\emptyset + L = L + \emptyset = L$
- E es la identidad para la concatenación:  $EL = LE = L$
- $\emptyset$  es el aniquilador para la concatenación:  $\emptyset L = L\emptyset = \emptyset$

NOTA: Estas leyes las utilizamos para hacer simplificaciones

### 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 Idempotencia

- 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:  $x + x \neq x$  (aunque para algunos valores sí aplica como  $0 + 0 = 0$ )
- En general la multiplicación tampoco es idempotente:  $x \times x \neq x$
- La unión e intersección son ejemplos comunes de operadores idempotentes. Ley idempotente para la unión:  $L + L = L$

### Leyes que involucran la cerradura

- $(L^*)^* = L^*$  (Idempotencia para la cerradura)
- $\emptyset^* = E$
- $E^* = E$
- $L^+ = LL^* = L^*L$ ,  $L^+$  se define como  $L + LL + LLL + \dots$
- $L^* = E + L + LL + LLL + \dots$
- $LL^* = LE + LL + LLL + LLLL + \dots$
- $L^* = L^+ + E$
- $L^? = E + L$

### Descubriendo leyes para RE

- Se puede proponer una variedad infinita de leyes para RE
- Se reduce a probar la igualdad de dos lenguajes específicos
- Ejemplo: probar que  $(L + M)^* = (L^*M^*)^*$
- Para esto, probamos que las cadenas que están en  $(L + M)^*$  también están en  $(L^*M^*)^*$ , y
- Probamos que las cadenas que están en  $(L^*M^*)^*$  también están en  $(L + M)^*$
- Cualquier RE con variables se puede ver como una RE concreta sin variables, viendo cada variable como si fuera un símbolo diferente
- La expresión  $(L + M)^*$  se puede ver cómo  $(a + b)^*$ . Utilizamos esta forma como una guía para concluir sobre los lenguajes.
- Podemos analizar el lenguaje que nos describe:  $(a + b)^*$
- Y analizar el lenguaje que nos describe:  $(a^*b^*)^*$

## Bibliografía

*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>

*Transformar Un dfa en Una expresion regular.* (s/f). Prezi.com. Recuperado el 26

de enero de 2024, de <https://prezi.com/fzrkq7tpeo9k/transformar-un-dfa-en-una-expresion-regular/>

(S/f). Inaoep.mx. Recuperado el 26 de enero de 2024, de

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