Universidad Autónoma de Chiapas Facultad de Contaduría y Administración, Campus I
Licenciatura en Ingeniería y Desarrollo de Tecnologías de Software

ANÁLISIS LÉXICO

Actividad I. Define los siguientes conceptos

Compiladores

Luisa Maria Santiago Siliceo

D.S.C. Luis Gutiérrez Alfaro

Expresiones Regulares

Son un equivalente algebraico para un autómata, es una secuencia de caracteres que actúa como modelo para la coincidencia y manipulación de series.

Para crear una expresión regular debes utilizar una sintaxis específica, es decir, caracteres especiales y reglas de construcción.

Tipos de operadores

.	Coincide con cualquier carácter excepto una nueva línea.
\d	Coincide con cualquier dígito (equivalente a [0-9]).
\D	Coincide con cualquier carácter que no sea un dígito (equivalente a [^0-9]).
.	Coincide con cualquier carácter excepto una nueva línea.
\d	Coincide con cualquier dígito (equivalente a [0-9]).
\w	Coincide con cualquier carácter alfanumérico (letras y números) y el guion bajo (equivalente a [a-zA-Z0-9_]).
\s	Coincide con cualquier carácter de espacio en blanco (espacios, tabulaciones, saltos de línea).
*	Coincide con 0 o más repeticiones del carácter o grupo anterior.
+	Coincide con 1 o más repeticiones del carácter o grupo anterior.
?	Coincide con 0 o 1 repetición del carácter o grupo anterior.
{n}	Coincide exactamente con n repeticiones del carácter o grupo anterior.
{n,}	Coincide con al menos n repeticiones del carácter o grupo anterior.

Ejemplo.

Esta es una expresión regular simple que coincide con cualquier número de teléfono de diez cifras con el patrón nnn-nnn-nnnn:

`\d{3}-\d{3}-\d{4}`

{n,m}	Coincide con entre n y m repeticiones del carácter o grupo anterior.
^	Coincide con el inicio de una línea o cadena.
$	Coincide con el final de una línea o cadena.
[abc]	Coincide con cualquiera de los caracteres dentro de los corchetes (a, b o c).
[^abc]	Coincide con cualquier carácter que no esté dentro de los corchetes.
[a-z]	Coincide con cualquier carácter en el rango de a a z.
(abc)	Coincide con la secuencia exacta de caracteres abc.
a|b	Coincide con a o b (alternancia).
\	Se usa para escapar caracteres especiales, por ejemplo, \. para buscar un punto literal.

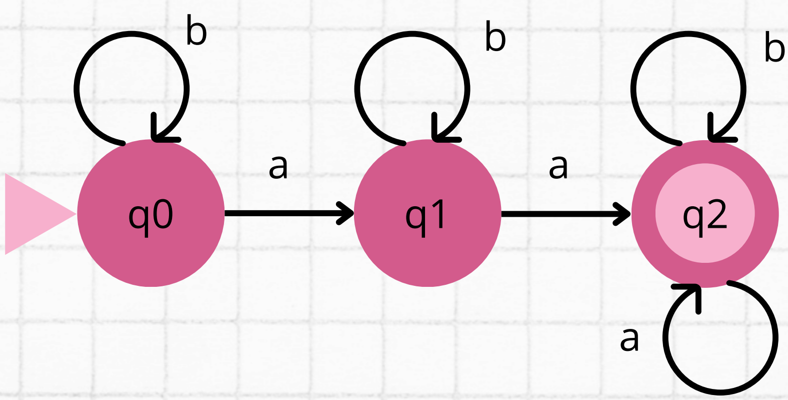
Conversión de DFA a expresiones regulares.

Supongamos que tenemos el siguiente DFA:

Estados: q0(inicio), q1, q2(final)

Alfabeto: {a, b}

Transiciones:
q0 con a → q1
q0 con b → q0
q1 con a → q2
q1 con b → q1
q2 con a → q2
q2 con b → q2



Paso 1. Representar el DFA

Un DFA se define por un conjunto de componentes:

- Un conjunto de estados.
- Un alfabeto de entrada.
- Una función de transición que define cómo los estados cambian con cada símbolo de entrada.
- Un estado inicial.
- Un conjunto de estados finales o de aceptación.

Estados	a	b
q0	q1	q0
q1	q2	q1
q2	q2	q2

Paso 2. Construir una Tabla de Transiciones

Cada fila de la tabla corresponde a un estado y cada columna corresponde a un símbolo del alfabeto. Las celdas indican el estado al que se transita dado el símbolo correspondiente.

Paso 3. Agregar un Estado de Inicio y un Estado Final Únicos

Para simplificar el proceso, añade un nuevo estado de inicio y un nuevo estado final único al DFA. Conecta el nuevo estado de inicio al estado inicial original con una transición vacía (ε). Conecta el estado final único a todos los estados finales originales con transiciones vacías (ε).

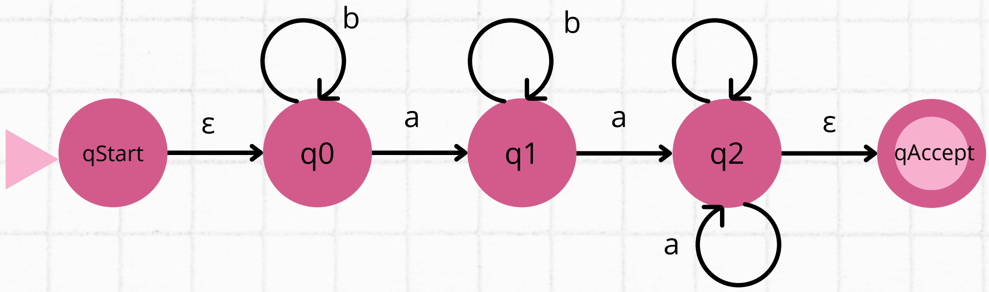
Agregamos un nuevo estado inicial qStart y un nuevo estado final qAccept.

Conectamos:

qStart a q0 con una transición vacía (ε).

q2 a qAccept con una transición vacía (ε).

El nuevo estado qacceptq_{accept} qaccept será nuestro único estado final.



Paso 4. **Aplicar el Algoritmo de Eliminación de Estados**

Este algoritmo implica eliminar los estados del DFA uno por uno y actualizar las transiciones y las expresiones regulares asociadas. Aquí está el procedimiento:

1. Seleccionar un Estado para Eliminar:

Elige un estado que no sea ni el estado inicial ni el estado final.

2. Actualizar las Transiciones:

Para cada transición que pasa por el estado que estás eliminando, combina las transiciones entrantes y salientes utilizando expresiones regulares.

Esto se hace mediante la sustitución de transiciones a través del estado eliminado con expresiones regulares que describen todas las posibles transiciones a través de ese estado.

Si hay una transición del estado qi qk con un símbolo a, y otra transición del estado qk a qj con un símbolo b, y estás eliminando el estado qk, entonces la nueva transición entre qi y qj debería incluir la expresión regular que representa el paso por qk.

3. Repetir el Proceso

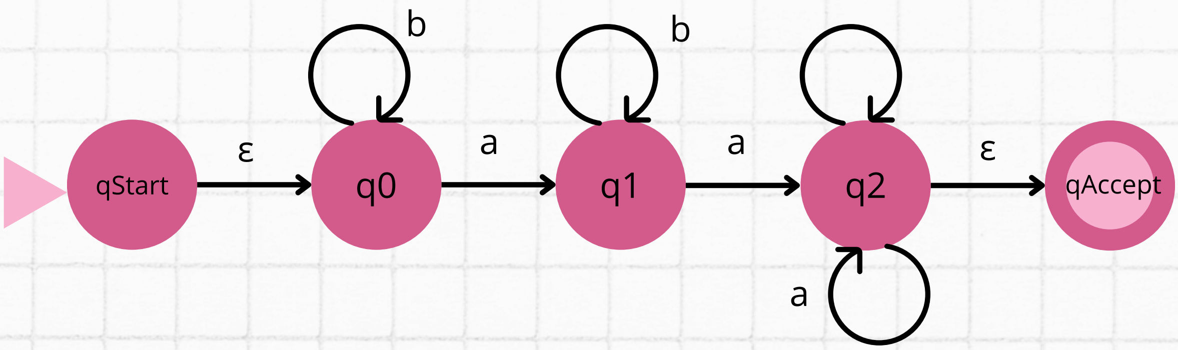
Repite el proceso de eliminación de estados hasta que sólo queden el estado inicial y el estado final. En este punto, las transiciones entre el estado inicial y el estado final serán representadas por una expresión regular.

Eliminamos estados intermedios (en este caso, q1) y actualizamos las transiciones:

Eliminar el Estado q1

Identificar Transiciones Entrantes y Salientes:

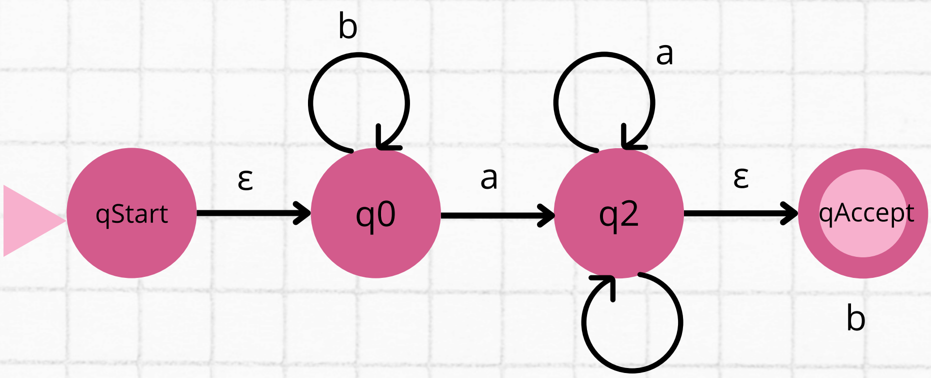
- Transiciones entrantes a q1:
  q0→q1 con a
- Transiciones salientes desde q1:
  q1→q2 con a
  q1→q1 con b



Actualizar Transiciones:
De q0 a q2 pasando por q1:
- Primero, q0→q1 con a y luego q1→q2 con a. Así que la transición de q0 a q2 es a · a=aa

Transiciones desde q0 a q1 con b:
- q1→q1 con b significa que q1 puede tener cualquier cantidad de b's. La transición de q0 a q2 a través de q1 será a(b∗)a.

Después de actualizar, las nuevas transiciones son:
q0 con a → q2
q0 con b → q0
q2 con a → q2
q2 con b → q2



Paso 5.

Obtener la Expresión Regular Final

Una vez que hayas eliminado todos los estados intermedios, la expresión regular que representa el conjunto de cadenas aceptadas por el DFA será la expresión regular que describe las transiciones entre el estado inicial y el estado final.

Ahora que solo quedan q0 y q2, la expresión regular que describe las transiciones de q0 a q2 es:

- Para la transición desde q0 a q2:

La expresión regular es la combinación de todas las posibles cadenas que llevan de q0 a q2. Esto incluye:

- Cualquier cantidad de b en q0, seguido por a, luego a y cualquier cantidad de a y b en q2.

La expresión regular final es:
(b*)a(a(b | a)*))

Leyes algebraicas

Hay una variedad de leyes algebraicas para las expresiones regulares cada ley afirma que las expresiones de dos formas distintas son equivalentes.

Existen algunas leyes que se aplican a las expresiones regulares pero no tienen su análoga en la aritmética, especialmente cuando se utiliza el operador de clausura, dichas leyes y propiedades serán expuestas a continuación.

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

La 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:

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

Leyes distributivas

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.