

Universidad Autónoma de Chiapas



Facultad de Contaduría y Administración, Campus I

Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software

Compiladores

Actividad 1 – Investigación y Ejemplos

Elaborado por:

Diego Arturo Anzá Díaz

6°M

Catedrático:

Dr. Luis Gutiérrez Alfaro

Tuxtla Gutiérrez, Chiapas

A día jueves 15 de agosto de 2024

Concepto de Expresión Regular

Una expresión regular es una secuencia de caracteres que define un patrón de búsqueda en texto. Se utilizan para buscar y manipular cadenas de texto en base a patrones específicos. Las expresiones regulares son ampliamente utilizadas en programación para validaciones de datos, búsqueda y reemplazo de texto, y procesamiento de cadenas.

Ejemplo: La expresión regular `^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$` se usa para validar direcciones de correo electrónico.

I. Tipos de Operadores de Expresiones Regulares

Los operadores en expresiones regulares permiten la creación de patrones de búsqueda complejos. Algunos de los tipos de operadores más comunes son:

1° Concatenación: Representa la secuencia de caracteres.

- Ejemplo: La expresión `abc` coincide con la secuencia exacta `"abc"`.

2° Unión (Alternancia) |: Permite que el patrón coincida con una de varias alternativas.

- Ejemplo: La expresión `a|b` coincide con `"a"` o `"b"`.

3° Cierre de Kleene *: Indica que el carácter o subexpresión anterior puede aparecer cero o más veces.

- Ejemplo: La expresión `a*` coincide con `""`, `"a"`, `"aa"`, `"aaa"`, etc.

4° Cierre Positivo +: Indica que el carácter o subexpresión anterior debe aparecer al menos una vez.

- Ejemplo: La expresión `a+` coincide con `"a"`, `"aa"`, `"aaa"`, etc., pero no con `""`.

5° Cierre Opcional ?: Indica que el carácter o subexpresión anterior puede aparecer una vez o no aparecer.

- Ejemplo: La expresión $a?$ coincide con "" o "a".

II. Proceso de Conversión de DFA a Expresiones Regulares

La conversión de un Autómata Finito Determinista (DFA) a una expresión regular implica encontrar una expresión que describa el lenguaje aceptado por el DFA. El proceso general puede ser descrito como:

1° Construir una matriz de transiciones para el DFA, donde cada entrada indica las transiciones posibles entre los estados.

2° Eliminar los estados intermedios del DFA uno por uno, reemplazándolos por las correspondientes expresiones regulares que describen las transiciones directas entre los estados restantes.

3° Simplificación: Al final, quedará una expresión regular que describe las transiciones desde el estado inicial al estado final, que representa el lenguaje aceptado por el DFA.

Ejemplo: Si un DFA tiene dos estados, q_0 y q_1 , donde q_0 tiene una transición a q_1 con el símbolo 'a', y q_1 tiene una transición a sí mismo con 'b', la expresión regular correspondiente sería ab^* .

III. Leyes Algebraicas de Expresiones Regulares

Las expresiones regulares obedecen ciertas leyes algebraicas que permiten la manipulación y simplificación de las mismas. Algunas de estas leyes incluyen:

1° Ley de Idempotencia:

- $R+R=RR + R = RR+R=R$
- Ejemplo: $a + a$ es equivalente a a .

2° Ley de Asociatividad:

- $(R_1+R_2)+R_3=R_1+(R_2+R_3)$
 $(R_1 + R_2) + R_3 = R_1 + (R_2 + R_3)$
 $R_3)(R_1+R_2)+R_3=R_1+(R_2+R_3)$
- Ejemplo: $(a + b) + c$ es equivalente a $a + (b + c)$.

3° Ley de Conmutatividad (para la unión):

- $R_1+R_2=R_2+R_1$
 $R_1 + R_2 = R_2 + R_1$
 $R_1+R_2=R_2+R_1$
- Ejemplo: $a + b$ es equivalente a $b + a$.

4° Ley de Distribución:

- $R_1(R_2+R_3)=R_1R_2+R_1R_3$
 $R_1(R_2 + R_3) = R_1R_2 + R_1R_3$
 $R_1R_3)R_1(R_2+R_3)=R_1R_2+R_1R_3$
- Ejemplo: $a(b + c)$ es equivalente a $ab + ac$.

5° Ley de Neutro:

- $R+\emptyset=RR + \emptyset = RR+\emptyset=R$ y $R\emptyset=\emptyset R \emptyset = \emptyset$
 $R\emptyset=\emptyset$
- Ejemplo: $a + \emptyset$ es equivalente a a , y $a\emptyset$ es equivalente a \emptyset .

Estas leyes permiten la transformación y simplificación de expresiones regulares en formas más manejables o equivalentes.