Universidad Autónoma De Chiapas

Facultad De Contaduría y Administración

Licenciatura En Ingeniería En Desarrollo y Tecnologías De Software

Alumno
Daniel Hernandez Castillo
A221761

Docente
DR. Luis Gutierrez Alfaro

Unidad de competencia
Copiladores

Actividad
Actividad I

6to semestre grupo M

Fecha
15/08/2024

Tuxtla Gutierrez, Chiapas.

Definir los siguientes conceptos y de ejemplo de cada uno de los incisos de I, II, III.

Definir los conceptos de expresión regular

1. Explica los tipos de operadores de expresiones regulares

Coincidencia básica:

. : Coincide con cualquier carácter excepto un salto de línea.

^ : Coincide con el inicio de una cadena.

$: Coincide con el final de una cadena.

Agrupación y rango:

[] : Define un conjunto de caracteres. Por ejemplo, [a-z] coincide con cualquier letra minúscula.

() : Agrupa partes de una expresión para aplicar operadores a todo el grupo.

Repetición:

* : Coincide con 0 o más repeticiones del patrón anterior.

+ : Coincide con 1 o más repeticiones del patrón anterior.

? : Coincide con 0 o 1 repetición del patrón anterior.

{n} : Coincide exactamente con n repeticiones del patrón anterior.

{n,} : Coincide con n o más repeticiones.

{n,m} : Coincide con entre n y m repeticiones.

Alternancia y escape:

| : Alternancia lógica (OR). Por ejemplo, a|b coincide con a o b.

\ : Escapa caracteres especiales para tratarlos como literales. Por ejemplo, \. coincide con un punto literal.

Clases de caracteres predefinidas:

\d : Coincide con cualquier dígito (equivalente a [0-9]).

\D : Coincide con cualquier carácter que no sea un dígito.

\w : Coincide con cualquier carácter de palabra (letras, dígitos y guion bajo).

\W : Coincide con cualquier carácter que no sea de palabra.

\s : Coincide con cualquier espacio en blanco (espacio, tabulación, etc.).

\S : Coincide con cualquier carácter que no sea un espacio en blanco.

Ejemplo: Cierre de Kleene

Descripción: El cierre de Kleene representa cero o más repeticiones de una expresión.

- Expresión Regular: (a^*)
- Interpretación: Acepta las cadenas “”, “a”, “aa”, “aaa”, etc.

2. Explicar el proceso de conversión de DFA a expresiones regulares.

Primero, se simplifica el DFA eliminando estados innecesarios y añadiendo un nuevo estado inicial y uno final con transiciones vacías. Luego, se eliminan los estados intermedios uno por uno, ajustando las transiciones para que las rutas que pasaban por estos estados se reemplacen por transiciones directas entre los estados restantes.

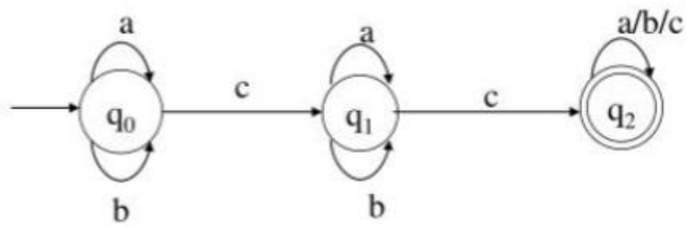
Finalmente, una vez eliminados todos los estados intermedios, la expresión regular resultante describe todas las posibles cadenas que el DFA puede aceptar, representando así el mismo lenguaje que el DFA original.

Ejemplo de DFA

Q={q0,q1,q2}

Σ={a,b,c}

Comienza desde q0



	a	b	c
q_0	q_0	q_0	q_1
q_1	q_1	q_1	q_2
q_2	q_2	q_2	q_2

3. Explicar leyes algebraicas de expresiones regulares

Las leyes algebraicas de las expresiones regulares son herramientas fundamentales que permiten manipular y simplificar expresiones regulares de manera sistemática y eficiente.

Ley Conmutativa para la Unión, Permite reordenar los términos en una unión sin cambiar el resultado. Esto es útil para simplificar y reorganizar expresiones regulares.

Ley Asociativa para la Unión, Facilita la agrupación de términos en una unión, permitiendo que se combinen de diferentes maneras sin afectar el resultado. Esto es útil para simplificar expresiones complejas.

Ley Asociativa para la Concatenación, Permite agrupar términos en una concatenación de diferentes maneras sin cambiar el resultado. Esto es útil para simplificar y reorganizar expresiones regulares.

Elemento Identidad y Elemento Nulo, Ayuda a simplificar expresiones eliminando elementos que no afectan el resultado. La cadena vacía actúa como identidad en la concatenación, y el conjunto vacío actúa como identidad en la unión.

Leyes Distributivas, Permiten distribuir la concatenación sobre la unión, facilitando la simplificación y manipulación de expresiones regulares.

Ley Idempotente para la Unión, Simplifica expresiones eliminando duplicados en una unión. Esto es útil para reducir la redundancia en las expresiones regulares.

Ejemplo sobre la ley de Idempotente para la Unión

Supongamos que tenemos un lenguaje:

(L = {a, b})

La unión de este lenguaje consigo mismo se puede expresar como:

(L + L)

Cálculo

({a, b} + {a, b} = {a, b})

Resultado

La unión de (L) consigo mismo no añade nuevos elementos, por lo que el resultado es simplemente (L):

(L + L = {a, b})

Este ejemplo muestra que unir un lenguaje consigo mismo no cambia el conjunto de elementos, lo que simplifica la expresión.