UNIVERSIDAD AUTÓNOMA DE CHIAPAS

Facultad de Contaduría y Administración, Campus I

Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software

Materia:
Compiladores

Unidad de competencia:
Analizador Léxico

Actividad:
Actividad I.- Investigación y Ejemplos.

Nombre del alumno:
Oscar Abel Torres Gómez

Nombre del profesor:
Dr. Luis Gutiérrez Alfaro

Lugar:
Tuxtla Gutiérrez, Chiapas

Fecha:
15 de agosto del 2024

Definir el concepto de expresión regular

Las expresiones regulares son un equivalente algebraico para un autómata. Utilizado en muchos lugares como un lenguaje para describir patrones en texto que son sencillos pero muy útiles. Pueden definir exactamente los mismos lenguajes que los autómatas pueden describir: Lenguajes regulares. Además, ofrecen algo que los autómatas no: Manera declarativa de expresar las cadenas que queremos aceptar.

Sirven como lenguaje de entrada a muchos sistemas que procesan cadenas tales como:

• Comandos de búsqueda, e.g., grep de UNIX

• Sistemas de formateo de texto: Usan notación de tipo expresión regular para describir patrones

• Convierte la expresión regular a un DFA o un NFA y simula el autómata en el archivo de búsqueda

• Generadores de analizadores-léxicos. Como Lex o Flex.

• Los analizadores léxicos son parte de un compilador. Dividen el programa fuente en unidades lógicas (tokens). Tokens como while, números, signos (+, −, <, etc.)

• Produce un DFA que reconoce el token.

De forma más precisa, Las expresiones regulares denotan lenguajes. Por ejemplo, la expresión regular:

$01^* + 10^*$ denota todas las cadenas que son o un 0 seguido de cualquier cantidad de 1's o un 1 seguida de cualquier cantidad de 0's.

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

Comúnmente existen tres operadores de las expresiones regulares las cuales se describen a continuación:

- UNION:

  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.

- CONCATENACION:

  La concatenación de lenguajes se denota como LM o LM e.g. L = 001,10,111, M = ,001, entonces la concatenación será LM =001,10,111,001001,10001,111001.

- CERRADURA:

  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. sí L = 01, L son todas las cadenas con 0s y 1s. Si L = 011, entonces L son todas las cadenas de 0s y 1s tal que los 1s están en pareja. Para calcular L se debe calcular Li para cada i y tomar la unión de todos estos lenguajes. Li tiene 2i elementos. Aunque cada Li es nito, la unión del número de términos de Li es en general un conjunto innato e.g. = o 0=. Generalizando, para todo i mayor o igual que uno, i es el conjunto vaco, no se puede seleccionar ninguna cadena del conjunto vaco.

EJEMPLOS:

1.Si E y F son expresiones regulares, entonces E + F también lo es denotando la unión de L(E) y L(F). L (E +F) = L(E) L(F).

2.Si E y F son expresiones regulares, entonces EF también lo es denotando la concatenación de L(E) y L(F). L(EF) = L(E)L(F).

3.Si E es una expresión regular, entonces E también lo es y denota la cerradura de L(E). Ósea L(E) =(L(E)) *

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

Si se necesita demostrar que, para cada expresión regular, existe un autómata nito que acepta el mismo lenguaje. Se selecciona el autómata más apto NFA. Si, por el contrario, se necesita demostrar que por cada autómata nito, hay una expresión regular de siendo su lenguaje se selecciona el autómata con mayores restricciones: DFA. Los lenguajes aceptados por DFA, NFA, -NFA, RE son llamados lenguajes regulares.

Para convertir expresiones regulares a-NFA se realizan pruebas por inducción en los diferentes operadores (+, concatenación, *) en la expresión regular. Siempre se construye un autómata de una forma especial. Se mostrará que un NFA con transiciones puede aceptar el lenguaje de una RE. Después, se mostrará que un RE puede describir el lenguaje de un DFA (la misma construcción funciona para un NFA). Los lenguajes aceptados por DFA, NFA, -NFA, RE son llamados lenguajes regulares.

Teorema 1 Si L=L(A) para algún DFA A, entonces existe una expresión regular R tal que L = L(R).

Prueba: Suponiendo que A tiene estados {1, 2, n}, n infinito. Tratemos de construir una colección de RE que describan progresivamente conjuntos de rutas del diagrama de transiciones de A

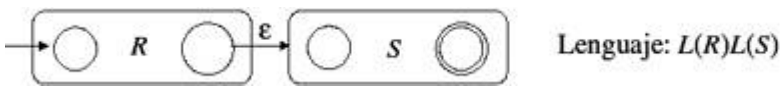
III.- Explicar leyes algebraicas de expresiones regulares.

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= 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. Por ejemplo, 0 es la identidad de la suma, ya que 0+x = x+0 = x, la identidad para la multiplicación es 1 debido a que 1x
= x1 = 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 x = x 0 = 0. No hay aniquilador para la suma.

Estas leyes las utilizamos para hacer simplificaciones: 0 es la identidad para la union: 0+L = L+0 = L. es la identidad para la concatenación: L = L = L. es el aniquilador para la concatenación: 0L=L0 =0.

Ley distributiva

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= x (aunque para algunos valores aplica como 0 + 0 = 0). En general la multiplicacion tampoco es idempotente: x *x = x. La unión e intersección son ejemplos comunes de operadores idempotentes. La ley idempotente para la unión: L + L = L.

Leyes relacionadas con la propiedad de cerradura

Las leyes involucradas con la propiedad de cerradura:

- (L*) * =L* (Idempotencia para la cerradura)
- 0* = e
- E*=e
- L+ =LL* =L*, LL+ se define como L+LL+LLL+ …
- L*= L++eL+LL+LLL+…
- LL* =Le+LL+LLL+LLLL+
- L *=L++ e
- L? = +L

Ejemplo:

Probar que (L+M*) = (L*M) Es necesario probar que las cadenas que estan en (L *+ M*) * también estan en (L*M*), y probamos que las cadenas que estan en (LM)* también estan 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 guia para concluir sobre los lenguajes. Podemos analizar el lenguaje que nos describe: (a +b) * y analizar el lenguaje que nos describe: (a*b*).

Bibliografía:

Regulares, 1. Expresiones. (s/f). Propedéutico: Teoría de Autómatas y Lenguajes Formales Expresiones regulares y lenguajes. Inaoep.mx. Recuperado el 15 de agosto de 2024, de https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Automatas/03_Automatas_ExpresionesRegularesLenguajes/CAPTUL1.PDF