Universidad Autónoma de Chiapas
Facultad de contaduría y administración C-I

Carrera:
Lic. En ing en desarrollo y tecnologías de software.

Materia:
Compiladores.

Catedrático:
Mtro. Luis Gutiérrez Alfaro.

Nombre del alumno:
González Aguilar Eduardo - A211154

Semestre: 6. **Grupo:** M.

Nombre de la actividad:
Conceptos del analizador léxico.

Fecha de entrega:
26/08/2023.

Introducción.

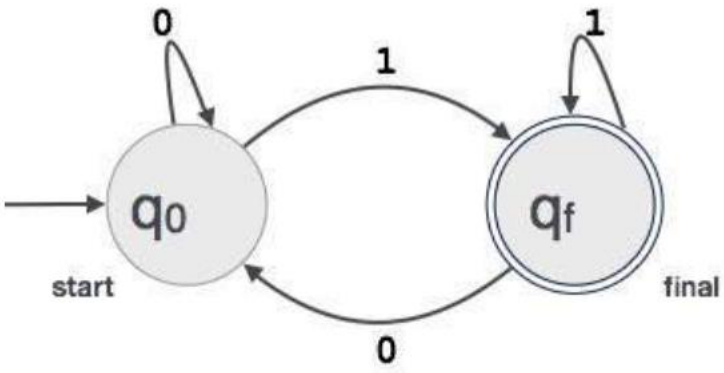
En esta actividad de investigación de la materia de compiladores se desarrollará el aprendizaje sobre el tema de expresiones regulares y como se desenvuelve este sobre la parte del proyecto de desarrollo de software de bajo nivel.

Desarrollo.

- **1.1 Expresiones regulares.**

El analizador léxico debe analizar e identificar sólo un conjunto finito de cadena válida/token/lexema que pertenecen al lenguaje. Busca el modelo definido por las normas del lenguaje. Las expresiones regulares tienen la capacidad de expresar finitos idiomas definiendo un modelo finito de cadenas de símbolos.

Ejemplo: suponemos FA tres dígitos acepta cualquier valor binario que termina en dígito 1. FA = {Q(q_0, q_f), Σ(0,1), q_0, q_f, δ}



1.2 Autómatas.

1.2.1 Autómatas no determinísticos.

- La transición desde un estado puede tener múltiples destinos. Por eso se le llama no determinista.
- Permite transiciones con cadenas vacías.
- No siempre se permite el uso de back tracking.
- Requiere menos espacio.
- Una cadena es aceptada si solo una de todas sus posibles transiciones es hacia un estado final.

1.2.2 Autómatas determinísticos.

- La transición desde un estado puede tener como destino un único estado. Por eso se llama determinista.
- No se aceptan transiciones con cadenas vacías.
- Se permite el uso de back tracking
- Requiere mas espacio.
- Una cadena es aceptada si su transición es hacia un estado final.

- **1.3 Matrices de transición.**

1. Todos los elementos de la matriz son no negativos, por lo tanto, pij > 0. 2.
2. La suma de los elementos de cada fila es igual a 1, por lo tanto, Spij = 1 para todo i.

   Si denominamos A como la matriz de probabilidades de transición con un horizonte de tiempo dado, esta se puede representar de manera general como (tabla 2):

Tabla 2. Matriz de probabilidades de transición

Categoría inicial	*Categoría después de transición*				
	1	*2*	*3*	...	*j (default)*
1	P11	P12	P13		P1j
2	P21	P22	P23		P2j
...					...
...					...
i - 1	P(i – 1)1	P(i – 1)2	P(i – 1)3	...	P(i – 1)j
i (default)	0	0		...	1

- **1.4 Tabla de símbolos.**

TABLA DE SÍMBOLOS

Operador	Significado
!	Negación
+	Suma
-	Resto
*	Multiplicación
/	División
%	Módulo
<	Menor
<=	Menor igual
>	Mayor
>=	Mayor igual
!=	Diferente
&&	Conjunción Lógica(Y)
||	Disyunción Lógica(O)
==	Igualdad

- **1.5 Diferentes herramientas automáticas para generar analizadores léxicos.**

Herramienta	Descripción
Bison	Generador de Analizadores Sintácticos Ascendentes tipo YACC
COCO/R	Generador de Analizadores Léxicos y Sintácticos Descendentes Recursivos
Flex	Generador de Analizadores Léxicos tipo Lex
Lex	Generador de Analizadores Léxicos
SDGLL1	Sistema Detector de Gramáticas LL(1)
TS 2006	Tipo abstracto de datos Tabla de Símbolos de uso sencillo (beta0.4)
TS	Tipo abstracto de datos Tabla de Símbolos
TS-OO	Tipo abstracto de datos Tabla de Símbolos
YACC	Generador de Analizadores Sintácticos Ascendentes LR(1)

- **1.6 Gramática libre de Contexto**

Una gramática libre de contexto (GLC) es una descripción estructural precisa de un lenguaje. Formalmente es una tupla G=, donde

• **Vn** es el conjunto finito de símbolos no terminales

• **Vt** es el conjunto finito de símbolos terminales ($Vn \cap Vt = \emptyset$)

• **P** es el conjunto finito de producciones que se pueden ver como relaciones definidas en $Vn \times (Vn \cup Vt^*)$

• **S** es el símbolo inicial de la gramática.

La ambigüedad significa que una expresión del lenguaje puede tener más de una interpretación, lo cual no está permitido.

Una gramática libre de contexto es una gramática formal en la que cada regla de producción es de la forma $V \rightarrow w$, donde V es un símbolo no terminal y w es una cadena de terminales y/o no terminales[1]. Las gramáticas libres de contexto permiten describir la mayoría de los lenguajes de programación[1].

Las gramáticas de lenguajes compiladores son especificaciones sintácticas y precisas de lenguajes de programación. Se pueden describir la sintaxis de las construcciones de los lenguajes de programación por medio de gramáticas de contexto libre o notación BNF (Backus-Naur Form). Las gramáticas ofrecen ventajas significativas a los diseñadores de lenguajes y a los desarrolladores de compiladores. A partir de una gramática se puede generar automáticamente un analizador sintáctico.

Eliminación de la ambigüedad: podemos eliminar la ambigüedad únicamente sobre la base de las siguientes dos propiedades:

1. Precedencia:
si se utilizan diferentes operadores, consideraremos la precedencia de los operadores. Las tres características importantes son:

1. El nivel al que está presente la producción denota la prioridad del operador utilizado.
2. La producción a **niveles más altos** tendrá **operadores con menor prioridad** . En el árbol de análisis, los Nodes que se encuentran en los niveles superiores o cerca del Node raíz contendrán los operadores de menor prioridad.
3. La producción en **los niveles inferiores** tendrá operadores con **mayor prioridad** . En el árbol de análisis, los Nodes que se encuentran en niveles más bajos o cerca de los Nodes hoja contendrán los operadores de mayor prioridad.

2. Asociatividad:
si los mismos operadores de precedencia están en producción, entonces tendremos que considerar la asociatividad.

- Si la asociatividad es de izquierda a derecha, entonces tenemos que provocar una recursión a la izquierda en la producción. El árbol de análisis también se dejará recursivo y crecerá en el lado izquierdo.
  +, -, *, / son operadores asociativos a la izquierda.
- Si la asociatividad es de derecha a izquierda, entonces tenemos que provocar la recursión a la derecha en las producciones. El árbol de análisis también será recursivo a la derecha y crecerá en el lado derecho.
  ^ es un operador asociativo derecho.
-

Ejemplo 1 – Considere la gramática ambigua

E -> EE | id

El lenguaje en la gramática contendrá { id, id-id, id-id-id, ….}

Digamos que queremos derivar la string **id-id-id.** Consideremos un solo valor de id=3 para obtener más información. El resultado debería ser:

3-3-3 =-3

Dado que los mismos operadores de prioridad, debemos considerar la asociatividad que es de izquierda a derecha.

Una gramática libre de contexto es una descripción estructural precisa de un lenguaje. Formalmente es una tupla G=<Vn, Vt, P, S>, donde Vn es el conjunto finito de símbolos no terminales y Vt es el conjunto finito de símbolos terminales ($Vn \cap Vt = \emptyset$). Las producciones tienen la forma $A\rightarrow$ donde $A \in Vn$ y es una expresión ya sea compuesta por símbolos no terminales como terminales ($\in (Vt \cup Vn)^*$) ó la cadena nula Ɛ. Una producción puede verse como una regla de reescritura que indica cómo reemplazar símbolos no terminales por la expresión correspondiente en la parte derecha de la producción. Así partiendo de algún $A \in Vn$, se pueden aplicar las reglas de P hasta alcanzar a una expresión compuesta únicamente por símbolos terminales. A ésta expresión se le denomina frase o lexema. A las expresiones formadas por símbolos terminales y no terminales se les denomina formas de frase[12].

Conclusión.

Las expresiones regulares son muchos mas extensas de lo que imaginaba, son gramáticas que bien pueden ser tanto determinísticas como no determinísticas y cada uno de estas tiene su propia función y características diferentes de las cuales tienen puntos importantes para su desarrollo.

Fuente de información.

Autor desconocido, (26 de agosto del 2023), Compilador diseño – Expresiones regulares". Compilador DiseÃ±o - Expresiones Regulares | Tutorialspoint

Ricardo G, (13 de Marzo del 2017), "Autómatas finitos deterministas y no deterministas". Automatas Finitos Deterministas Y No Deterministas | RicardoGeek

Tamara-Ayus A, Raul A, Ermilson V, (22 de mayo del 2012), "Matrices de transcision en el análisis del riesgo crediticio como elemento fundamental en el calculo de de la perdida esperada en una institución financiera colombiana", pp. 110. v11n20a09.pdf (scielo.org.co)

Autor desconocido, (26 de agosto del 2023), "Gramatticas libres de contexto", Pp. 2 y 5. Gramáticas libres de contexto (buap.mx)

Rudeus Greyrat, (5 de Julio del 2022), "Eliminación de la ambigüedad". Eliminación de la ambigüedad (Conversión de una gramática ambigua en una gramática inequívoca) – Barcelona Geeks