



UNIVERSIDAD AUTONOMA DE CHIAPAS

FACULTAD DE CONTADURIA Y

ADMINISTRACIONCAMPUS 1

MATERIA;
COMPILADORES

DOCENTE:
LUIS GUTIEREZ ALFARO

TAREA:
ACTIVIDAD
1.3

ALUMNO:
CARLOS ANTONIO AGUILAR RAMOS

GRADO Y

GRUPO:

6M

LUGAR Y FECHA:
TUXTLA GUTIERREZ CHIAPAS, 30 DE ENERO DE
2023

Expresiones regulares.

Son una serie de caracteres que forman un patrón, normalmente representativo de otro grupo de caracteres mayor, de tal forma que podemos comparar el patrón con otro conjunto de caracteres para ver las coincidencias.

Las expresiones regulares están disponibles en casi cualquier lenguaje de programación, pero aunque su sintaxis es relativamente uniforme, cada lenguaje usa su propio dialecto.

Autómatas.

Es un modelo computacional que consiste en un conjunto de estados bien definidos, un estado inicial, un alfabeto de entrada y una función de transición.

En un autómata, un estado es la representación de su condición en un instante dado. El autómata comienza en el estado inicial con un conjunto de símbolos; su paso de un estado a otro se efectúa a través de la función de transición, la cual, partiendo del estado actual y un conjunto de símbolos de entrada, lo lleva al nuevo estado correspondiente.

Ejemplos:

Un ejemplo de autómata en la vida cotidiana es un elevador, ya que es capaz de memorizar las diferentes llamadas de cada piso y optimizar sus ascensos y descensos.

Autómatas no determinísticos.

Es un autómata finito en donde δ no es necesariamente una función de transición, es decir, que para cada par (estado actual y símbolo de entrada) le corresponde cero, uno, dos o más estados siguientes. Normalmente la relación de transición para un AFND se denota con Δ .

Autómatas determinísticos.

El termino determinista se refiere al hecho de que para cada entrada hay uno y solo un estado al que el autómata puede cambiar.

Es un autómata finito en donde δ (delta) es una función de transición, es decir, que para cada par (estado actual y símbolo de entrada) le corresponde un único estado siguiente.

- ✓ Un autómata determinista consiste de:
- ✓ Un estado finito de estados Q .
- ✓ Un estado finito de símbolos de entrada Σ .
- ✓ Una función de transición δ que toma como argumentos un estado y un símbolo de entrada y regresa un estado.
- ✓ Un estado inicial perteneciente a Q .
- ✓ Un conjunto de estados finales F , correspondientes a un subconjunto de Q .

Matrices de transición

una matriz de transición define la probabilidad de cambio de estado a partir del tiempo actual (t_1) hasta un tiempo futuro ($t+1$). En un modelo de Markov la probabilidad para llegar a un estado futuro solo depende del estado actual

Tabla de símbolos

La tabla de símbolos (TS) es la estructura utilizada por el compilador para almacenar los atributos asociados a los símbolos que se utilizan en un lenguaje de programación.

Es importante debido a que todas las fases del compilador, traductor o intérprete se apoyan en ella para escribir o para obtener información, las de análisis para insertar y actualizar y las de síntesis para obtener la información con la que generar el código.

Diferentes herramientas automáticas para generar analizadores léxicos

Herramientas tradicionales.

Lex/yacc: es un programa para generar analizadores léxicos, se utiliza comúnmente con el programa yacc que se utiliza para generar análisis sintáctico.

Yacc: es una herramienta general para describir la entrada a un programa de computadora. El usuario especifica las estructuras de su entrada, junto con el código que puede invocarse como cada estructura como se reconoce y yacc devuelve una subrutina que controla el proceso de entrada, con frecuencia, es conveniente y apropiado que la mayor parte del flujo de control en la aplicación del usuario al cargo de este subprograma.

Flex: herramienta que implementa un analizador léxico. Es una herramienta para generar escáneres: programas que reconocen patrones léxicos en un texto. Flex lee los ficheros de entrada dados, o la entrada estándar si no se le ha indicado ningún nombre de fichero, con la descripción de un escáner a generar.

Bison: herramienta que implementa un analizador sintáctico. Es un generador de analizadores sintácticos de propósito general perteneciente al proyecto GNU disponible para prácticamente todos los sistemas operativos. Bison convierte la descripción formal de un lenguaje, escrita como una gramática libre de contexto LALR, en un programa en C, C++, o Java que realiza análisis sintáctico.

Herramientas de nueva generación.

ANTLR (ANother Tool for Language Recognition – otra herramienta para reconocimiento de lenguajes): es una herramienta creada principalmente por Terence Parr, que opera sobre lenguajes, proporcionando un marco para construir reconocedores (parsers), intérpretes, compiladores y traductores de lenguajes a partir de las descripciones gramaticales de los mismos (conteniendo acciones semánticas a realizarse en varios lenguajes de programación).

JavaCC (Java Compiler Compiler): es un generador de analizadores sintácticos de código abierto para el lenguaje de programación Java, es similar a Yacc en que genera un parser para una gramática presentada en notación BNF, con la diferencia de que la salida es en código Java y genera analizadores descendentes.

COCO/JAVA: es una herramienta de distribución de Java para Linux, es un generador de compiladores que a partir de la descripción del lenguaje mediante una gramática LL genera un analizador sintáctico y un analizador léxico para dicho lenguaje o sea es un meta compilador.

JELL: es un generador de analizadores sintácticos que genera analizadores descendentes a partir de gramáticas LL, también es un meta compilador

Bibliografía.

<https://docs.vmware.com/es/vRealize-Log-Insight/8.0/com.vmware.log-insight.user.doc/GUID88B2952D-3112-46BC-B126-84C9BF38B6D2.html>

<https://core.ac.uk/download/154797605.pdf>

https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2_M4_U5_T2.pdf
<http://cursocompiladoresuaeh.blogspot.com/2010/11/unidad-iv-herramientas-basicas-para.html>