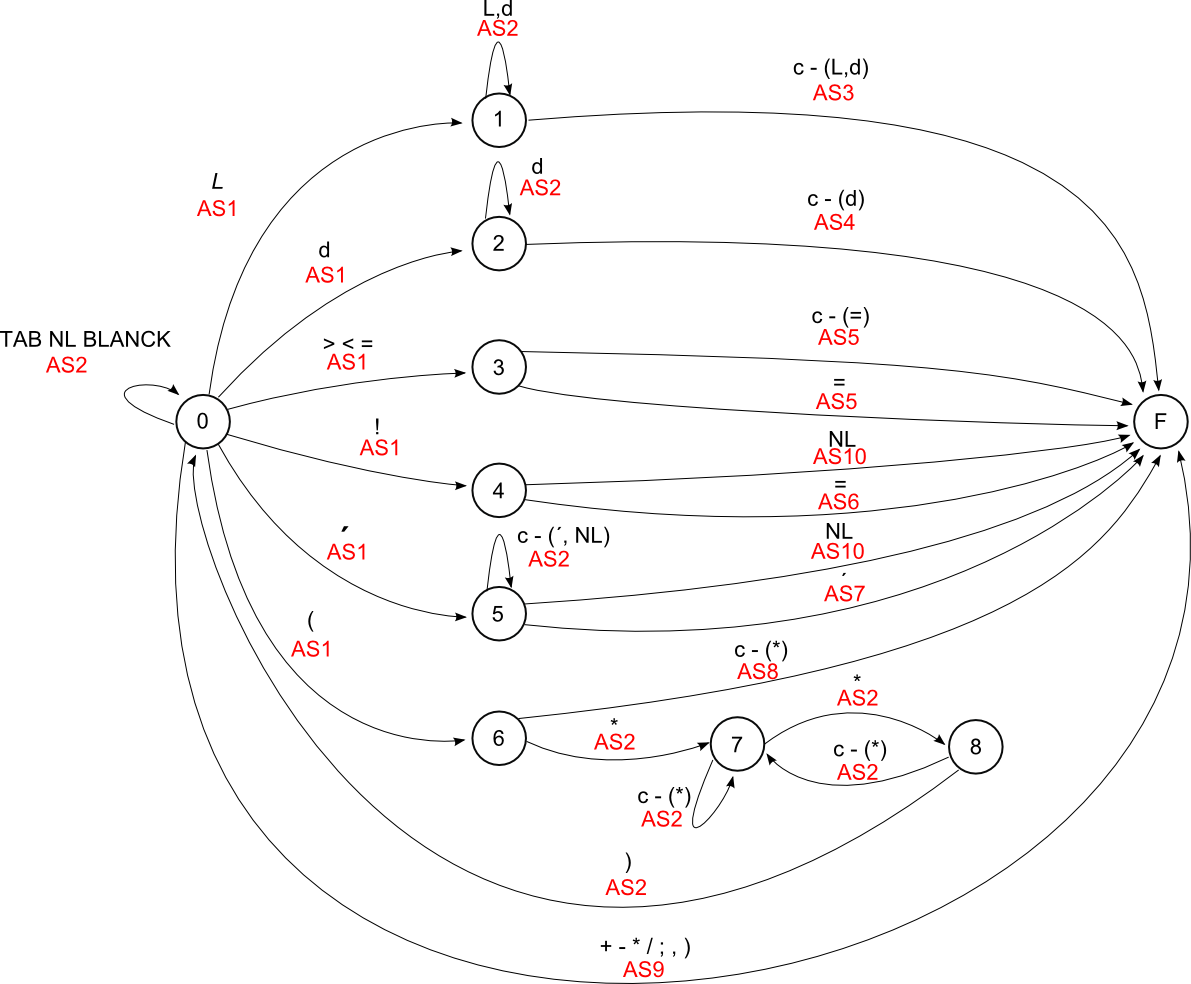
Informe Diseño de compiladores

El siguiente describirá y explicara la creación de analizador léxico como parte de un trabajo practico propuesto por la catedra de Diseño de Compiladores.  
  
**¿Qué es un analizador léxico?**   
Un analizador léxico (AL) es un componente de un compilador de lenguaje de programación, que se encarga de analizar y encontrar errores léxicos en un código fuente dado. El AL buscará palabras reservadas propias del lenguaje, además de detectar identificadores, constantes, ---------------. A estos elementos se los denominan *Tokens.*

**¿Cómo funciona?**  
El AL funciona en conjunto con otro componente llamado analizador sintáctico (AS). En éste caso el sintáctico es una herramienta que ya se encuentra desarrollada y se denomina YACC. Éste es quien pone en funcionamiento al procesador léxico solicitándole un token.  
El AL trabaja recorriendo un archivo de texto caracter a caracter procesándolos en un autómata finito. Cada vez que se arribe a un estado final el analizador habrá encontrado un token que deberá ser entregado al procesador sintáctico.  
Es importante aclarar que el AL se pone en funcionamiento solo cada vez que el AS le solicita un token y que una vez encontrado y entregado, se detiene hasta nuevo pedido.  
Cada token está compuesto por un número que lo identifica y un valor de posición que indica en que parte de la tabla de símbolos está almacenado.

**Desarrollo de la aplicación**Como primer medida el grupo comenzó por la creación del autómata que describe los estados por los que atraviesa el AL cada vez que lee un carácter y hasta encontrar un token. (Imagen 1)  
  
L = carácter letra  
d = carácter digito   
TAB = carácter de tabulación  
NL = carácter de Nueva Línea  
BLANK = carácter de espacio  
c – () = cualquier carácter menos los que se encuentran especificados entre paréntesis  
Cualquier otro símbolo o carácter se representa así mismo.

Los escritos en rojos hacen referencia a las Acciones Semánticas que se ejecutan en cada transición de estados.



*Imagen 1 correspondiente al autómata que representa el Analizador Léxico*

Matriz de transición de estados

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Carácter** | | | | | | | | | | | | | | | | | |
| **Estado** | L | d | > | < | = | ! | ‘ | ( | ) | + | - | \* | / | ; | , | NL | TAB | BLANCK |
| 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 6 | F | F | F | F | F | F | F | 0 | 0 | 0 |
| 1 | 1 | 1 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 2 | F | 2 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 3 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 4 | **X** | **X** | **X** | **X** | F | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | F | **X** | **X** |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | F | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | F | 5 | 5 |
| 6 | F | F | F | F | F | F | F | F | F | F | F | 7 | F | F | F | F | F | F |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 8 | 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 0 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| F |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Los estados marcados con **X** son estados de error.

Matriz de Acciones semánticas

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Carácter** | | | | | | | | | | | | | | | | | |
| **Estado** | L | d | > | < | = | ! | ‘ | ( | ) | + | - | \* | / | ; | , | NL | TAB | BLANCK |
| 0 | AS1 | AS1 | AS1 | AS1 | AS1 | AS1 | AS1 | AS1 | AS9 | AS9 | AS9 | AS9 | AS9 | AS9 | AS9 | AS2 | AS2 | AS2 |
| 1 | AS2 | AS2 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 | AS3 |
| 2 | AS4 | AS2 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 | AS4 |
| 3 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 | AS5 |
| 4 |  |  |  |  | AS6 |  |  |  |  |  |  |  |  |  |  | AS10 |  |  |
| 5 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS7 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS10 | AS2 | AS2 |
| 6 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 | AS2 | AS8 | AS8 | AS8 | AS8 | AS8 | AS8 |
| 7 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 |
| 8 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 |  | AS2 | AS2 | AS2 | AS2 | AS2 | AS2 |
| F |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Una vez delineado el autómata y las matrices que lo representan se comenzó con la implementación. El grupo eligió el lenguaje Java para desarrollarse.