Lenguajes y Compiladores

Implementación de FLEX

Fast LEXical Analizer

FLEX

Introducción

Está herramienta está escrita en lenguaje C y genera un analizador léxico para un lenguaje determinado.

A. Utiliza expresiones regulares (regex) para reconocer los lexemas del lenguaje.
B. Puede incluir código para ser invocado cuando esas expresiones son reconocidas
C. Puede incluir código auxiliar para el soporte del código de entrada

FLEX
Introducción

Flex lleva adelante el proceso de análisis léxico a través de la función especial propia:

yylex()

Y tiene variables reservadas que pueden interactuar con el analizador sintáctico :

int yylval

FLEX

Estructura – Secciones – Especificaciones Básicas

Un programa **FLEX** se organiza en 4 grandes secciones:

1. Definiciones
2. Conjuntos y expresiones regulares
3. Reglas
4. Código

FLEX

Estructura–Secciones–Especificaciones Básicas

Sección Definiciones:

En esta sección deberán incluir todas las declaraciones que se utilizarán en el código y deberán enmarcarse por los signos

%{ %}

Las declaraciones pueden ser:

- Includes
- Defines
- Variables Globales

FLEX

Estructura–Secciones–Especificaciones Básicas

Sección Declaraciones - Ejemplo:

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include "y.tab.h"
FILE  *yyin;
int yylval;
%}
```

FLEX

Estructura–Secciones–Especificaciones Básicas

Sección Conjuntos y Regex:

En esta sección deberá incluir todas las declaraciones de:

- **opciones** (%option)
- **conjuntos**
- **expresiones regulares**

Las expresiones regulares son patrones que tienen un lenguaje propio y describen el formato de un lexema.

Las expresiones regulares pueden definirse en este sector o en el sector de reglas.

FLEX

Estructura – Secciones – Especificaciones Básicas

Sección Conjuntos y Regex:

%option …….

```
DIGITO          [0-9]
LETRA           [a-zA-Z]

CONST_REAL      {DIGITO}+"."{DIGITO}+
CONST_INT       {DIGITO}+
ID              {LETRA}({LETRA}|{DIGITO}|_)*
```

FLEX

Estructura–Secciones–Especificaciones Básicas

Sección Definición de Reglas:

Esta sección deberá enmarcarse por los signos:

%% %%

Esta sección deberá incluir todas los patrones necesarios para llevar adelante el análisis léxico a través de acciones semánticas, su formato será:

patron acción semántica

FLEX

Estructura–Secciones–Especificaciones Básicas

Un patrón puede ser:

- un símbolo o conjunto de símbolos que se escribe entre comillas dobles
- el nombre de la regex definida en la sección anterior que se escribe entre llaves
- una nueva regex

FLEX

Estructura – Secciones – Especificaciones Básicas

Las **acciones** son ejecutadas cada vez que una regla es reconocida con el proceso de entrada.
Esas acciones pueden retornar valores, componentes léxicos (tokens) y pueden ejecutar algún codigo.

Una acción es una sentencia arbitraria escrita en el lenguaje que compila **flex**. Deben aparecer encerradas entre llaves **"{""}"** y finalizada con un **";"**

FLEX

Estructura – Secciones – Especificaciones Básicas

Sección Definición de Reglas : Ejemplo

```
%%
"define"                { return DEFINE;}
"enddefine"             { return ENDDEFINE;}
":="                    { return OP_AS;}
"+"                     { return OP_SUMA;}
{ID}                    { guardar_en_TS () ;return ID;}
{CONST_REAL}            { return CONST_INT;}
{DIGITO}+               { yylval = atoi( yytext ) ;return ENTERO;}
%%
```

FLEX

Estructura – Secciones – Especificaciones Básicas

Sección Código:

Esta última sección deberá incluir todo el código necesario para la ejecución del programa.

FLEX

Estructura – Secciones – Especificaciones Básicas


```
int main(int argc,char *argv[])
{
  if ((yyin = fopen(argv[1], "rt")) == NULL)
  {
        printf("\nNo se puede abrir el archivo: %s\n", argv[1]);
  }
  else
  {
        yylex();
  }
  fclose(yyin);
```

FLEX

Compilación

Todas las especificaciones vistas en los puntos anteriores deberán incluirse en un archivo de extensión “.l” , por ejemplo:

miLexico.l

Este archivo se compilará con el comando:

flex miLexico.l

FLEX
Compilación

Si el **resultado** de la compilación es **satisfactorio** FLEX generará el archivo:

- **lex.yy.c** (archivo que contiene el analizador léxico del lenguaje)