

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'COMPILADORES – 3CV7'. In the bottom-left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

COMPILADORES – 3CV7

# Práctica 4

Introducción a Bison

Oscar Eduardo López Cabagné  
2015070715

# Índice

Introducción	2
Desarrollo	3
Pruebas	5
Conclusión	8
Referencias	8

# Introducción

## ¿Qué es Bison?

Es un programa generador de analizadores sintácticos de propósito general perteneciente al proyecto GNU disponible para prácticamente todos los sistemas operativos, se usa normalmente acompañado de flex aunque los analizadores léxicos se pueden también obtener de otras formas.

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. Es utilizado para crear analizadores para muchos lenguajes, desde simples calculadoras hasta lenguajes complejos. Para utilizar Bison, es necesaria experiencia con la sintaxis usada para describir gramáticas.

Bison fue escrito en un principio por Robert Corbett; Richard Stallman lo hizo compatible con Yacc y Wilfred Hansen de la Carnegie Mellon University añadió soporte para literales multicaracter y otras características.

## Estructura de un programa Flex

El archivo de codificación de Yacc/Bison tiene cuatro secciones:

%{

Declaraciones C

%}

Declaraciones Yacc/Bison

%Declaración de Token

%%

Reglas de la gramática

%%

Código C

Las cadenas “%{”, “%}” sirve para delimita el encabezado que pudieran ser bibliotecas “%%” sirven para indicar cuales son las reglas Y por último va programación en C o llamadas a funciones en caso de ser necesario. %Token Se utiliza para definir los símbolos no terminales (tokens) de la gramática. %token NOMBRE\_TOKEN Por convenio, los nombres de los tokens se ponen en mayúsculas y se pueden agrupar varios token en una línea si tienen en mismo tipo.<sup>2</sup>

## Desarrollo

Esta práctica es una extensión de la anterior, pues, ahora no solo se reconocerá cada uno de los lexemas leídos, si no que se les asignará un valor y ahora el programa será capaz de realizar las acciones correspondientes.

A continuación, se muestran las secciones de código para cada uno de los requisitos de la práctica:

- Variables

```
{VAR}      {
            char *aux;
            aux=copia(yytext);
            printf("\tVariable %s\n",aux);
            yylval.cadena = aux;
            return (VARIABLE);
            }
```

- Números enteros

```
{NUM}      {
            printf("\tNumero entero %s\n",yytext);
            yylval.entero = atoi(yytext);
            return (ENTERO);
            }
```

- Números decimales

```
{DEC}      {
            printf("\tNumero decimal %s\n",yytext);
            yylval.decimal = atof(yytext);
            return (DECIMAL);
            }
```

- Módulo

```
"mod"{PARENTESISLEFT}({NUMERODECIMAL}|{NUMEROENTERO})", "({NUMEROENTERO}){PARENTESISRIGHT} {printf("Modulo");}
```

```
[Mm][Oo][Dd]    {  
    printf("\tModulo\n");  
    return (MO);  
}
```

- Operaciones matemáticas

```
"+"    {  
    printf("\tOperador mas\n");  
    return (yytext[0]);  
}  
  
"- "   {  
    printf("\tOperador menos\n");  
    return (yytext[0]);  
}  
  
"*"    {  
    printf("\tOperador multiplicar\n");  
    return (yytext[0]);  
}  
  
"/"    {  
    printf("\tOperador division\n");  
    return (yytext[0]);  
}  
  
[Pp][Oo][Ww]  {  
    printf("\tPotencia\n");  
    return (POW);  
}
```

## Pruebas

- Números enteros con y sin signo

```
5
    Numero entero 5
    Salto de linea
    resultado: 5
34
    Numero entero 34
    Salto de linea
    resultado: 34
-100
    Operador menos
    Numero entero 100
    Salto de linea
    resultado: -100
_
```

- Números decimales con y sin signo

```
0.05
    Numero decimal 0.05
    Salto de linea
    resultado: 0.050000
0.51
    Numero decimal 0.51
    Salto de linea
    resultado: 0.510000
-13.2
    Operador menos
    Numero decimal 13.2
    Salto de linea
    resultado: -13.200000
-3.1416
    Operador menos
    Numero decimal 3.1416
    Salto de linea
    resultado: -3.141600
_
```

- Operaciones Matemáticas

```
5+3
    Numero entero 5
    Operador mas
    Numero entero 3
    Salto de linea
        resultado: 8

7-10
    Numero entero 7
    Operador menos
    Numero entero 10
    Salto de linea
        resultado: -3

8*5
    Numero entero 8
    Operador multiplicar
    Numero entero 5
    Salto de linea
        resultado: 40

178/23
    Numero entero 178
    Operador division
    Numero entero 23
    Salto de linea
        resultado: 7
```

- Módulo

```
mod(5,3)
    Modulo
    Parentesis Izquierdo
    Numero entero 5
    coma
    Numero entero 3
    Parentesis Derecho
    Salto de linea
        resultado: 2

mod(85,3)
    Modulo
    Parentesis Izquierdo
    Numero entero 85
    coma
    Numero entero 3
    Parentesis Derecho
    Salto de linea
        resultado: 1
```

## Conclusión

Bison es un analizador semántico, que en conjunto con un analizador léxico como Flex, es capaz de leer una cadena de entrada, verificar su estructura, reconocer los datos obtenidos, asignarles un valor semántico y hasta realizar operaciones o acciones con ellos.

En conjunto, Bison y Flex pueden ser grandes complementos en muchas más aplicaciones de las que pensé en un principio. Sin embargo siempre será importante recordar que mientras Flex puede operar por sí solo, Bison siempre necesitará de un analizador léxico que le provea de tokens válidos.

## Referencias

- [1] *Repositori.uji.es*, 2021. [Online]. Available:  
[http://repositori.uji.es/xmlui/bitstream/handle/10234/5998/Primera\\_Practica\\_IS17\\_Curso\\_06\\_07.pdf?sequence=1&isAllowed=y#:~:text=FLEX%20es%20el%20analizador%20de,para%20obtener%20un%20programa%20ejecutable](http://repositori.uji.es/xmlui/bitstream/handle/10234/5998/Primera_Practica_IS17_Curso_06_07.pdf?sequence=1&isAllowed=y#:~:text=FLEX%20es%20el%20analizador%20de,para%20obtener%20un%20programa%20ejecutable).  
[Accessed: 23- Jan- 2021].
- [2] *Cartagena99.com*, 2021. [Online]. Available:  
[https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2\\_M4\\_U2\\_T2.pdf](https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2_M4_U2_T2.pdf). [Accessed: 23- Jan- 2021].
- [3] *Webdiis.unizar.es*, 2021. [Online]. Available:  
[http://webdiis.unizar.es/asignaturas/TC/wp/wp-content/uploads/2011/09/Intro\\_Flex\\_Bison.pdf](http://webdiis.unizar.es/asignaturas/TC/wp/wp-content/uploads/2011/09/Intro_Flex_Bison.pdf). [Accessed: 23- Jan- 2021].