

# Funcionamiento

## Análisis y compilación

Dado el archivo de entrada `practica_flex.l` con el contenido descrito anteriormente, se procede a analizar y obtener el código de salida en C en el archivo `lex.yy.c` con Flex . Posteriormente se compila dicho archivo con GCC 7.5 y se obtiene el ejecutable definido como `scanlex`. (Fig.4).

```
ivan@ivan-OptiPlex-790: ~/Escom/compiladores/flex 1
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$ flex -o source/lex.yy.c source/practica_flex.l
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$ ls source/*.yy.c
source/lex.yy.c ← fuente C
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$ gcc source/lex.yy.c -o scanlex -lfl
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$ ls
scanlex source
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$
↑ fuente Flex
↑
ejecutable
```

Fig.4. Análisis y compilación del analizador léxico.

## Pruebas

Dado un conjunto de cadenas, se ingresan al analizador léxico mediante la entrada estándar de la terminal. El siguiente conjunto de figuras muestra el resultado entregado por el programa para cada cadena (primera línea en cada figura).

|  |  |
|--|--|
| 123var1<br>[entero positivo:123]<br>[variable:var1]<br>[salto de línea]                    | Observamos que la cadena de entrada no cumple con las condiciones para ser un variable, es por eso que el analizador empata el la primera subcadena <b>123</b> como un número entero positivo y lo segunda subcadena <b>var1</b> como un variable.   |
| -5,+5<br>[entero negativo:-5]<br>[,]<br>[entero positivo:+5]<br>[salto de línea]           | La primera subcadena <b>-5</b> empata en la definición de entero negativo, la coma "," se separa y se identifica la subcadena <b>+5</b> como un entero positivo.   |
| -5.5,+5.5<br>[decimal negativo:-5.5]<br>[,]<br>[decimal positivo:+5.5]<br>[salto de línea] | De forma similar al caso anterior se empata la subcadena <b>+5.5</b> como un número decimal positivo, la coma se separa y la subcadena <b>-5.5</b> se empata con la definición de un decimal negativo.   |
| +5.5^-5<br>[potencia: +5.5^-5]<br>[salto de línea]   | En este caso, dada la definición de la expresión regular correspondiente a la potencia, la primera subcadena <b>+5.5</b> se empata con la definición de decimal negativo, el acento circunflejo ^ se asocia a la operación de potencia y la subcadena <b>-5</b> se empata con la definición de entero negativo. Estas tres cadenas en conjunto corresponden al patrón que describe la regla de potencia, por lo cual la cadena en su totalidad es identificada como tal. |

|  |   |
|--|---|
| <pre>Mod(5,3) [operacion: Mod(5,3)] [salto de línea]</pre> | <p>La cadena corresponde a la definición de la operación módulo, esto ya que los paréntesis no forman parte de la expresión regular para las variables. La coma no es separada como en el los casos anteriores de los números, dado que está compone a la expresión regular de módulo que a su vez conforma a la expresión regular de la operación.</p> |
| <pre>5+5+5+5 [operacion: 5+5+5+5] [salto de línea]</pre>   | <p>La cadena de entrada es un conjunto de números enteros separados por el signo +, esta empata completamente con la expresión regular de la operación.</p>   |

```
ivan@ivan-OptiPlex-790: ~/Escom/compiladores/flex
ivan@ivan-OptiPlex-790:~/Escom/compiladores/flex$ ./scanlex
Compiladores - Practica 2
Escáner léxico con Flex
-----
123var1
[entero positivo:123]
[variable:var1]
[salto de línea]

-5,+5
[entero negativo:-5]
[,]
[entero positivo:+5]
[salto de línea]

-5.5,+5.5
[decimal negativo:-5.5]
[,]
[decimal positivo:+5.5]
[salto de línea]

+5.5^-5
[potencia: +5.5^-5]
[salto de línea]

Mod(5,3)
[operacion: Mod(5,3)]
[salto de línea]

5+5+5+5
[operacion: 5+5+5+5]
[salto de línea]
```

*Fig.5. Ejecución completa del programa.*

## Conclusiones

La disponibilidad de herramientas como Flex para la generación de analizadores léxicos supone una gran ventaja en el proceso de comprensión de la fase correspondiente; pues si bien, la base teórica se adquiere desde cursos anteriores, la aplicación de dicho conocimiento enfocado en la materia enriquece de manera sustancial el entendimiento del trabajo del compilador, entonces, la posibilidad de centrar la atención en la construcción de las expresiones regulares que definen al lenguaje y no en la construcción del analizador en sí, es la que propicia el aprovechamiento al que se refiere.