Compiladores

Carlos Eduardo Atencio Torres

Semestre 2020-II

Práctica 2 – FLEX

Objetivo:

Utilizar la herramienta FLEX para construir un analizador léxico

Herramientas:

FLEX: Es una herramienta que nos permite generar analizadores léxicos. Tiene su propia sintaxis y pide como requisito tener un conocimiento básico sobre manipulación de expresiones regulares.

Para instalar en linux:

sudo apt-get install flex

Para instalar en windows:

http://gnuwin32.sourceforge.net/packages/flex.htm

1. Recordando:

- Las instrucciones en flex deben estar en un archivo con extensión **.l** (punto ele).
- Compilar el archivo de instrucciones utilizando el comando flex:

  flex instrucciones.l
- Si todo está bien, se generará un archivo **lex.yy.c**
- Luego hay que generar el ejecutable utilizando C:

  gcc lex.yy.c -L/lib -lfl
- El comando anterior deberá generar el ejecutable **a.out** o **a.exe** (windows) y podemos ejecutarlo con algún archivo de prueba:

  ./a.out < prueba.txt

Consideren el siguiente texto para el punto 2, 3 y 4:

Primer numero es 123

```
El segundo numero es 456.789
Y puedo tener ambos tipos: 999 y 999.999
```

Considerar el siguiente texto para el punto 5:

```
1
armando
2
belinda
3
carlos
4
dora
5
esteban
```

2. Definiendo macros:

En flex es posible definir macros colocándolos en la primera parte del archivo de instrucciones. Por ejemplo, vamos a definir progresivamente un real a partir de un entero y a partir de expresiones regulares así:

```
digito [0-9]
entero {digito}+
real {entero}\.{entero}

%%
{entero} { printf("TOKEN ENTERO"); }
{real} { printf("TOKEN REAL"); }
```

Al realizar pruebas con este ejemplo notamos que podemos identificar los tokens enteros y números reales.

3. ECHO;

Este es un comando especial que nos permite manipular, hasta cierto punto, el token que hemos podido identificar. Por ejemplo:

```
digito [0-9]
entero {digito}+
real {entero}\.{entero}

%%
{entero} { printf("TOKEN: ["); ECHO; printf("] ENTERO") ; }
{real} { printf("TOKEN: ["); ECHO; printf("] REAL); }
```

4. yytext

Este es un comodín en flex y es el responsable de capturar el contenido reconocido en las reglas.

Por ejemplo:

```
digito [0-9]
entero {digito}+
real {entero}\.{entero}

%%
{entero} { printf("TOKEN: [%s] ENTERO", yytext); }
{real} { printf("TOKEN: [%s] ENTERO", yytext); }
```

Obtendremos el mismo resultado que el ejemplo anterior usando ECHO;

5. Uso de C

Podemos adicionar código en C en la tercera parte del archivo de configuración de flex. Por ejemplo:

```
      int contador = 0;

%%

[0-9]  { contador++; }

%%

int main() {
  yylex();
  printf("Se conto: %d\n numeros", contador);
}
```

Tener cuidado con el primer espacio en blanco ya que está colocado de forma intencional.

Otro ejemplo más avanzado:

```
digito [0-9]
entero {digito}+
        void imprime(char *sToken) {
                int iToken = atoi(sToken);
                if( iToken < 5 )
                        printf("[%d es menor que 5]", iToken);
                else
                        printf("[%d es mayor o igual que 5]", iToken);
        }

%%
{entero}  { imprime(yytext); }
%%
```

Ejercicios

Transformar una conversación de whatsapp a un formato de tablas utilizando latex.

Ejemplo de conversación:

```
[5:57 p. m., 9/9/2020] Carlos EAT: Feliz cumpleaños Edward!
[6:55 p. m., 9/9/2020] Edward H: Hola Carlos, muchas gracias =D
[6:57 p. m., 9/9/2020] Carlos EAT: Espero que la pases bien
[10:02 p. m., 9/9/2020] Edward H: Si mi esposa hizo una torta de chocolate, saludos
```

Ejemplo de salida en latex (para la primera conversación)

```
\begin{tabular}{|c|p{3cm}|p{8cm}|p{2cm}|} \hline
 \textsc{Carlos EAT} &
 1 &
 Feliz cumpleaños! &
 \begin{tabular}{c}
    \footnotesize{9/9/2020} \\
    \Large{\textbf{5:57}} \\
    pm
 \end{tabular} \\
\hline
\end{tabular}
```

Ejemplo de salida para la conversación:

1	Carlos EAT	Feliz cumpleaños!	9/9/2020 **5:57** pm
2	Edwar H	Hola Carlos, muchas gracias =D	9/9/2020 **6:55** pm
3	Carlos EAT	Espero que la pases bien	9/9/2020 **6:57** pm
4	Edwar H	Si mi esposa hizo una torta de chocolate, saludos	9/9/2020 **6:57** pm