

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

Proyecto 1

Un generador de Scanners

José Ceciliano Granados
2016087245

Audra Rodríguez Mora
2015101893

David Valverde Zuñiga
200922986

Instituto Tecnológico de Costa Rica
Compiladores e Intérpretes
I Semestre 2019

Introducción

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

Introducción

Flex es una herramienta de análisis lexico desarrollada para la generación de Scanners, programas que reconocen patrones léxicos en el texto.

Su nombre significa "fast lexical analyzer generator" y se encarga de leer las entradas recibidas para generar la descripción de un scanner en forma de pares de expresiones regulares y código C, llamadas "reglas".

Scanning

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

Scanning

Mediante proceso de Scanning se identifican los diferentes lexemas de un lenguaje. Flex genera como salida un archivo C que define una rutina en específico, que junto con una biblioteca genera un ejecutable con la capacidad de analizar su entrada para la aparición de expresiones regulares. Cada vez que encuentra una, ejecuta el código C correspondiente.

Todo esto se puede traducir a que Flex es capaz de crear un "Deterministic Finite Automaton" (DFA) que se utilizará para adquirir los diferentes lexemas que se pretenden escánear.

Histograma I

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

Histograma

A continuación se presenta un histograma que muestra la cantidad de cada tipo de *token* encontrado en el código fuente:

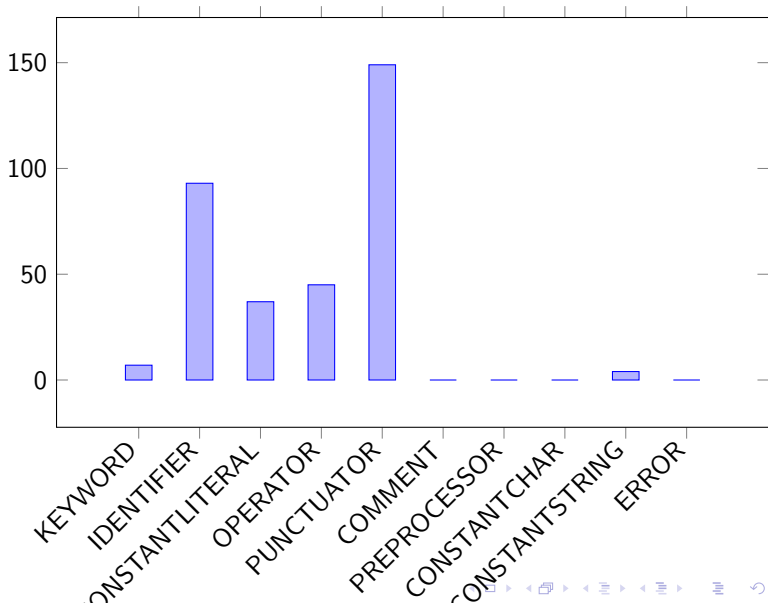
Histograma

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986



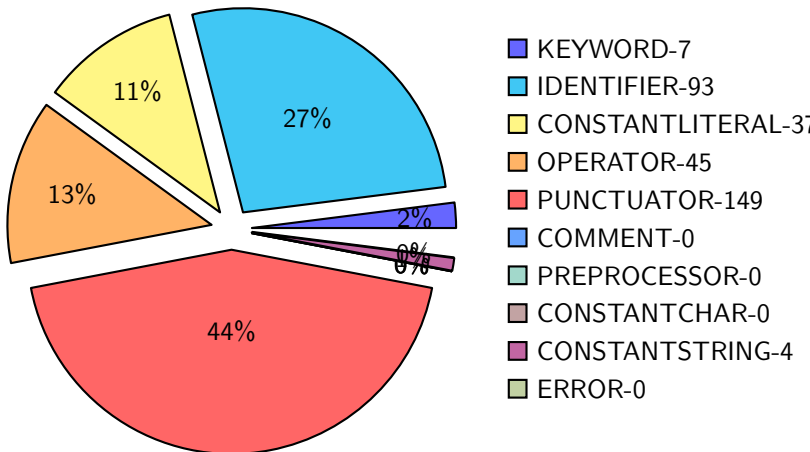
Histograma

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986



Histograma I

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

```
[1mTSource.in:3:1: [31;1mError de sintaxis: [0mcarácter  
IDENTIFIER inesperado main() [32;1m[0m
```

```
[1mTSource.in:15:30: [31;1mlexical error: [0m"' ' is not a valid token  
j=write(1,(",1))j=write(1,(""); [32;1m[0m
```

```
[1mTSource.in:17:13: [31;1mlexical error: [0m"' ' is not a valid token  
j=write(1,(",1) [32;1m[0m
```

Compilacion terminada con 3 errores.

Analisis Léxico I

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

Codigo fuente

A continuación se presenta el código fuente con colores demostrando la división de *tokens*.

Resultado de sintaxis I

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

```
int
M[5000]={2},*u=M,N[5000],R=22,a[4],l[]={0,-1,39-1,-1},m[] =
*d=N,c,e,f,g,i,j,k,s;
main(){
for(M[i=39*R-1]=24;f|d>=b;){c=M[g=i];i=e;for(s=f=0;
s<4;s++)if((k=m[s]+g)>=0&&k<39*R&&l[s]!=k%39&&(!M[k]||!j&&
]=s;
if(f){f=M[e=m[s=a[rand()/(1+2147483647/f)]]]+g];
j=j<f?f:j;f+=c&-16*!j;M[g]=
c|1<<s;
M[*d++=e]=f|1<<(s+2)%4;
}else e=d>b++?b[-1]:e;
}
j=write(1,(""));
,1)for(s=39;--s;
j=write(1,(",1))j=write(1,(""));1)for(;j=write(1,(" "),,1) e--;j
```

Resaltado de sintaxis II

Proyecto 1

José Ceciliano
Granados
2016087245

Audra Rodríguez
Mora
2015101893

David Valverde
Zuñiga
200922986

```
write(1, (" ,1))}
```