

Instituto Tecnológico de Costa Rica

Proyecto 1 - Analizador Léxico

Compiladores e Intérpretes

Luis Rojas - Ariel Montero - Bryan Jiménez
IS 2017

April 4, 2017

Explicación General

Datos Generales

Scanning

Flex

Programa Fuente

Diagramas

Histograma

Diagrama Pastel

Definiciones

Token: Familia de lexemas, es una categoría donde se clasifican los lexemas.

Lexema: Ejemplo particular de la variable.

Hilera: Cadenas de texto generalmente marcadas entre comillas.

Constante: Valor que no cambia, ya está definido.

Identificadores: Conjunto de caracteres alfanuméricos que sirven para identificar las entidades del programa

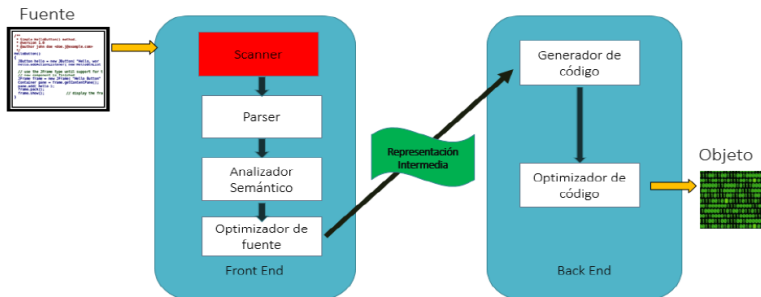
Definiciones

Palabras reservadas: Palabras que son usadas por el lenguaje y no pueden modificarse ni su función.

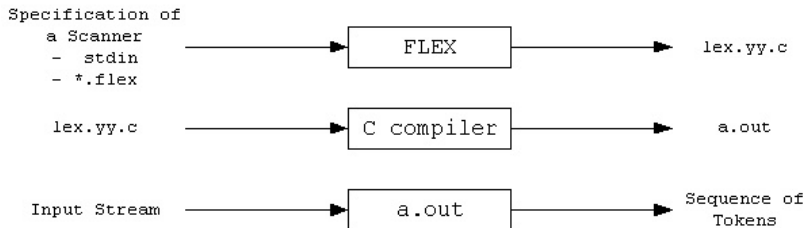
Símbolos especiales: Símbolos con significado especial en el lenguaje por lo que no se pueden usar para otro propósito.

Operadores: Símbolos utilizados para realizar operaciones lógicas y matemáticas.

Scanner



Un analizador léxico o scanner es la primera fase de un compilador y consistente en un programa que recibe como entrada el código fuente de otro programa (secuencia de caracteres) y produce una salida compuesta de tokens (componentes léxicos) o símbolos. Estos tokens sirven para una posterior etapa del proceso de traducción, siendo la entrada para el analizador sintáctico o parser. La especificación de un lenguaje de programación suele incluir un conjunto de reglas que definen el léxico. Estas reglas consisten comúnmente en expresiones regulares que indican el conjunto de posibles secuencias de caracteres que definen un token o leóxima.



Flex es una herramienta para generar scanners, flex lee las entradas para tener una descripción del scanner que debe generar. La descripción se realiza en pares de expresiones regulares y código en C, que se llaman reglas. Genera una salida en un archivo en C 'lex.yy.c' que define a una rutina 'yylex()'. Este archivo es compilado y unido con la librería '-lfl' para generar el ejecutable. Cuando se llama al ejecutable este analiza la entrada por ocurrencias de las expresiones regulares, cuando encuentra una ejecuta el código correspondiente en C.


```
int main ( )
{
    int cant2 = 0 ;
    if ( cant2 == 2 ) { }
    printf ( " Hola mundo " ) ;
    x = 2 ;
    return 0 ;
}

int gcd ( )
{
    int n1 , n2 , i , gcd ;

    printf ( " Enter two integers : " ) ;
    scanf ( " %d %d " , &n1 , &n2 ) ;
```

```
for ( i = 1 ; i <= n1 && i <= n2 ; ++ i )  
{  
if ( n1 % i == 0 && n2 % i == 0 )  
gcd = i ;  
}
```

```
printf ( " G . C . D of %d and %d is %d " , n1 , n2  
, gcd ) ;
```

```
return 0 ;  
}
```

```
int palindrome ( )  
{
```

```
int n , reversedInteger = 0 , remainder , originalInteger ;
```

```
printf ( " Enter an integer : " ) ;  
scanf ( " %d " , &n ) ;
```

```
originalInteger = n ;
```

```
while ( n != 0 )  
{  
    remainder = n % 10 ;  
    reversedInteger = reversedInteger * 10 + remainder ;  
    n / 10 ;  
}
```

```
if ( originalInteger == reversedInteger )
```

```
printf ( " %d is a palindrome . " , originalInteger ) ;  
else printf ( " %d is not a palindrome . " , originalInteger  
) ;
```

```
return 0 ;  
}
```

```
int primo ( int n ) {  
int i , m ;  
for ( i = 2 ; i < n ; i ++ ) {  
m = n % i ;  
if ( m == 0 ) {  
return 1 ;  
} else {
```

```
return 0 ;  
}  
}  
}
```

```
int armstrong ( )  
{  
int number , originalNumber , remainder , result = 0 , n =  
0 ;
```

```
printf ( " Enter an integer : " ) ;  
scanf ( " %d " , &number ) ;
```

```
originalNumber = number ;
```

```
while ( originalNumber != 0 )  
{  
originalNumber / 10 ;  
++ n ;  
}
```

```
originalNumber = number ;
```

```
while ( originalNumber != 0 )  
{  
remainder = originalNumber % 10 ;  
result + = pow ( remainder , n ) ;  
originalNumber / 10 ;  
}
```

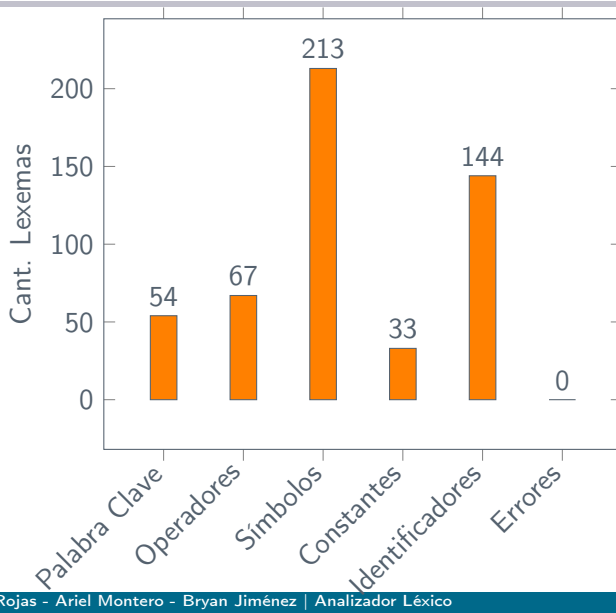
```
if ( result == number )  
printf ( " %d is an Armstrong number . " , number ) ;  
else printf ( " %d is not an Armstrong number . " ,  
number ) ;
```

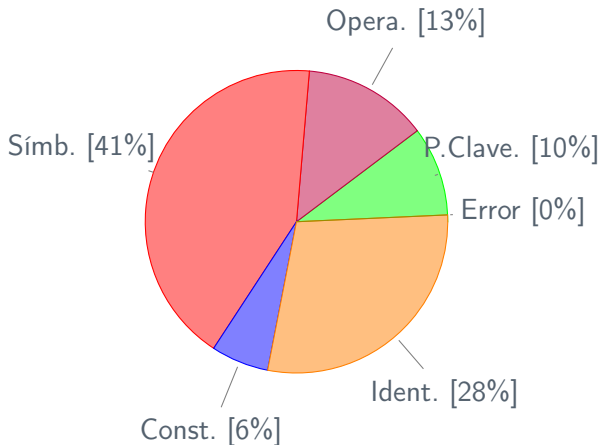
```
return 0 ;  
}
```

```
int main ( ) {  
int factorial , result = 1 ;
```

```
x = 2 - 3 ;  
hola = 23 ;  
if ( x == 2 ) {  
printf ( " Yo soy Luis " ) ;  
} else {  
printf ( " No soy Luis " ) ;  
}  
int ariel = 3 ;  
int bryan = 7 ;  
while ( x != bryan ) {  
printf ( " NO SOY BRYAN " )  
}  
}
```


Histograma







MUCHAS GRACIAS