



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Practica 1

Lenguajes Formales

Alumno/s: Dickinson Bedoya Perez
Anna Gracia Colmenarejo
Mariona Valero Canadell
Enseñamiento: Ingeniería Informática
Profesor/es: David Riaño
Sergi Gómez
Fecha: 04/2021

Índice

Índice	1
Sección de definiciones	2
Reglas léxicas	3
Código usuario	3
Juego de pruebas	3

Sección de definiciones

En la sección de definiciones hemos declarado las variables que hemos utilizado en el programa para contabilizar los tags de apertura, los tags de cierre, los atributos, los tags con atributos, los elementos simples, los elementos vacíos y el número de caracteres.

También hemos creado dos funciones. La primera es **contar_caracteres()** que se ocupa de mirar cuando acaba el tag de apertura y contar los caracteres que hay antes de llegar al tag de cierre. Para esta función se usa una variable que proporciona flex, *yytext*. Esta variable contiene la última coincidencia con una Regex que se ha producido y la guarda como una lista de caracteres.

Y la segunda **tagsCoinciden()** que comprueba que el tag de apertura y el tag de cierre tienen el mismo nombre, para eso hemos guardado el nombre del tag de apertura en una lista de caracteres que hemos ido comparando con el nombre del tag de cierre. Para poder coger como string todo el nombre y así mirar cada caracter del string hemos utilizado la función *yytext*.

Y finalmente hemos creado las expresiones regulares que corresponden a cada uno de los elementos principales.

```
texto [^<>&'"]+
nombreTag [a-zA-Z0-9_]*
atributo [ ]+{nombreTag}[ ]*=[ ]*['"]{texto}['"] [ ]*
tagApertura [<]{nombreTag}{atributo}* [>]
tagCierre [<][ / ]{nombreTag} [>]
elemSimple {tagApertura}{texto}{tagCierre}
elemVacio {tagApertura}{tagCierre}
```

Reglas léxicas

En cuanto al uso de las expresiones regulares usamos básicamente dos que son las que comprueban todo el lenguaje introducido.

Para ello hacemos uso de **elemSimple** que comprueba si un es elemento simple con o sin atributo. Si resulta que es un elemento simple incrementa el número de elementos simples, el número de aperturas (con o sin atributos y el número de atributos que hay) y el número de cierres además de comprobar que estos dos coinciden.

En el caso de **elemVacío** sólo se incrementa el número de elementos vacíos y el número de atributos que tiene (si es que hay). También se comprueba que el tag de apertura y el de cierre coincida.

Código usuario

En la parte de código de usuario abrimos el fichero que se pasa por parámetros con la variable `flex_yyin`, si eso se hace correctamente se ejecuta el código escrito anteriormente, que se compila usando la función `yylex()`. Seguidamente se muestra por pantalla el resultado de los contadores de tags y atributos que se pedían en el enunciado.

Juego de pruebas

Para probar el analizador léxico hemos un script que ejecuta el flex y compila el archivo `.c` que se genera, además de ejecutar el archivo ejecutable con un archivo que se le pasa por parámetro.

Para probar que el funcionamiento de las expresiones regulares hemos hecho varios archivos que contienen el lenguaje pedido con errores para ver si estas hacen bien su funcionamiento.

Todos los resultados son correctos.