Universidad de Santiago de Chile
Facultad de Ciencias
Departamento de Matemáticas y Ciencia de la Computación

UNIVERSIDAD DE SANTIAGO DE CHILE

Laboratorio n°2

Expresiones Regulares y Lex

Integrante: Rodrigo Kobayashi Araya
Profesor: Rodrigo Abarzúa
Fecha de entrega: 12 de agosto de 2020
Santiago, Chile

Índice de Contenidos

Índice de Tablas

1. Problema

1.1. Desarrollar un programa en Lex tal que:

1. Analice un texto de entrada y elimine espacios y tabulaciones redundantes.

2. Al recibir un fichero de texto y una palabra, cuente las apariciones del string en dicho fichero.

3. Borre los comentarios que aparezcan en el fichero de texto, Se suponen de comentarios de una sola línea que empiezan con el símbolo #.

4. Imprima un texto tal y como está en el archivo de entrada, pero cada vez que aparezca un ";", reemplazar por un salto de línea.

5. Indique cuántas veces ha detectado un número entero en un fichero de texto.

6. Finalmente, que imprima el texto de salida con los cambios de los puntos anteriores, cifrar el texto utilizando la siguiente tabla de sustitución:

Tabla 1: Codificación.

a	**C**
b	**I**
c	**S**
d	**Q**
e	**V**
f	**N**
g	**F**
h	**O**
i	**W**
j	**A**
k	**X**
l	**M**
m	**T**
n	**G**
o	**U**
p	**H**
q	**P**
r	**B**
s	**K**
t	**L**
u	**R**
v	**E**
w	**Y**
x	**D**
y	**Z**
z	**J**
0	**5**
1	**4**
2	**7**
3	**2**
4	**9**
5	**1**
6	**3**
7	**0**
8	**6**
9	**8**

2. Resolución

2.1. Lex y expresiones regulares

Lex y Flex son programas que permiten crear analizadores léxicos, es decir, son programas que permiten crear "scanners"de strings, se pueden implementar de manera que lean un archivo y ejecuten algún tipo de código, escrito en C, al encontrar un stiring definido anteriormente por una expresión regular.

En el laboratorio se crea un analizador léxico que detecta ciertos strings claves para satisfacer las indicaciones entregadas anteriormente. Para llevar a cabo esto se tiene que encontrar la expresión regular para cada punto y luego, describir la salida del programa.

La Entrada y Salida escogidas para el programa son dos archivos, El programa pide el nombre del archivo de entrada y crea uno de salida (llamado "salida").

Para la compilación basta con acceder a la carpeta donde se encuentra el programa por consola y luego ejecutarlo:

```
lex parser.lex
gcc lex.yy.c
./a.out
```

luego de esto se pedirá el nombre del archivo de prueba, luego de ingresar el nombre del archivo, el programa analizará el fichero de texto según las reglas especificadas anteriormente.

2.2. Expresiones regulares y Salida

1. Analice un texto de entrada y elimine espacios y tabulaciones redundantes:
Las expresiones regulares necesarias para detectar redundancias de espacios y tabulaciones son ()+ y ()+ , al detectar estos strings son reemplazados por un solo espacio y una sola tabulación respectivamente.
2. Al recibir un fichero de texto y una palabra, cuente las apariciones del string en dicho fichero:
No se pudo llevar a cabo este punto
3. Borre los comentarios que aparezcan en el fichero de texto, Se suponen de comentarios de una sola línea que empiezan con el símbolo #.
La expresión regular para los comentarios es #(.)*, al detectar este string, el prorama no escribe nada en el archivo de salida, eliminando el comentario.
4. Imprima un texto tal y como está en el archivo de entrada, pero cada vez que aparezca un ";", reemplazar por un salto de línea.
La expresión regular para lo descrito es simplemente ";.ªl encontrar esto, el programa escribe un salto de linea en el archivo de salida.
5. Indique cuántas veces ha detectado un número entero en un fichero de texto.
La expresión regular para detectar números es [0-9]+ , al detectar el string, el programa suma uno a un contador global descrito como .ºcurrenciaInt".
6. Finalmente, que imprima el texto de salida con los cambios de los puntos anteriores, cifrar el texto utilizando la siguiente tabla de sustitución:
Para este punto, se hace una expresión regular para cada caracter, indicando al programa que debe escribir su forma codificada en el archivo de salida.

3. Conclusiones

Luego de hacer la investigación pertinente sobre las expresiones regulares y el uso de ellas en parsing o en programas de uso diario como el buscador de palabras de los exploradores, se puede concluir que estas son un pilar fundamental para la computación, desde sus inicios ha sido necesario generar una conexión entre el lenguaje y la máquina. La base matemática enseñada en cátedra y su aplicación en este laboratorio demuestran la importancia de una herramienta tan simple como efectiva. Las expresiones regulares.

Con respecto a lo que se puede mejorar del laboratorio, un gran error por parte del estudiante es no poder manejar bien los tiempos de desarrollo del programa, debido a esto no se pudo completar a cabalidad.