

Alexandro Francisco Marcelo González A01021383

Profesor: Dr. Víctor Manuel de la Cueva Hernández

Matemáticas Computacionales, grupo 1

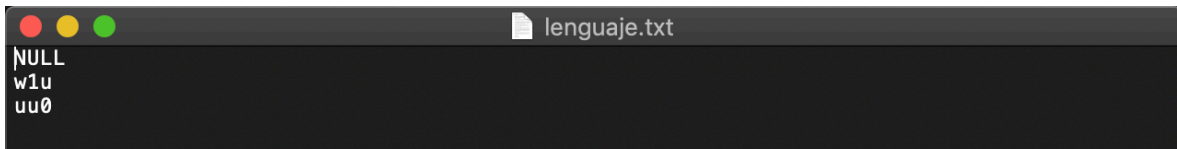
22/03/2019

Proyecto 1. Definiciones Recursivas

Bienvenido al manual de usuario, el cual te ayudará a aprender cómo usar el programa para obtener los substrings de un lenguaje mediante definiciones recursivas, dado un archivo de texto que contenga el lenguaje.

Para usar el programa de manera adecuada, primero se debe descargar los archivos cuyos nombres son: **“definiciones_recursivas.h”** y **“main.cpp”**, los anteriores deberán ser guardados dentro de una carpeta creada con el nombre que quiera, para una mejor gestión. Para este caso, se deberá tener también un archivo de texto el cual deberá ser guardado dentro de la misma carpeta donde guardaste los anteriores, para este ejemplo, se creará un archivo de texto con terminación **“.txt”**, cuyo nombre será **“lenguaje.txt”**.

Una vez creado un archivo **“lenguaje.txt”** dentro de la carpeta mencionada, se deberá editar el texto del archivo dicho haciendo doble click sobre él para abrirlo, para el caso se deberá escribir lo siguiente dentro de él:



Respetando los saltos de línea, tal cual se muestra en la imagen. Una vez escrito lo anterior, guardaremos el archivo, dando click en **“Archivo”** (parte superior izquierda), después en la opción **“Guardar”**.

El formato puesto dentro del archivo **“lenguaje.txt”** significa que en el primer renglón especifica el caso base del lenguaje, en los siguientes renglones se especifican los pasos recursivos del lenguaje, donde las variables que representen strings que pertenecen al lenguaje sólo pueden ser {u,v,w,x,y,z}:

caso_base

paso recursivo 1

paso recursivo 2

paso recursivo 3

...

paso recursivo n

*Nota: si se requiere que el caso base sea nulo (∅) se deberá escribir: **NULL** (respetando mayúsculas)

Una vez teniendo los archivos “**definiciones_recursivas.h**”, “**main.cpp**” y “**lenguaje.txt**” dentro de la misma carpeta, se deberá utilizar un compilador de preferencia, para el caso, se utilizará la terminal de linux, se puede utilizar DevC++ o cualquier otro compilador de C++.

Por lo que abriremos una terminal que se encuentre en el “path” de la carpeta donde se encuentran nuestros archivos. Por lo que procederemos a escribir lo siguiente en la terminal para poder compilarlo: **<g++ main.cpp -o run.out>**, en la imagen siguiente se aprecia mejor.

```
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$ ls
definiciones_recursivas.h      lenguaje.txt      main.cpp
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$ g++ main.cpp -o run.out
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$ ls
definiciones_recursivas.h      lenguaje.txt      main.cpp      run.out
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$
```

Una vez compilado el programa nos generará “**run.out**” el cual es nuestro ejecutables del programa, para lo que procederemos a correrlo en la terminal con el siguiente comando: **<./run.out>**, como lo muestro en la siguiente imagen.

Escrito lo anterior nos pedirá el nombre del archivo donde se encuentra nuestro lenguaje, en este caso es “**lenguaje.txt**” y no pedirá el numero de pasos recursivos que deseemos hacer, lo introduciremos, en este caso **3**, dandonos como resultado los substrings generados mediante pasos recursivos dado un lenguaje. En la siguiente imagen se muestra de mejor manera lo anterior:

```
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$ ./run.out
Dame el nombre del archivo
lenguaje.txt
Cuántas veces quieres hacer el paso recursivo
3
Substrings generados en el paso recursivo 0: NULL,
Substrings generados en el paso recursivo 1: 1, 0,
Substrings generados en el paso recursivo 2: 11, 01, 111, 011, 10, 110, 010, 000,
Substrings generados en el paso recursivo 3: 1111, 0111, 101, 1101, 0101, 0001, 11111, 01111, 1011, 11011, 01011, 00011, 1110, 0110, 11110, 01110, 1010, 11010,
01010, 00010, 111111, 011111, 10111, 110111, 010111, 000111, 11101, 01101, 111101, 011101, 10101, 110101, 010101, 000101, 1111111, 0111111, 101111, 1101111, 0
101111, 0001111, 111011, 011011, 111011, 011011, 101011, 1101011, 0101011, 0001011, 111110, 011110, 10110, 110110, 010110, 000110, 1111110, 0111110, 101110, 1
101110, 0101110, 0001110, 111010, 011010, 1111010, 0111010, 101010, 1101010, 0101010, 0001010, 1000, 11000, 01000, 111000, 011000, 1111000, 0111000, 101000, 1
101000, 0101000, 0001000, 0110110, 10100, 1101100, 0100100, 0000000,
MacBook-Pro-de-Alexandro:proyecto1 alexmarcelo$
```

Gracias por usar el programa para poder generar todos los substrings de un lenguaje mediante pasos recursivos.