

Compiladores – CCOMP8-1

Rodrigo Andre Cayro Cuadros

Analizador léxico propuesta “TRISTAGE”:

Tema:

Como parte del trabajo del curso de compiladores la propuesta es “tristage” un lenguaje orientado a la manipulación de imágenes, es decir que podamos modificar ciertas características de una o de varias imágenes seleccionadas previamente por usuario, facilitando las operaciones entre ellas, como por ejemplo unir varias imágenes en una o aplicarles ciertos efectos especiales, ejemplo: ESCALA de la imagen, ROTACIÓN de la imagen, RUIDO, entre otros. Para poder realizar estas funciones haremos uso de magick++.

Los tipos de datos permitidos para este lenguaje serán string que representan los nombres de los archivos, flotantes y enteros para hacer las operaciones de las imágenes. Dentro de sus operadores nos permitirán combinar imágenes y también aplicarles distintas funciones anidadas como el uso del operador “++”, también podremos comparar ambas imágenes usando el operador “==”, al igual que los objetos se asignan a las variables para así manipular las imágenes.

Funciones a implementar:

- Lectura de caracteres.
 - Getchar() devuelve el siguiente carácter de la entrada y mueve el puntero del carácter.
 - Peekchar() devuelve el siguiente carácter sin mover el puntero.
- Generación de tokens:
 - antokens(string, vector<string>&) => <TIPO_DE_TOKEN, LEXEMA>
- Funciones especiales:
 - addNoise - Añade una cantidad de ruido especificado por el usuario a la imagen:



- reduceNoise – Permite reducir el ruido de una imagen determinada, es el proceso inverso a lo que realiza la función anterior.
- colorRi – Coloriza una imagen usando una cantidad o valor de opacidad determinada dentro de los colores verdes, rojos y azules.



- compare – Compara dos imágenes proporcionadas por el usuario he indica si son iguales, la comparación se hace pixel por pixel.



- read - lee una imagen externa para utilizarla como objeto.



- save – Guardamos una imagen con el nombre específico tras ya haber trabajado con ella.



- rotate - Rota la imagen específica, Scale() escala una imagen, ambas funciones con un valor determinado.



- combine – Con esta función podremos combinar dos imágenes específicas, de esta manera se sobrepondrán entre ellas.



- solar – Aplica el efecto de solarize a la imagen seleccionado por el usuario.



- Tipos de tokens:

- OP: ++, -, *, +, <-, ==, =
- DELIM: ENDFUNCTION - ENDIF – ENDEIF – ENDEL
- RESERVED: FUNCTION , IF, ELSEIF, ELSE, RETURN, PRINT
- ID: [a-z, A-Z] [a-z, A-Z, 0-9]*
- STR: “PALABRAS”, en el caso de tener que poner las comillas como parte de la palabra se hará uso de ´” de esa manera: “Hola ¡’mundo¡” “ == Hola ´mundo´

- NUM: (1-9)(0-9)*.(0-9)*
- Los espacios en blanco, tabulaciones, finales de línea y comentarios se consumen, pero no se devuelven nada.
 - Comentarios: % TU COMENTARIO %

Ejemplo 1:

```
11f1.2
FUNCTION funcion1
    variable = read "name.png"
    variable2 = "name.png"
    variable3 = variable + variable2
    noise = 14.15
    ang = 30
    rotate variable3 30
    addNoise variable3 30
    return variable3
ENDFUNCTION

main:
    save funcion1
    imag = read "path.png"
    imag2 = read "path2.png"

    IF imag == imag2
        PRINT "SON IGUALES"
        imag3 = imag + imag2
        imag3 = rotate imag3 30 ++ scale imag3 2
        save imag3
    ENDIF
;
```

Ejemplo 2:

```
main:
    imag = read "path.png"
    imag2 = read "path2.png"
    %DOS PRIMERAS IMAGENES A COMBINAR%
    imag3 = read "path3.png"
    imag4 = read "path4.png"
    imag5 = read "path5.png"

    imag6 = imag + imag2
    reduceNoise imag3
    colorRi imag4
    opaque imag5

    save imag6
    save imag3
    save imag4
    save imag5
;
```