

# 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.
- colori – 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.



- 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 – ENDEL
  - RESERVED: FUNCTION , IF, THEN, 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 j" de esa manera: "Hola j'mundoj' " == 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
;

```

Gramática:

Especificación de los Operadores:

- “++” – Este operador sirve tanto para concatenar funciones como para concatenar operadores comparativos. Ej: IF some1 == some2 ++ some2 <- some3, en este caso hace referencia a un “AND”.

- “--” – Este operador sirve para concatenar operadores comparativos. Ej: IF some1 == some2 -- some2 <- some3, en este caso hace referencia a un “OR”.
- “-” – Este operador recorta los bordes de una imagen según y cómo lo indique el usuario, está acompañado de un numero entero que indica el porcentaje a recortar.
- “\*” – Este operador indica que ambas imágenes serán combinadas y se las aplicara una función determinada. Ej: some1 = \* some2 some3 function1, en este caso estamos combinando en some1 las imágenes some2, some3 y con una función aplicada.
- “+” – Este operador permite combinar imágenes varias veces.
- “<-” – Operador comparativo, este indica si ambas imágenes son diferentes, entonces se cumple con la condición.
- “==” – Operador comparativo, este indica si ambas imágenes comparadas con iguales, entonces se cumple con la condición.
- “=” – Este operador permite asignar un valor a una variable.

#### GRAMATICA:

```

S → M
M → F main: B ;
F → Function: string B return id endFunction F
  | ε
B → I
  | R
  | A
  | ε
A → id assign L B
L → R H
  | G
G → id O id D
  | * id id
D → O id
  | ε
O → +
  | -
R → X
X → addnoise id num
  | reducennoise id num
  | print string
  | colorri id num num
  | save id
  | read string
  | rotate id num
  | scale id num

```

```
H → ++ R H
    | .
I → if Q then B Z
Z → endif
    | else B endel
Q → id P
P → assign assign id J
    | <- id J
J → ++ Q
    | -- Q
    | ε
```

LINK DEL REPOSITORIO:

<https://github.com/rodRigocaU/compiladores->