

Práctica de Teoría de autómatas y computación

Entrega 1



Enrique albalate prieto, Lucía de ancos VILLA, PABLO del hoyo abad

24/03/2023

1. Descripción de la práctica.
2. Tabla de elementos léxicos.

|  |  |  |  |
| --- | --- | --- | --- |
| EXPRESIÓN REGULAR | DESCRIPCIÓN | NOMBRE TOKEN | EJEMPLO CADENA |
| arbol | Palabra reservada | ARBOL | árbol |
| node | Palabra reservada | NODE | node |
| shape | Palabra reservada | SHAPE | shape |
| label | Palabra reservada | LABEL | label |
| color | Palabra reservada | COLOR | color |
| fontcolor | Palabra reservada | FONTCOLOR | fontcolor |
| style | Palabra reservada | STYLE | style |
| edge | Palabra reservada | EDGE | edge |
| dir | Palabra reservada | DIR | dir |
| hijos | Palabra reservada | HIJOS | hijos |
| ; | Signo de puntuación | SEMI | ; |
| [ | Signo de puntuación | LSQB | [ |
| ] | Signo de puntuación | RSQB | ] |
| = | Signo de puntuación | EQUAL | = |
| , | Signo de puntuación | COMMA | , |
| “ | Signo de puntuación | QUOTE | “ |
| { | Signo de puntuación | LBRACE | { |
| } | Signo de puntuación | RBRACE | } |
| \/\/.\*(\r\n)? | Cualquier combinación de caracteres que comience con “//” (hasta el final de la línea) | COMMENT\_1 | // comment\_1 |
| #.\*(\r|\n)? | Cualquier combinación de caracteres que comience con “#” (hasta el final de la línea) | COMMENT\_2 | # comment\_2 |
| \/\\*(.\*|\r\n)\*\\*\/ | Cualquier combinación de caracteres que comience con “/\*” y acabe por “\*/” (incluso en diferentes líneas) | COMMENT\_3 | /\* comment\_3 \*/ |
| \"[a-zA-Z0-9\_ ]+\" | Cualquier cadena formada por números o letras (debe tener al menos un caracter) situada en comillas (“”) | STR\_QUOTES | “cadena” |
| [a-zA-Z0-9\_]+ | Cualquier cadena formada por números o letras (debe tener al menos un caracter) | IDENTIFIER | grafo1 |
| . | Resto de casos | - | @ |

1. Manual de usuario.

Para poder realizar la práctica, previamente es necesario haber integrado Jflex en Eclipse, para ello, creamos un workspace y un proyecto para Jflex, además de crear la configuración externa de ejecución.

Una vez preparado el entorno, en la raíz del proyecto implementamos en Jflex un fichero “practica.fle” con cada una de las expresiones regulares del lenguaje propuesto en la práctica. Para probar su funcionamiento, ejecutamos dicho fichero con la herramienta externa Jflex, lo cual generará el fichero “Detectatokens.java” dentro de la carpeta src, a esto le consideramos el Main.

Para ejecutar el Main, pinchamos sobre el propio fichero y en Run – Run configurations añadimos como argumento el fichero con el que vayamos a hacer la prueba, después pulsamos en run y se imprimirán por pantalla las cadenas reconocidas.

1. Ejemplo de fichero de entrada y su salida.

Ejemplo de fichero de entrada:

arbol ejemplo {

node [shape = square, label = "a", color = "green", fontcolor = blue , style = bold]

edge [dir = forward, label = "0", color = "blue", fontcolor = red , style = dashed]

a hijos = {b [label = "6", color = "red"],c [label = "100"]} [label = "A", color = "blue", shape = circle];

b hijos = {d [label = "80"] } [label = "B"];

c hijos = {f [label = "20"], g [label = "22"]} [label = "C"];

d hijos = {e [label = "14"] } [label ="D"];

e [label = "E 14", shape = circle];

f [label = "F 5", shape = circle];

g [label = "G 4", shape = circle];

}

# hello

//HOLA

/\*\*\*\*\*

adios

bye

\*/

Salida del fichero:

ARBOL => arbol

IDENTIFIER => ejemplo

LBRACE => {

NODE => node

LSQB => [

SHAPE => shape

EQUAL => =

IDENTIFIER => square

COMMA => ,

LABEL => label

EQUAL => =

STR\_QUOTES => "a"

COMMA => ,

COLOR => color

EQUAL => =

STR\_QUOTES => "green"

COMMA => ,

FONTCOLOR => fontcolor

EQUAL => =

IDENTIFIER => blue

COMMA => ,

STYLE => style

EQUAL => =

IDENTIFIER => bold

RSQB => ]

EDGE => edge

LSQB => [

DIR => dir

EQUAL => =

IDENTIFIER => forward

COMMA => ,

LABEL => label

EQUAL => =

STR\_QUOTES => "0"

COMMA => ,

COLOR => color

EQUAL => =

STR\_QUOTES => "blue"

COMMA => ,

FONTCOLOR => fontcolor

EQUAL => =

IDENTIFIER => red

COMMA => ,

STYLE => style

EQUAL => =

IDENTIFIER => dashed

RSQB => ]

IDENTIFIER => a

HIJOS => hijos

EQUAL => =

LBRACE => {

IDENTIFIER => b

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "6"

COMMA => ,

COLOR => color

EQUAL => =

STR\_QUOTES => "red"

RSQB => ]

COMMA => ,

IDENTIFIER => c

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "100"

RSQB => ]

RBRACE => }

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "A"

COMMA => ,

COLOR => color

EQUAL => =

STR\_QUOTES => "blue"

COMMA => ,

SHAPE => shape

EQUAL => =

IDENTIFIER => circle

RSQB => ]

SEMI => ;

IDENTIFIER => b

HIJOS => hijos

EQUAL => =

LBRACE => {

IDENTIFIER => d

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "80"

RSQB => ]

RBRACE => }

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "B"

RSQB => ]

SEMI => ;

IDENTIFIER => c

HIJOS => hijos

EQUAL => =

LBRACE => {

IDENTIFIER => f

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "20"

RSQB => ]

COMMA => ,

IDENTIFIER => g

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "22"

RSQB => ]

RBRACE => }

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "C"

RSQB => ]

SEMI => ;

IDENTIFIER => d

HIJOS => hijos

EQUAL => =

LBRACE => {

IDENTIFIER => e

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "14"

RSQB => ]

RBRACE => }

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "D"

RSQB => ]

SEMI => ;

IDENTIFIER => e

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "E 14"

COMMA => ,

SHAPE => shape

EQUAL => =

IDENTIFIER => circle

RSQB => ]

SEMI => ;

IDENTIFIER => f

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "F 5"

COMMA => ,

SHAPE => shape

EQUAL => =

IDENTIFIER => circle

RSQB => ]

SEMI => ;

IDENTIFIER => g

LSQB => [

LABEL => label

EQUAL => =

STR\_QUOTES => "G 4"

COMMA => ,

SHAPE => shape

EQUAL => =

IDENTIFIER => circle

RSQB => ]

SEMI => ;

RBRACE => }

COMMENT\_2 => # hello

COMMENT\_1 => //HOLA

IDENTIFIER => adios

IDENTIFIER => bye