

## TRABAJO PRÁCTICO GRUPAL

### Consigna - Primera parte:

Generar un **analizador léxico o lexer con ANTLR** llamado **LexerAFg4** que permita reconocer los tokens de un archivo de entrada como los presentados en **pruebaAFD.af**, **pruebaAFN.af**, **pruebaAFNepsilon.af**. Para ello deberán definirse una serie de reglas léxicas (expresiones regulares) con el nombre indicado a continuación.

- Reglas para reconocer tokens que son cadenas “fijas” o literales:
  1. **AFD**: representa la cadena “AFD”
  2. **AFN**: representa la cadena “AFN”
  3. **AFN\_EPSILON**: representa la cadena “AFN-*epsilon*”
  4. **SIGMA**: representa la cadena “\Sigma”
  5. **EPSILON**: representa la cadena “*epsilon*”
  6. **IGUAL**: representa la cadena “=”
  7. **PARENTESIS\_ABRE**: representa la cadena “(“
  8. **PARENTESIS\_CIERRA**: representa la cadena “)”
  9. **COMA**: representa la cadena “,”
  10. **FUNCION**: representa la cadena “f” o la cadena “\delta”
  11. **TAG\_AUTOR**: representa la cadena “@autor.”
  12. **TAG\_FECHA**: representa la cadena “@fecha.”
  13. **TAG\_VERSION**: representa la cadena “@version.”
  14. **CONJUNTO\_ABRE**: representa la cadena “{“
  15. **CONJUNTO\_CIERRA**: representa la cadena “}”
  16. **CONJUNTO\_VACIO**: representa la cadena “\emptyset”
  17. **COMENTARIO\_LINEA**: representa la cadena “//” seguida por una *secuencia de caracteres* hasta el salto de línea “\n”.
- Reglas que reconocen tokens que tienen un formato dado y pueden estar conformados por diferentes cadenas o secuencias de símbolos.
  18. **NOMBRE\_MAYUSCULA**: es una palabra de longitud mayor o igual a 1, su primer símbolo es una *letra mayúscula*, luego, es seguida de *letras, dígitos y/o guiones bajos*.
  19. **NOMBRE\_ESTADO**: es una palabra de longitud mayor o igual a 2, su primer símbolo es la letra “q”, luego, es seguida de *letras, dígitos y/o guiones bajos*.
  20. **NOMBRE\_AUTOR**: es una secuencia de *caracteres alfabéticos* entre *comillas simples*, pueden contener *espacios en blanco* para separar los nombres y el apellido del autor.
  21. **FECHA**: tiene el formato *dd/mm/aaaa*, donde *dd*, *mm* y *aaaa* son dígitos entre el 0 y el 9.
  22. **VERSION**: tiene el formato *vMayor.vMenor.revision-opcional*. *vMayor*, *vMenor* y *revision* son números naturales, *opcional* es opcional y si es incorporado es la palabra “alpha”, “beta”, o “rc”.
  23. **SIMBOLO**: símbolo imprimible ASCII (desde el 33 “!” al 38 “&”, desde el 48 “0” al 57 “9”, el 64 “@”, y desde el 97 “a” al 122 “z”).

Se debe tener en cuenta que:

- Se pueden ignorar los saltos de línea y las tabulaciones. Sólo se incluyen en el archivo de entrada por cuestiones de legibilidad.

El analizador léxico deberá imprimir un mensaje por cada token reconocido y mostrarlo siguiendo las siguientes reglas:

- Para los tokens enumerados del 1 al 17, se deberá imprimir el nombre del token identificado.
- Para los tokens enumerados del 18 al 23, se deberá imprimir el nombre del token identificado seguido por el valor del token (separado por dos puntos).

Por ejemplo, para el siguiente archivo de entrada

```
AFD Acepta0Pares = (Q, \Sigma, f, q_par0, F)
@autor: 'Silvio'
@fecha: 11/04/2023
@version: 1.0.0-beta
// Definición del conjunto de estados Q
Q= {q_par0, q_impar0}
\Sigma= {0, 1} // definición del alfabeto
F= {q_par0} // definición del conjunto de estados de aceptación
// Definición de la función de transición f
f(q_par0,0)= q_impar0
f(q_par0,1)= q_par0
f(q_impar0, 0)= q_par0
f(q_impar0, 1)= q_impar0
```

se imprimirán los mensajes que se listan a continuación:

```
AFD NOMBRE_MAYUSCULA: Acepta0Pares IGUAL PARENTESIS_ABRE NOMBRE_MAYUSCULA: Q COMA SIGMA COMA
FUNCION COMA NOMBRE_ESTADO: q_par0 COMA NOMBRE_MAYUSCULA: F PARENTESIS_CIERRA
TAG AUTOR NOMBRE AUTOR: 'Silvio'
TAG_FECHA FECHA: 11/04/2023
TAG_VERSION VERSION: 1.0.0-beta
COMENTARIO_LINEA
NOMBRE_MAYUSCULA: Q IGUAL CONJUNTO_ABRE NOMBRE_ESTADO: q_par0 COMA NOMBRE_ESTADO: q_impar0 CONJUNTO_CIERRA
SIGMA IGUAL CONJUNTO_ABRE SIMBOLO: 0 COMA SIMBOLO: 1 CONJUNTO_CIERRA
COMENTARIO_LINEA
NOMBRE_MAYUSCULA: F IGUAL CONJUNTO_ABRE NOMBRE_ESTADO: q_par0 CONJUNTO_CIERRA
COMENTARIO_LINEA
COMENTARIO_LINEA

FUNCION PARENTESIS_ABRE NOMBRE_ESTADO: q_par0 COMA SIMBOLO: 0 PARENTESIS_CIERRA
IGUAL NOMBRE_ESTADO: q_impar0
FUNCION PARENTESIS_ABRE NOMBRE_ESTADO: q_par0 COMA SIMBOLO: 1 PARENTESIS_CIERRA
IGUAL NOMBRE_ESTADO: q_par0
FUNCION PARENTESIS_ABRE NOMBRE_ESTADO: q_impar0 COMA SIMBOLO: 0 PARENTESIS_CIERRA
IGUAL NOMBRE_ESTADO: q_par0
FUNCION PARENTESIS_ABRE NOMBRE_ESTADO: q_impar0 COMA SIMBOLO: 1 PARENTESIS_CIERRA
IGUAL NOMBRE_ESTADO: q_impar0
```

Nota: para depurar el lexer, puede ser útil ejecutar

grun LexerAF tokens -tokens <pruebaAFD.af

la opción -tokens permite visualizar los tokens reconocidos. Para este archivo de ejemplo tendríamos la siguiente salida:

```
@1,4:15='Acepta0Pares',<NOMBRE_MAYUSCULA>,1:4]
[@2,17:17='',<'='>,1:17]
[@3,19:19='(',<'('>,1:19]
[@4,20:20='Q',<NOMBRE_MAYUSCULA>,1:20]
[@5,21:21=',',<','>,1:21]
[@6,23:28='\Sigma',<'\Sigma'>,1:23]
[@7,29:29=',',<','>,1:29]
[@8,31:31='f',<FUNCION>,1:31]
[@9,32:32=',',<','>,1:32]
[@10,34:39='q_par0',<NOMBRE_ESTADO>,1:34]
[@11,40:40=',',<','>,1:40]
[@12,42:42='F',<NOMBRE_MAYUSCULA>,1:42]
[@13,43:43=')',<')'>,1:43]
[@14,45:51='@autor:',<'@autor:'>,2:0]
[@15,53:60='Silvio',<NOMBRE_AUTOR>,2:8]
[@16,62:68='@fecha:',<'@fecha:'>,3:0]
[@17,70:79='11/04/2023',<FECHA>,3:8]
[@18,81:89='@version:',<'@version:'>,4:0]
[@19,91:100='1.0.0-beta',<VERSION>,4:10]
[@20,102:141='// Definición del conjunto de estados Q\n',<COMENTARIO_LINEA>,5:0]
[@21,142:142='Q',<NOMBRE_MAYUSCULA>,6:0]
[@22,143:143='',<'='>,6:1]
[@23,145:145='{',<'{'>,6:3]
[@24,146:151='q_par0',<NOMBRE_ESTADO>,6:4]
[@25,152:152=',',<','>,6:10]
[@26,154:161='q_impar0',<NOMBRE_ESTADO>,6:12]
[@27,162:162='}',<'}'>,6:20]
[@28,164:169='\Sigma',<'\Sigma'>,7:0]
[@29,170:170='',<'='>,7:6]
[@30,172:172='{',<'{'>,7:8]
[@31,173:173='0',<SIMBOLO>,7:9]
[@32,174:174=',',<','>,7:10]
```

```
[@33,176:176='1',<SIMBOLO>,7:12]
[34,177:177='}',<'>,7:13]
[35,179:205="// definición del alfabeto\n",<COMENTARIO_LINEA>,7:15]
[36,206:206='F',<NOMBRE_MAYUSCULA>,8:0]
[37,207:207='=',<'='>,8:1]
[38,209:209='{',<'{'>,8:3]
[39,210:215='q_par0',<NOMBRE_ESTADO>,8:4]
[40,216:216='}',<'>,8:10]
[41,218:269="// definición del conjunto de estados de aceptación\n",<COMENTARIO_LINEA>,8:12]
[42,270:314="// Definición de la función de transición f \n",<COMENTARIO_LINEA>,9:0]
[43,315:315='f',<FUNCION>,10:0]
[44,316:316='(',<'('>,10:1]
[45,317:322='q_par0',<NOMBRE_ESTADO>,10:2]
[46,323:323=',',<','>,10:8]
[47,324:324='0',<SIMBOLO>,10:9]
[48,325:325=')',<')'>,10:10]
[49,326:326='=',<'='>,10:11]
[50,328:335='q_impar0',<NOMBRE_ESTADO>,10:13]
[51,337:337='f',<FUNCION>,11:0]
[52,338:338='(',<'('>,11:1]
[53,339:344='q_par0',<NOMBRE_ESTADO>,11:2]
[54,345:345=',',<','>,11:8]
[55,346:346='1',<SIMBOLO>,11:9]
[56,347:347=')',<')'>,11:10]
[57,348:348='=',<'='>,11:11]
[58,350:355='q_par0',<NOMBRE_ESTADO>,11:13]
[59,357:357='f',<FUNCION>,12:0]
[60,358:358='(',<'('>,12:1]
[61,359:366='q_impar0',<NOMBRE_ESTADO>,12:2]
[62,367:367=',',<','>,12:10]
[63,369:369='0',<SIMBOLO>,12:12]
[64,370:370=')',<')'>,12:13]
[65,371:371='=',<'='>,12:14]
[66,373:378='q_par0',<NOMBRE_ESTADO>,12:16]
[67,380:380='f',<FUNCION>,13:0]
[68,381:381='(',<'('>,13:1]
[69,382:389='q_impar0',<NOMBRE_ESTADO>,13:2]
[70,390:390=',',<','>,13:10]
[71,392:392='1',<SIMBOLO>,13:12]
[72,393:393=')',<')'>,13:13]
[73,394:394='=',<'='>,13:14]
[74,396:403='q_impar0',<NOMBRE_ESTADO>,13:16]
[75,406:405='<EOF>',<EOF>,14:0]
```