

# Práctica PL

Elena Kaloyanova Popova y Álvaro Borja Velasco García

2018

# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. Fase 1: Analizador léxico</b>	<b>3</b>
2.1. Clases Léxicas . . . . .	3
2.2. Especificación Formal . . . . .	4
2.3. Diseño . . . . .	6

# Capítulo 1

## Introducción

Esta práctica consistirá en el desarrollo de un procesador de lenguajes sobre el siguiente lenguaje:

## Capítulo 2

# Fase 1: Analizador léxico

### 2.1. Clases Léxicas

Todo programa consta de dos secciones: una para las declaraciones y otra para las instrucciones, separadas por un token «&&». La sección de declaraciones está formada por una serie de declaraciones compuestas por el nombre de tipo y el de variable y separadas por un punto y coma. La sección de instrucciones, por su parte, consta de una serie de asignaciones (variable=expresión), separadas también por un punto y coma. Las clases léxicas que hemos considerado para representar los tokens del lenguaje son las siguientes:

- **SEC:** Representa el seccionador de las dos partes del programa («&&»).
- **NUM:** Palabra reservada «num».
- **BOOL:** Palabra reservada «bool».
- **VAR:** Representa el nombre de la variable. Comienza necesariamente por una letra, seguida por una secuencia de cero o más letras, dígitos o el símbolo «\_».
- **ASIG:** Representa el signo igual de las asignaciones.
- **TRUE:** Palabra reservada «true».
- **FALSE:** Palabra reservada «false».
- **NUM:** Representa un número real. Puede empezar opcionalmente con un signo seguido de una secuencia de uno o más dígitos cualesquiera,

pudiendo poner ceros no significativos a la izquierda. Puede opcionalmente estar seguido por una parte decimal y/o una parte exponencial.

- **MAS:** Operador suma (+).
- **MENOS:** Operador resta (-).
- **POR:** Operador multiplicación (\*).
- **DIV:** Operador división (/).
- **AND:** Palabra reservada «and».
- **OR:** Palabra reservada «or».
- **NOT:** Palabra reservada «not».
- **MAY:** Operador mayor (>).
- **MEN:** Operador menor (<).
- **MAYI:** Operador mayor o igual (>=).
- **MENI:** Operador menor o igual (<=).
- **IGUAL:** Operador igual a (==).
- **DIST:** Operador distinto a (!=).
- **PAP:** Signo de apertura de paréntesis.
- **PCI:** Signo de cierre de paréntesis.
- **SEP:** Separadores: espacio en blanco, tabulador, retorno de carro, salto de línea y backspace.

## 2.2. Especificación Formal

Las definiciones regulares correspondientes a las clases léxicas definidas son:

- **SEC:** &&
- **VAR:** LETRA([LETRA|DIG, \j\*)**LETRA:**([a - z, A - Z])

- **NUM:** ([n][u][m])
- **BOOL:** ([b][o][o][l])
- **TRUE:** ([t][r][u][e])
- **FALSE:** ([f][a][l][s][e])
- **NUM:** SIGNO?(DIG+(DEC)?(EXP)?)
- **SIGNO:** [+,-]
- **DIG:** [0-9]
- **DEC:** .(DIG+)
- **EX:** [e|E](SIGNO?DIG+(DEC)?)
- **AND:** ([a][n][d])
- **OR:** ([o][r])
- **NOT:** ([n][o][t])
- **MAS:** (\+)
- **MENOS:** (\-)
- **DIV:** (\/)
- **POR:** (\\*)
- **MAY:** (>)
- **MEN:** (<)
- **MAYI:** ([>][=])
- **MENI:** ([<][=])
- **IGUAL:** ([=][=])
- **DIST:** (![=])
- **PAP:** («»)
- **PCIERRE:** («»)
- **SEP:** [BLANCO,TAB,ENDL,CARRO,BACK]

### **2.3. Diseño**

El autómata que reconocería el lenguaje es el siguiente: