

Universidad de Antioquia  
Departamento de Ingeniería de Sistemas  
2508508 / Teoría de lenguajes y laboratorio  
Semestre 2020 - I  
Práctica #2  
Fecha de entrega: 18/01/2021

## INFORME DE ANÁLISIS DEL MINI-COMPILADOR V2

### Índice:

1. Introducción
2. Aspectos generales
3. Gramática
4. Estructuras
5. Aplicación de la gramática

### 1. Introducción:

Por medio del conocimiento en gramáticas se construirá un autómata que reconocerá fragmentos de código y determinará si la sintaxis es válida o no. En este documento se especificarán los detalles del programa, como el lenguaje de programación utilizado, las gramáticas, los autómatas y cómo se implementó todo esto en el código.

### 2. Aspectos Generales:

El programa a realizar será un Mini-compilador, se tratara de la segunda versión de uno anteriormente implementado con el uso de autómatas finitos, en este caso se seguirá utilizando como lenguaje de programación a Java, y el modo de trabajo será similar a la versión anterior, con una lectura que se hará línea por línea y ahora con más capacidad para reconocer secuencias con llaves para los comandos condicionales.

### 3. Gramática:

La gramática que se usará es:

1.  $\langle S \rangle \rightarrow \langle ESP \rangle \langle TIPO \rangle \langle ESP \rangle ;$
2.  $\langle S \rangle \rightarrow \langle ESP \rangle \langle TIPO \rangle \langle ESP \rangle \langle NV \rangle \langle ESP \rangle \langle INSTANCIADO \rangle ; \langle ESP \rangle$
3.  $\langle S \rangle \rightarrow \langle ESP \rangle \text{if} \langle SENTENCIA \rangle \langle ELSE \rangle \langle ESP \rangle$
4.  $\langle S \rangle \rightarrow \langle ESP \rangle \text{while} \langle SENTENCIA \rangle \langle ESP \rangle$
5.  $\langle INSTANCIADO \rangle \rightarrow = \langle ESP \rangle \langle EX \rangle \langle ESP \rangle$
6.  $\langle INSTANCIADO \rangle \rightarrow \lambda$

7.  $\langle \text{SENTENCIA} \rangle \rightarrow \langle \text{ESP} \rangle (\langle \text{ESP} \rangle \langle \text{EX-LOG} \rangle \langle \text{ESP} \rangle) \langle \text{ESP} \rangle \{ \langle \text{ESP} \rangle \langle \text{BLOQUE} \rangle \langle \text{ESP} \rangle \} \langle \text{ESP} \rangle$
8.  $\langle \text{ESP} \rangle \rightarrow ' ' \langle \text{ESP} \rangle$
9.  $\langle \text{ESP} \rangle \rightarrow \lambda$
10.  $\langle \text{TIPO} \rangle \rightarrow \text{int}$
11.  $\langle \text{TIPO} \rangle \rightarrow \text{char}$
12.  $\langle \text{TIPO} \rangle \rightarrow \text{double}$
13.  $\langle \text{TIPO} \rangle \rightarrow \text{boolean}$
14.  $\langle \text{TIPO} \rangle \rightarrow \lambda$
15.  $\langle \text{NV} \rangle \rightarrow \text{Gramática de nombres de variables}$
16.  $\langle \text{EX} \rangle \rightarrow \langle \text{EX-ARIT} \rangle$
17.  $\langle \text{EX} \rangle \rightarrow \langle \text{EX-LOG} \rangle$
18.  $\langle \text{EX} \rangle \rightarrow \langle \text{EX-CHAR} \rangle$
19.  $\langle \text{EX-ARIT} \rangle \rightarrow \text{Gramática de expresiones aritméticas}$
20.  $\langle \text{EX-LOG} \rangle \rightarrow \text{Gramática de expresiones lógicas}$
21.  $\langle \text{EX-CHAR} \rangle \rightarrow \text{Gramática de expresiones de carácter}$
22.  $\langle \text{BLOQUE} \rangle \rightarrow \langle \text{S} \rangle \langle \text{BLOQUE} \rangle$
23.  $\langle \text{BLOQUE} \rangle \rightarrow \langle \text{S} \rangle$
24.  $\langle \text{ELSE} \rangle \rightarrow \text{else} \langle \text{ESP} \rangle \{ \langle \text{ESP} \rangle \langle \text{BLOQUE} \rangle \langle \text{ESP} \rangle \} \langle \text{ESP} \rangle$
25.  $\langle \text{ELSE} \rangle \rightarrow \lambda$

:: Gramática de nombres de variables ::

1.  $\langle \text{NV} \rangle \rightarrow \alpha^+$
2.  $\langle \text{NV} \rangle \rightarrow \alpha^+ \alpha^+$
3.  $\langle \text{NV} \rangle \rightarrow \alpha^+ \alpha^+ \alpha^+$
4.  $\langle \text{NV} \rangle \rightarrow \alpha^+ \alpha^+ \alpha^+ \alpha^+ \langle \text{NV} \rangle$
5.  $\langle \text{NV} \rangle \rightarrow V^+ \langle \text{NV} \rangle$
6.  $\langle \text{NV} \rangle \rightarrow vO^+ \langle \text{NV} \rangle$
7.  $\langle \text{NV} \rangle \rightarrow vol^+ \langle \text{NV} \rangle$
8.  $\langle \text{NV} \rangle \rightarrow void^+ \langle \text{NV} \rangle$
9.  $\langle \text{NV} \rangle \rightarrow N^+ \langle \text{NV} \rangle$
10.  $\langle \text{NV} \rangle \rightarrow nU^+ \langle \text{NV} \rangle$
11.  $\langle \text{NV} \rangle \rightarrow nuL^+ \langle \text{NV} \rangle$
12.  $\langle \text{NV} \rangle \rightarrow nulL^+ \langle \text{NV} \rangle$

:: Gramática de expresiones aritméticas ::

1.  $\langle E \rangle \rightarrow \langle NV \rangle \rightarrow$  Gramática de nombres de variables
2.  $\langle E \rangle \rightarrow \langle NUM \rangle$
3.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle \langle SIGNO \rangle \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
4.  $\langle E \rangle \rightarrow (\langle ESP \rangle \langle E \rangle \langle ESP \rangle) \langle ESP \rangle$
5.  $\langle SIGNO \rangle \rightarrow +$
6.  $\langle SIGNO \rangle \rightarrow -$
7.  $\langle SIGNO \rangle \rightarrow *$
8.  $\langle SIGNO \rangle \rightarrow /$
9.  $\langle SIGNO \rangle \rightarrow \%$

:: Gramática de expresiones lógicas ::

1.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle \& \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
2.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle \& \& \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
3.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle | \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
4.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle || \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
5.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle \< \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
6.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle \> \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
7.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle == \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
8.  $\langle E \rangle \rightarrow \langle E \rangle \langle ESP \rangle != \langle ESP \rangle \langle E \rangle \langle ESP \rangle$
9.  $\langle E \rangle \rightarrow \langle NV \rangle \rightarrow$  Gramática de nombres de variables

:: Gramática de expresiones de carácter ::

10.  $\langle E \rangle \rightarrow ' \alpha '$
11.  $\langle E \rangle \rightarrow \langle NV \rangle \rightarrow$  Gramática de nombres de variables

Con esta gramática se puede generar una línea de código en java con ciertas restricciones que se explicaran a continuación.

#### 4. Restricciones:

- El compilador sólo comprende líneas, es decir, que instrucciones como el if y el while deben permanecer contenidas en un solo renglón, así:  
`if ( fool > 29 ) { index = index + 22; }`
- Los bloques de código están restringidos a una única instrucción, es decir, después de un ';' no se aceptan nuevas instrucciones.
- El compilador sólo lee sintaxis por lo que una condición de un if o un while pueden ser una expresión aritmética o de carácter. Por este mismo motivo una variable de tipo int puede contener una expresión booleana por ejemplo.

## 4. Estructuras:

Es fácil inferirlas desde la gramática, pero para una mayor claridad las estructuras a analizar son:

### 1. Definición:

<tipo> <variable> ;

### 2. Instanciado:

<variable> = <expresión>

### 3. Definición + Instanciado:

<tipo> <variable> = <expresión> ;

### 4. If:

if (<expresión>) { <bloque> }

### 5. If-else:

if (<expresión>) { <bloque> } else { <bloque> }

### 6. While:

while (<expresión>) { <bloque> }

## 4. Tipos de dato:

Se han seleccionado los siguientes:

- int
- char
- boolean
- double

Nota: el hecho de que double sea un tipo de dato no implica que el compilador lea datos decimales.

## 5. Aplicación de la gramática:

Este proyecto abarca la gramática de una forma poco convencional, se basa en los símbolos no terminales para generar funciones que ejecuten una solicitud sobre la línea dependiente a su representación, esto implica que no se requiere el uso de autómatas finitos pues cada función extraída de los no terminales tiene una estructura única y no emplea ningún patrón para responder las solicitudes, simplemente las responde correctamente. Esto se abarca más a fondo en el manual técnico del proyecto.