

INFORME DE ANÁLISIS DEL MINI-COMPILADOR

Índice:

1. Introducción
2. Aspectos generales
3. Estructuras a analizar
 - 3.1. Definición
 - 3.2. Definición múltiple
 - 3.3. Instanciado
 - 3.4. Definición con instanciado
4. Restricciones de la estructura
 - 4.1. Composición de las estructuras
 - 4.2. Palabras reservadas
 - 4.3. Caracteres especiales
 - 4.4. Símbolos básicos.
5. Construcción del autómata
 - 5.1. Autómata A1 (Principal)
 - 5.2. Autómata AG1
 - 5.3. Autómata A1.1
 - 5.4. Autómata A3
 - 5.5. Autómata A3.1
 - 5.6. Autómata A3.3
 - 5.7. A4
 - 5.8. AP1

1. Introducción:

Un autómata finito es una estructura matemática en la que se puede implementar un programa básico de computador con el fin de dar respuestas de aceptación o de rechazo al final de esta estructura entre estados.

En este documento se explicará la lógica con la cual se desarrolló el autómata finito que trabajará como un pequeño compilador, que está basado en la estructuras muy básicas del lenguaje de programación Java; también están descritas las líneas y situaciones aceptadas, como las restringidas.

El autómata finito aquí explicado tiene un manual porque se encuentra plasmado en código, por lo cual solo será necesario contar con el ejecutable y una aplicación para correrlo y finalmente ver su funcionamiento real.

2. Aspectos generales:

En este trabajo se ha seleccionado el lenguaje Java, en el cual se analizará línea por línea para saber qué tal está la sintaxis del programa. La sintaxis no abarca absolutamente todo lo del lenguaje porque este es realmente grande o cargado para nuestro alcance, por esto a continuación se mostrará la limitación del software con respecto a cada línea que se desee escribir.

3. Estructuras a analizar:

Hay algo muy importante de comprender y son las limitaciones que se impondrán para este programa. Pues el programa abordará en tres grupos grandes en su estructura que son:

- Tipo de dato
- Nombre de variable
- Expresión

3.1. Definición:

<TipoDeDato> <NombreDeVariable> ;

3.2. Definición múltiple:

<TipoDeDato> <NombreDeVariable>, <...>, <NombreDeVariable> ;

3.3. Instanciado:

<NombreDeVariable> = <Expresión> ;

3.4. Definición con instanciado:

<TipoDeDato> <NombreDeVariable> = <Expresión> ;

3.5. Operación de variable:

<NombreDeVariable> <Signo> = <Expresión> ;

4. Restricciones de la estructura

4.1. Composición de las estructuras:

En cada uno de estos tres grandes grupos no se distingue de qué es lo que se escribe en realidad, por ejemplo: En el tipo de dato entero, que espera una expresión numérica, realmente no es posible que sea igualado a una expresión lógica como por ejemplo:

`int puerta = 89 > 3;`

`89 > 3` es “true” como respuesta, y true es una expresión correcta para el tipo de dato booleano. Entonces aquí es donde decimos que no se escatimaré de que tipo de dato se espera, sino de que el orden en el cual se está utilizando estos tres grandes grupos. Por lo cual, la expresión anterior en nuestro programa se aceptará.

A continuación más información sobre los tres grandes grupos:

4.1.1. Tipo De Dato:

Incluimos los siguientes:

- ☐ int
- ☐ double
- ☐ float
- ☐ char
- ☐ boolean

4.1.2. Nombre De Variable:

Se aceptarán los siguientes caracteres

- ☐ _ (guión bajo)
- ☐ 0-9 (Número del 0 al 9)
- ☐ Az (Abecedario en mayuscula y minuscula sin la ñ.)

Importante: un nombre de variable no puede comenzar con un número, puede comenzar con _ (Guión bajo), o con una letra.

Ejemplos de nombres de variables		
CORRECTOS	INCORRECTOS - razón de incorrecto	
_hojas	_ (Guión bajo)	No se permiten secuencias de guiones bajos únicamente.
Contador2	2contador	No puede iniciar con número

entero2caracter	año	No se permite la letra ñ
bonit_o	int	Palabra reservada

4.1.3. Expresión:

La expresión es la más compleja, pues está compuesta por más expresiones.

Para llegar a este análisis desde los autómatas, se tuvo en cuenta que una <expresión> puede estar antecedita por un signo, así

<expresión> → <dato>

<expresión> → <operador signo> <expresión>

<expresión> → <expresión> <operador> <expresión>

Las expresiones también permiten paréntesis.

Por eso a continuación se verán los caracteres o integrantes de cada subgrupo.

<dato>

Puede ser:

- ☐ Nombre de variable.
- ☐ Número.
- ☐ Dato booleano.
- ☐ Caracter.

<operador>

Puede ser:

- ☐ +
- ☐ -
- ☐ *
- ☐ /
- ☐ %
- ☐ >
- ☐ >=
- ☐ <
- ☐ <=
- ☐ ==
- ☐ !=
- ☐ |
- ☐ ||
- ☐ &
- ☐ &&

<operador signo>

Puede ser:

- ☐ +
- ☐ -
- ☐ !

Casos especiales:	
paréntesis ()	El número de caracteres abre y cierra paréntesis debe ser el mismo, además de tener un orden lógico. Para la ubicación de los paréntesis, se utilizó la misma lógica que en Java.
<operador signo> + - !	Acepta los operadores signo '+' y '-' consecutivos <u>si están separados por espacio</u> o espacios, así: int a = - + - - + + 2; int b = !!!!!8; c += !- -!!x;

4.2. Palabras reservadas:

Las palabras reservadas que seleccionamos son:

- ☐ bool
- ☐ char
- ☐ int
- ☐ double
- ☐ float
- ☐ if
- ☐ else
- ☐ false
- ☐ true

4.3. Caracteres especiales:

De esta categoría hacen parte todos los símbolos que no son palabras reservadas, símbolos básicos, números, ni letras del alfabeto, como por ejemplo los emojis o las letras con acento. Estos caracteres no son válidos en ningún caso, y cualquier línea que los contenga retornará un error.

4.4. Símbolos básicos:

Nota: el símbolo de espacio se representa con \Box

\Box	\Box	(espacio)
\Box	;	(punto y coma)
\Box	,	(coma)
\Box	.	(punto)
\Box	+	(más)
\Box	-	(menos)
\Box	/	(slash)
\Box	*	(por)
\Box	=	(igual)
\Box	>	(mayor que)
\Box	<	(menor que)
\Box	!	(signo de admiración de cierre)
\Box	((abre paréntesis)
\Box)	(cierra paréntesis)
\Box	&	(et)
\Box		(pleca)

5. Construcción del autómata:

Hay dos tipos de autómatas implementados, uno es el autómata principal (A1), y otro es el autómata Pila (AP1), los cuales se llaman cada vez que se lee una nueva línea.

Hay una representación de caracteres o grupo de caracteres para todos los autómatas y es la siguiente:

DEFINICIONES

AZ: Cualquier letra del alfabeto, mayúsculas, minúsculas y sin tildes.

#: Cualquier caracter número

\Box : espacio.

Θ : Cualquier carácter diferente a los símbolos de entrada del autómata.

5.1. Autómata A1 (Principal):

El autómata 1 inicia leyendo lo que se encuentra al inicio de todas las líneas. Inicialmente este, con ayuda de otros autómatas de los que más adelante se hablará, espera leer un tipo de dato o un nombre de una variable, para una correcta definición de variable o instanciado, estos responden con un estado del A1, también pueden responder con error.

Cuando responde con un estado del A1 es porque leyó una definición o un nombre de variable, aquí él también se apoya de otros autómatas los cuales le enviarán un estado A, rechazo (error) según sea el caso.

Finalmente cuando estos responden al estado A de A1, este autómata se comunica con el Autómata de pila (Autómata que analiza los paréntesis) para que este mande el estado de Aceptación o de Rechazo según el estado de la pila.

SÍMBOLOS PERMITIDOS

AZ	#	_	⌘	=	+	-	*	/	%	,	;	θ	¬
----	---	---	---	---	---	---	---	---	---	---	---	---	---

ESTADOS

- **S** Estado inicial.
- **D** Se leyó un tipo de dato.
- **NV** Se está leyendo un posible nombre de variable.
- **N1** Después de que lee un nombre de variable en una definición.
- **N2** Después de leer un nombre de variable en un instanciado.
- **I** Se espera un igual '='
- **I+** Se espera un signo más '+' o un igual '='
- **I-** Se espera un signo menos '-' o un igual '='
- **P** Se espera un punto y coma ';'.
- **A** Estado en el que se leyó el punto y coma final.

ESTADOS DE ERROR

- **EC** Uso de caracter inválido.
- **EP** Uso de palabra reservada.
- **ES** Signos consecutivos.
- **EO** Se esperaba un operador.
- **EOA** Se esperaba un operador aritmético.
- **E1** Se esperaba algún tipo de variable.
- **E2** Se esperaba un nombre de variable.
- **E3** Se esperaba un separador ','
- **E4** Se esperaba un igual '='
- **E5** Se esperaba un más '+'
- **E6** Se esperaba un menos '-'
- **E7** Se esperaba un asterisco '*'
- **E8** Se esperaba un slash '/'
- **E9** Se esperaba un punto y coma ';'.
- **E10** Se esperaba fin de línea '¬'
- **E11** Se esperaba una expresión.

- **E12** Se esperaba un punto '.'
- **E13** Se esperaba una comilla '
- **E14** Se esperaba un '&'
- **E15** Se esperaba un '|'
- **E16** Hace falta cierre paréntesis ')'

	AZ	#	_	☒	=	+	-	*	/	%	,	;	Θ	¬
S	S(A1.1)	E{1,2}	S2(A3)	S	E{1,2}	E{1,2}	E{1,2}	E{1,2}	E{1,2}	E{1,2}	E{1,2}	E{1,2}	E{1,2}	AP1{¬}
D	S1(A3)	E2	S1(A3)	D	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2
NV	S2(A3)	S2(A3)	S2(A3)	NV	S(A4)	I+	I-	I	I	I	E2	E2	E2	E2
N1	E{3,4,9}	E{3,4,9}	E{3,4,9}	N1	S(A4)	E{3,4,9}	E{3,4,9}	E{3,4,9}	E{3,4,9}	E{3,4,9}	D	A	E{3,4,9}	E{3,4,9}
N2	EOA	EOA	EOA	N2	S(A4)	I+	I-	I	I	I	EOA	EOA	EOA	EOA
I	E4	E4	E4	I	S(A4)	E4	E4	E4	E4	E4	E4	E4	E4	E4
I+	E5	E5	E5	I+	S(A4)	P	E5	E5	E5	E5	E5	E5	E5	E5
I-	E6	E6	E6	I-	S(A4)	E6	P	E6	E6	E6	E6	E6	E6	E6
P	E9	E9	E9	P	E9	E9	E9	E9	E9	E9	E9	A	E9	E9
A	E10	E10	E10	A	E10	E10	E10	E10	E10	E10	E10	E10	E10	AP1{¬}

5.2. Autómata AG1:

Es un autómata generalizado, lo que hace es desligar a otros autómatas que funcionan muy similar porque utilizan los mismos estados, pero diferente respuesta de acuerdo al Autómata que lo llamó. Se utiliza para detectar palabras reservadas. Este autómata tiene parámetros, que es el lugar desde donde se llamó, hace el recorrido en sus estados apoyándose de los otros autómatas que se desligan de este y de acuerdo al valor que alguno de estos le responda es como va a responder a quien lo invocó.

Se ve la necesidad de realizar este autómata porque se el autómata de palabras reservadas se necesita en varias ocasiones en el programa, pero de acuerdo al lugar en donde se esté, esta palabra será válida o no.

Por ejemplo:

la palabra int que es tipo de dato se puede al inicio de una línea, pero esta palabra no se puede aceptar cuando el programa está leyendo una expresión.

SÍMBOLOS PERMITIDOS

a	b	c	d	e	f	h	i	l	n	o	r	s	t	u	☒	Θ
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ESTADOS

- **S** Estado inicial
- **bool1** Ingresa la b de bool
- **bool2** Ingresa la primera o de bool
- **bool3** Ingresa la segunda o de bool
- **bool4** Ingresa la l de bool
- **char1** Ingresa la c de char
- **char2** Ingresa la h de char
- **char3** Ingresa la a de char
- **char4** Ingresa la r de char
- **double1** Ingresa la d de double
- **double2** Ingresa la o de double
- **double3** Ingresa la u de double
- **double4** Ingresa la b de double
- **double5** Ingresa la l de double
- **double6** Ingresa la e de double
- **float1-false1** Ingresa la f de float o de false
- **float2** Ingresa la l de float
- **float3** Ingresa la o de float
- **float4** Ingresa la a de float
- **float5** Ingresa la t de float
- **int1-if1** Ingresa la i de int o de if
- **int2** Ingresa la n de int
- **int3** Ingresa la t de int
- **if2** Ingresa la f de if
- **else1** Ingresa la primera e de else
- **else2** Ingresa la l de else
- **else3** Ingresa la s de else
- **else4** Ingresa la segunda e de else
- **true1** Ingresa la t de true
- **true2** Ingresa la r de true
- **true3** Ingresa la u de true
- **true4** Ingresa la e de true
- **false2** Ingresa la a de false
- **false3** Ingresa la l de false
- **false4** Ingresa la s de false
- **false5** Ingresa la e de false

	a	b	c	d	e	f	h	i	l	n	o	r	s	t	u	Θ	⊠
S1	o	bool1	char1	double1	else1	float1-false1	o	int1-if1	o	o	o	o	o	true1	o	o	o
bool1	o	o	o	o	o	o	o	o	o	o	bool2	o	o	o	o	o	o
bool2	o	o	o	o	o	o	o	o	o	o	bool3	o	o	o	o	o	o
bool3	o	o	o	o	o	o	o	o	bool4	o	o	o	o	o	o	o	o
bool4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R1
char1	o	o	o	o	o	o	char2	o	o	o	o	o	o	o	o	o	o
char2	char3	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
char3	o	o	o	o	o	o	o	o	o	o	o	char4	o	o	o	o	o
char4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R1
double1	o	o	o	o	o	o	o	o	o	o	double2	o	o	o	o	o	o
double2	o	o	o	o	o	o	o	o	o	o	o	o	o	o	double3	o	o
double3	o	double4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
double4	o	o	o	o	o	o	o	o	double5	o	o	o	o	o	o	o	o
double5	o	o	o	o	double6	o	o	o	o	o	o	o	o	o	o	o	o
double6	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R1
float1-false1	false2	o	o	o	o	o	o	o	float2	o	o	o	o	o	o	o	o
float2	o	o	o	o	o	o	o	o	o	o	float3	o	o	o	o	o	o
float3	float4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
float4	o	o	o	o	o	o	o	o	o	o	o	o	o	float5	o	o	o
float5	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R1
int1-if1	o	o	o	o	o	if2	o	o	o	int2	o	o	o	o	o	o	o
int2	o	o	o	o	o	o	o	o	o	o	o	o	o	int3	o	o	o
int3	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R1
if2	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R2
else1	o	o	o	o	o	o	o	o	else2	o	o	o	o	o	o	o	o
else2	o	o	o	o	o	o	o	o	o	o	o	o	else3	o	o	o	o
else3	o	o	o	o	o	o	o	o	else4	o	o	o	o	o	o	o	o
else4	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	X1	R2
true1	o	o	o	o	o	o	o	o	o	o	o	true2	o	o	o	o	o

true2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	true3	0	0
true3	0	0	0	0	true4	0	0	0	0	0	0	0	0	0	0	0	0	0
true4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X2	R3
false2	0	0	0	0	0	0	0	0	false3	0	0	0	0	0	0	0	0	0
false3	0	0	0	0	0	0	0	0	0	0	0	0	0	false4	0	0	0	0
false4	0	0	0	0	false4	0	0	0	0	0	0	0	0	0	0	0	0	0
false5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X2	R3

5.3. Autómata A1.1:

Este autómata se implementará por AG1, justo en el comienzo de la línea, cuando el programa perciba que puede estar ingresando un tipo de variable o posible nombre de variable. De acuerdo a lo que se vaya leyendo en el A1.1, este responde con un valor a AG1 para que AG1 se encargue de responder a quien lo invocó inicialmente que en este caso es A1.

Tiene los mismos SÍMBOLOS y ESTADOS que el autómata generalizado AG1.

Generado a partir de AG1						
Parámetros	O	X1	X2	R1	R2	R3
Valores	NV(A1)	S	S	D[A1]	EP	EP

	a	b	c	d	e	f	h	i	l	n	o	r	s	t	u	θ	⊞
S	NV(A1)	bool1	char1	double 1	else1	float1-false 1	NV(A1)	int1, if1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	true1	NV(A1)	NV(A1)	NV(A1)
bool1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	bool2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
bool2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	bool3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
bool3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	bool4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
bool4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	D[A1]
char1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	char2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
char2	char3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
char3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	char4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
char4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	D[A1]
double 1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	double 2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
double 2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	double 3	NV(A1)	NV(A1)

double 3	NV(A1)	double 4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
double 4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	double 5	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
double 5	NV(A1)	NV(A1)	NV(A1)	NV(A1)	double 6	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
double 6	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	D[A1]
float1-false1	false2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	float2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
float2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	float3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
float3	float4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
float4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	float5	NV(A1)	NV(A1)	NV(A1)
float5	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	D[A1]
int1 - if1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	if2	NV(A1)	int2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
int2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	int3	NV(A1)	NV(A1)	NV(A1)
int3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	D[A1]
if2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	EP
else1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	else2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
else2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	else3	NV(A1)	NV(A1)	NV(A1)	NV(A1)
else3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	else4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
else4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	EP
true1	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	true2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
true2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	true3	NV(A1)	NV(A1)	NV(A1)
true3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	true4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
true4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	EP
false2	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	false3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
false3	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	false4	NV(A1)	NV(A1)	NV(A1)	NV(A1)
false4	NV(A1)	NV(A1)	NV(A1)	NV(A1)	false5	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)
false5	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	NV(A1)	S	EP

5.4. Autómata A3:

Este autómata se llamará en el momento en que se vaya a leer un nombre de variable. Responde si un nombre de variable es válido o no.

SÍMBOLOS

AZ	#	-	⌘	θ
----	---	---	---	---

ESTADOS

- S1 Se leerá un nombre de variable después de tipo de dato o de una coma en la definición múltiple.
- S2 Se va a leer o está leyendo un nombre válido de variable en una instanciación.
- S3 Se leerá un nombre de variable en una expresión.
- X1 Se leyó una palabra reservada seguida de un Θ (A3.1).
- X3 Se leyó una palabra reservada seguida de un Θ (A3.3) en una expresión.
- V1 Se está leyendo un nombre válido de variable en definición.
- V3 Se está leyendo un nombre válido de variable en una expresión.
- G1 Se está en un ciclo de guiones bajos en definición.
- G2 Se está en un ciclo de guiones bajos en una instanciación.
- G3 Se está en un ciclo de guiones bajos en una expresión.
- O Se leyó un dato.

	AZ	#	_	\boxtimes	Θ
S1	S(A 3.1)		G1		
X1	V1	V1	V1		EP
V1	V1	V1	V1	N1(A1)	N1(A1)
G1	V1	V1	G1	E2	E2
S2	S2	S2	G2	N2(A1)	N2(A1)
G2	S2	S2	G2	E2	E2
S3	S(A 3.3)		G3		
X3	V3	V3	V3		EP
V3	V3	V3	V3	O(A4)	O(A4)
G3	V3	V3	G3	E2	E2
O	O(A4)	O(A4)	O(A4)	O(A4)	O(A4)

5.5. Autómata A3.1:

Este autómata se implementará por AG1, en el momento en que se está leyendo un posible nombre de variable en definición. De acuerdo a lo que se vaya leyendo en el A3.1, este responde con un valor a AG1 para que AG1 se encargue de responder a quien lo invocó inicialmente que en este caso es A1.

Tiene los mismos SÍMBOLOS y ESTADOS que el autómata generalizado AG1.

Generado a partir de AG1						
Parámetros	O	X1	X2	R1	R2	R3
Valores	V1(A3)	X1(A3)	X1(A3)	EP	EP	EP

	a	b	c	d	e	f	h	i	l	n	o	r	s	t	u	Θ	⊠
S	V1(A3)	bool1	char1	double 1	else1	float1-false 1	V1(A3)	int1, if1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	true1	V1(A3)	V1(A3)	V1(A3)
bool1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	bool2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
bool2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	bool3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
bool3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	bool4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
bool4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
char1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	char2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
char2	char3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
char3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	char4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
char4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
double 1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	double 2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
double 2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	double 3	V1(A3)	V1(A3)
double 3	V1(A3)	double 4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
double 4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	double 5	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
double 5	V1(A3)	V1(A3)	V1(A3)	V1(A3)	double 6	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
double 6	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
float1-false1	false2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	float2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
float2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	float3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
float3	float4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
float4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	float5	V1(A3)	V1(A3)	V1(A3)	V1(A3)
float5	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
int1-if 1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	if2	V1(A3)	int2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
int2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	int3	V1(A3)	V1(A3)	V1(A3)	V1(A3)
int3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
if2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
else1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	else2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
else2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	else3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)

else3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	else4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
else4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
true1	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	true2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
true2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	true3	V1(A3)	V1(A3)
true3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	true4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
true4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP
false2	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	false3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
false3	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	false4	V1(A3)	V1(A3)	V1(A3)	V1(A3)
false4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	false4	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)
false5	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	V1(A3)	X1(A3)	EP

5.6. Autómata A3.3:

Este autómata se implementará por AG1, en el momento en que se está leyendo un posible nombre de variable en una expresión. De acuerdo a lo que se vaya leyendo en el A3.3, este responde con un valor a AG1 para que AG1 se encargue de responder a quien lo invocó inicialmente que en este caso es A1.

Tiene los mismos SÍMBOLOS y ESTADOS que el autómata generalizado AG1.

Generado a partir de AG1						
Parámetros	O	X1	X2	R1	R2	R3
Valores	V3(A3)	X3(A3)	O(A3)	EP	EP	O(A3)

	a	b	c	d	e	f	h	i	l	n	o	r	s	t	u	Θ	⊞
S	V3(A3)	bool1	char1	double1	else1	float1-false1	V3(A3)	int1, if1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	true1	V3(A3)	V3(A3)	V3(A3)
bool1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	bool2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
bool2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	bool3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
bool3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	bool4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
bool4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
char1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	char2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
char2	char3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
char3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	char4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
char4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
double1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	double2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
double2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	double3	V3(A3)	V3(A3)	V3(A3)

double3	V3(A3)	double 4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
double4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	double 5	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
double5	V3(A3)	V3(A3)	V3(A3)	V3(A3)	double 6	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
double6	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
float1-false1	false2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	float2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
float2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	float3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
float3	float4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
float4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	float5	V3(A3)	V3(A3)	V3(A3)
float5	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
int1-if1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	if2	V3(A3)	int2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
int2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	int3	V3(A3)	V3(A3)	V3(A3)
int3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
if2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
else1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	else2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
else2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	else3	V3(A3)	V3(A3)	V3(A3)
else3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	else4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
else4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	X3(A3)	EP
true1	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	true2	V3(A3)	V3(A3)	V3(A3)	V3(A3)
true2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	true3	V3(A3)	V3(A3)
true3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	true4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
true4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	O(A3)	O(A3)
false2	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	false3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
false3	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	false4	V3(A3)	V3(A3)	V3(A3)
false4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	false4	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)
false5	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	V3(A3)	O(A3)	O(A3)

5.7. Autómata A4:

Este autómata se implementará por A1 en el momento en el que se deba analizar una expresión.

Lamará la colaboración de dos autómatas; uno de ellos es al autómata de Pilas que se encarga de alojar los paréntesis en la expresión y otro es al autómata A3, en donde se deberá analizar si el nombre de variable es válido. El resto de situaciones será resuelto entre los estados propios de este autómata A4.

SÍMBOLOS

AZ	_	#	☒	+	-	*	/	%	'	.	=	<	>	!	&		()	;
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	---	---	---

ESTADOS

- S Estado inicial.
- SIG Lee la expresión después de un signo.
- NE Lee de número entero
- PD Leyó un punto para el número decimal.
- ND Leyó el número decimal.
- C1 Leyó la primera comilla para dato tipo char.
- C2 Leyó carácter tipo char, leerá la segunda comilla.
- O Espera operador.
- I Espera un igual.
- AND Leyó un '&'
- OR Leyó '|'
- MM Leyó un '< o >'

Para efectos de visualización la tabla se dividió en dos.

	AZ	_	#	☒	+	-	*	/	%
S	S3(A3)	S3(A3)	NE	S	SIG	SIG	E11	E11	E11
SIG	S3(A3)	S3(A3)	NE	S	ES	ES	E11	E11	E11
NE	EO	EO	NE	O	S	S	S	S	S
PD	E12	E12	ND	E12	E12	E12	E12	E12	E12
ND	EO	EO	ND	O	S	S	S	S	S
C1	C2	C2	C2	C2	C2	C2	C2	C2	C2
C2	E13	E13	E13	E13	E13	E13	E13	E13	E13
O	EO	EO	EO	O	S	S	S	S	S
I	E4	E4	E4	E4	E4	E4	E4	E4	E4
AND	S3(A3)	S3(A3)	NE	S	SIG	SIG	E{11,14}	E{11,14}	E{11,14}
OR	S3(A3)	S3(A3)	NE	S	SIG	SIG	E{11,15}	E{11,15}	E{11,15}
MM	S3(A3)	S3(A3)	NE	S	SIG	SIG	E{4,11}	E{4,11}	E{4,11}

	'	.	=	<	>	!	&		()	;
S	C1	PD	E11	E11	E11	S	E11	E11	AP1{O}	E11	E11

SIG	C1	PD	E11	E11	E11	S	E11	E11	AP1{O}	E11	E11
NE	EO	PD	I	MM	MM	I	AND	OR	EO	AP1{X}	A(A1)
PD	E12	E12	E12	E12	E12	E12	E12	E12	E12	E12	E12
ND	EO	EO	I	MM	MM	I	AND	OR	EO	AP1{X}	A(A1)
C1	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2	C2
C2	O	E13	E13	E13	E13	E13	E13	E13	E13	E13	E13
O	EO	EO	I	MM	MM	I	AND	OR	EO	AP1{X}	A(A1)
I	E4	E4	S	E4	E4	E4	E4	E4	E4	E4	E4
AND	C1	PD	E{11,1 4}	E{11,1 4}	E{11,1 4}	S	S	E{11,1 4}	AP1{O}	E{11,1 4}	E{11,1 4}
OR	C1	PD	E{11,1 5}	E{11,1 5}	E{11,1 5}	S	E{11,1 5}	S	AP1{O}	E{11,1 5}	E{11,1 5}
MM	C1	PD	S	E{4,11 }	E{4,11 }	S	E{4,11 }	E{4,11 }	AP1{O}	E{4,11 }	E{4,11 }

5.8. Autómata AP1:

Autómata de pila para el tratamiento de paréntesis en una expresión.

Este autómata es uno de los principales, podrán acceder a él, el autómata A1 y A4; A1 porque es el autómata que recibe la respuesta a la que llegó el final de la línea, y para ello se necesita verificar si todos los paréntesis que se abrieron en expresión si se cerraron; y A4 lo implementa porque cada vez que se utilizan paréntesis en una expresión, este le responde a sus estados ya sea del mismo A4 o estado de error.

SÍMBOLOS

O	X	¬
---	---	---

ESTADOS

- O Paréntesis en espera
- ▼ Pila vacía (ESTADO DE ACEPTACIÓN)

ESTADO INICIAL:

▼

TRANSICIONES

- #1 Apila(O), S[A4], Av
- #2 Desapilar, O[A4], Av

	O	X	\neg	
O	#1	#2	E16	
▼	#1	ES10		1

La información de los autómatas en cuanto a sus estructuras como tabla se encuentran en el archivo de excel “Autómatas”.