

Automatador

Automatas Explicados

El siguiente documento es la explicación de los autómatas usados en el **Automatador** compilador lexicográfico escrito por Juan Cardona y Roy Maestre.

Esta práctica fue realizada usando varios autómatas para reconocer las diferentes estructuras léxicas que se pueden encontrar en el código. En algunos casos, los autómatas presentan una jerarquía. Al entrar una serie de símbolos, primero se intentará reconocerlos como una palabra reservada, si los autómatas de palabras reservadas fallan, el autómata de variables toma el control, y almacena la palabra si se puede considerar una variable válida. Luego están los autómatas de símbolos especiales y de números, los cuales tienen la misma jerarquía que los de palabras reservadas puesto que usan símbolos diferentes.

En el origen de los tiempos (de esta práctica) se intentó hacer un solo autómata que reconociera todas las palabras reservadas, pero este era demasiado grande para controlar, por lo que se recurrió a varios pequeños autómatas que reconocieran cada una de las palabras reservadas los cuales quedaron mas o menos así:

String:

	a	b	d	e	f	g	h	i	l	n	o	r	s	t	_
Vacio													S1		
S1														S2	
S2												S3			
S3								S4							
S4										S5					
S5						S6									
S6															S7
S7 Accept String															

Int

	a	b	d	e	f	g	h	i	l	n	o	r	s	t	_
Vacio								I1							
I1										I2					
I2														I3	
I3															I4
I4 Accept Int															

For

	a	b	d	e	f	g	h	i	l	n	o	r	_	!
Vacio		B1	D1	E1	F1			I1						ER
F1									FL2	F2				
F2												F3		
F3													F4	
F4 Acept For														

Double

	a	b	d	e	f	g	h	i	l	n	o	r	s	t	u	_
Vacio		B1	D1	E1	F1			I1					S1			
D1												D2				
D2															D3	
D3		D4														
D4									D5							
D5				D6												
D6																D7
D7 Acept Double																

Boolean

	a	b	d	e	f	g	h	i	l	n	o	_
Vacio		B1	D1	E1	F1			I1				
B1											B2	
B2											B3	
B3									B4			
B4				B5								
B5	B6											
B6									B7			
B7											B8	
B8 Acept Boolean												

If

	a	b	d	e	f	_
Vacio		B1	D1	E1	F1	
IF2						IF3
IF3 Acept If						

While

	a	b	d	e	f	g	h	i	l	n	o	r	s	t	u	w	_
Vacio		B1	D1	E1	F1			I1					S1			W1	
W2							W3										
W3								W4									
W4									W5								
W5				W6													
W6																	W7
W7 Accept While																	

Else

	a	b	d	e	f	g	h	i	l	n	o	r	s	_
Vacio		B1	D1	E1	F1			I1					S1	
E1								E2						
E2													E3	
E3				E4										
E4														E5
E5 Accept Else														

Estos autómatas se pueden observar programados en la clase Automata Reservada, en el cual se explica su lógica. La misma clase funciona para reconocer todas las palabras reservadas al instanciar un objeto de dicha clase se le debe dar una secuencia la cual es la que será reconocida por el autómata. Dicho autómata se puede usar para reconocer cualquier palabra que se considere reservada y no sólo las incluidas en el trabajo.

Luego tenemos el autómata de variables, este es bastante sencillo, se debe garantizar que la variable empiece por una letra o guión bajo. Así, en caso de que el primer símbolo sea guión bajo, se quedará en este estado siempre que entren más guiones bajos, deberá entrar una letra para que sea una variable válida. Luego de garantizar esto, puede entrar cualquier carácter alfanumérico y se aceptará la secuencia cuando entre un carácter especial diferente de guión bajo; o un espacio.

	Numero	Letra	_	Símbolo o Espacio
Vacio	Error	S1	S2	
S1	S1	S1		Acepte
S2		S1	S2	

Luego encontramos los autómatas de símbolos los cuales sirven para identificar los símbolos válidos y sus combinaciones válidas, como += o & &.

	=	+	-	*	/	%	<	>	!		&	,	x
Vacio	e=	+1	-1	e=	e=	e=	e=	e=	!1	1	&1	,1	
e=	==1												A
==1													A
+1	+=1	++1											A
++1													A
+=1													A
-1	-=1		--1										A
-=1													A
--1													A
!1	!=1												A
!=1													A
1										1			A
1													A
&1										&&1			A
&&1													A
,1													A

Aquí se presenta el automata ya reducido, ya que existen muchos símbolos los cuales pueden cambiar de significado si son seguidos de un igual (=). Así los símbolos =, *, /, %, <, > llevan a un estado llamado “Esperando igual” los cuales los cambia de significado si entra un igual, o los acepta tal y como están si entra un espacio o un carácter diferente a los usados en este autómata.

Para finalizar tenemos un autómata que garantiza la integridad de los números. Y puede identificar diferencias entre Int, Double o Float (Estos últimos se identifican por tener una f al final). Se debe tener en cuenta que el tamaño de esos números no se está considerando, y puede que se guarde un número mayor al que puede albergar un int en java.

	0	d	.	f	S_
vacio	0	n	de1		
0	0	n	de2	fl	A
n	n	n	de2	fl	A
de1	de2	de2		fl	
de2	de2	de2		fl	A
fl					A

Así, se comienza desde una secuencia vacía. Si entra un cero se va a un estado 0, luego si entra cualquier número diferente de cero se lleva a un estado entero. Si entra un punto

mientras la secuencia está vacía, se debe esperar a que entre algún número para que este se considere una secuencia válida. Por último cabe reseñar que si entra una f, deberá entrar un espacio o un símbolo especial para que dicho número sea considerado válido.

Luego de leer todos los símbolos, se genera una lista ligada con la siguiente estructura: [Tipo de entrada , Entrada]. Por ejemplo, si la primera palabra reconocida es “String” este se almacenará como [Tipo : String], de igual manera si se reconoce una entrada como 0.5f se almacenará como [float : 0.5f]. De manera análoga se hace con todo lo que se logre reconocer en el automata.

Una vez se tiene la lista con todas las palabras y símbolos reconocidos se pasa a un autómata encargado de reconocer si la estructura léxica de la línea es correcta. Este autómata es un poco más grande que los anteriores.

	S	Tipo	Variable	Int	Double	Float	Bool	String	Reservada	Separador	Asignador	Modificador	Op.	Logic Op.	Arith	Suma1	Comilla	ParenAbre	ParenCierra	Final	
Vacio	1	2		E1	E1	E1	E1	E1	E5	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E3	
Tipo	1	E4	3	E4	E4	E4	E4	E4	E5	E4	E4	E4	E4	E4	E4	E4	E4	E4	E4	E4	
Variable	2	E1	E1		E1	E1	E1	E1	E5	E1	4	5	E1	E1	6		E1	E1	E1	E1	
TV	3	E1	E1		E1	E1	E1	E1	E5	7	8		E2	E2	E2	E2	E1	E1	E1	A	
V=	4	E5	9		10	10	10	10	E5	E6	E6	E6	E6	E6	E6	E6	11	4	4	E6	
V+=	5	E5	9		10	10	10	10	E5	E6	E6	E6	E6	E6	E6	E6	11	5	5	E6	
V++	6	E7	E7		E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	E7	A	
TV,	7	E5	12		E4	E4	E4	E4	E5	E4	E4	E4	E4	E4	E4	E4	E4	E4	E4	E4	
TV=	8	E5	13		14	14	14	14	E5	E6	E6	E6	E6	E6	E6	E6	15	8	8	E6	
V=V	9	E5	E7		E7	E7	E7	E7	E5	E1	E2	E2		16	16	17	E2	E2	E2	A	
V=d	10	E5	E7		E7	E7	E7	E7	E5	E1	E2	E2		16	16	E8	E2	E2	E2	E2	
V=""	11	11	11		11	11	11	11	11	11	11	11		11	11	11	18	11	11	11	
TV,V	12	E5	E1		E1	E1	E1	E1	E5	7	8		E2	E2	E2	E2	E1	E1	E1	A	
TV=V	13	E5	E7		E7	E7	E7	E7	E5	7		E2	E2		19	19	20	E2	E2	E2	A
TV=d	14	E5	E7		E7	E7	E7	E7	E5	7	E2	E2			19	19	E8	E2	E2	E2	A
TV=""	15	15	15		15	15	15	15	15	15	15	15		15	15	15	21	15	15	15	
V=VOp	16	E5	9		10	10	10	10	E5	E6	E6	E6		E6	E6	E6	11	16	16	E6	
V=V++	17	E7	E7		E7	E7	E7	E7	E7	E7	E7	E7		E7	E7	E7	E7	E7	E7	A	
V=""asd"	18	E5	E7		E7	E7	E7	E7	E5	E1	E2	E2		16	16	E2	E2	E2	E2	A	
TV=VOp	19	E5	13		14	14	14	14	E5	E6	E6	E6		E6	E6	E6	15	19	19	E6	
TV=V++	20	E7	E7		E7	E7	E7	E7	E7	E7	E7	E7		E7	E7	E7	E7	E7	E7	A	
TV=""asd'	21	E5	E7		E7	E7	E7	E7	E5	7	E2	E2		19	19	E2	E2	E2	E2	A	

Los nombres de los estados son bastante directos de la secuencia que se tiene:

T : Tipo

V : Variable

= : Igual

+= : Alberga todos los modificadores del estado -=, *=, %=, etc.

++ : Alberga el ++ y el --

“ : “

Op: Operador aritmético o lógico.

, : ,

“asd” : Cualquier secuencia de caracteres entre comillas.

La A representa el estado de aceptación y los estados que comienza por E son diferentes tipos de errores basados en los que enseña el IDE IntelliJ de java al escribir diferentes secuencia de código erróneas. Estos se presentan a continuación:

Not a statement	E1
Unexpected Token	E2
Unnecessary semi-colon	E3
Identifier Expected	E4
Reserved word	E5
Expresion expected	E6
Semi-colon expected	E7
Variable expected	E8
Missing "	E9

Cualquier otra duda que surja aquí, refierase la sección de restricciones de la descripción general de la práctica.