

GRAMÁTICAS

Ruth Nohemy Ardón Lechuga

Carnet: 201602975

EXPRESIONES REGULARES

decimal = 'd+\.\d+'

entero = 'd+'

label = [a-zA-Z_][a-zA-Z_0-9]*

cadena = ('\".*?\")|('.*?')

comentario multilínea = \#\"*(.\\n)*?\"\\#

comentario simple = \#. *\\n

temporal = \\$t\d+

parámetro = \\$a\d+

retorno = \\$v\d+

pila = \\$s\d+

PRECEDENCIA UTILIZADA

Nivel	Operador	Descripción	Asociatividad
11	[]	acceso elemento arreglo	izquierda
10	- !~	menos unario, not	derecha
9	• / % abs	multiplicativas	izquierda
8	+ -	aditivas y concatenación	izquierdo
7	< <= > >=	relacionales	no asociativo
6	== !=	igualdad	izquierda
5	xor ^	xor	izquierda
4	& &&	and	izquierda
3		or	izquierda
2	<< >>	Pasos	izquierda
1	=	asignación	izquierda

TERMINALES

1	PRINT	27	<=
2	GOTO	28	>=
3	EXIT	29	<
4	IF	30	>
5	READ	31	==
6	INT	32	!
7	FLOAT	33	&&
8	CHAR	34	
9	ABS	35	~
10	ARRAY	36	&
11	UNSET	37	
12	XOR	38	^
13	;	39	!=
14	:	40	\$ra
15	(41	\$sp
16)		
17	[
18]		
19	=		
20	+		
21	-		
22	*		
23	/		
24	%		
25	<<		
26	>>		

NO TERMINALES ASCENDENTE

	No terminal	Descripción
1	init	No terminal de inicio de la gramática.
2	labels	No terminal para declarar una lista de labels, este acepta uno o más label.
3	label	No terminal que deriva en la definición de una etiqueta.
4	instrucciones	No terminal para declarar una lista de instrucciones, acepta una o más instrucciones.
5	instruccion	No terminal que deriva en todas las posibles opciones de instrucción
6	print_inst	No terminal que define la gramática de una instrucción print.
7	goto_inst	No terminal que define la gramática de una instrucción goto.
8	exit_inst	No terminal que define la gramática de una instrucción exit.
9	unset_inst	No terminal que define la gramática de una instrucción unset.
10	if_inst	No terminal que define la gramática de una instrucción if.
11	asig_inst	No terminal que define la gramática de una instrucción de tipo asignación.
12	asig_array_inst	No terminal que define la gramática de una instrucción de tipo asignación de arreglo.
13	accesos	No terminal para declarar una lista de accesos, acepta uno o más accesos.
14	acceso	No terminal que define la gramática de un acceso.
15	asignable	No terminal que tiene como opción todos los registros asignables.
17	expresion	No terminal que define todas las posibles opciones para una expresión aritmetica, lógica, relacional, bit a bit o de acceso.
18	expresion_simple	No terminal que define todas las expresiones simples; ya sea entero, decimal, cadena, temporal.

GRAMÁTICA ASCENDENTE

init → labels

labels → labels label
| label

label → exp_label : instrucciones

instrucciones → instrucciones instruccion
| instruccion

instruccion → print_inst
| goto_inst
| exit_inst
| unset_inst
| if_inst
| asig_inst
| asig_array_inst

print_inst → PRINT (expresion_simple) ;

goto_inst → GOTO exp_label ;

exit_inst → EXIT ;

unset_inst → UNSET (assignable) ;

if_inst → IF (expresion) instruccion

asig_inst → assignable = expresion ;
| assignable = (INT) expresion_simple ;
| assignable = (FLOAT) expresion_simple ;
| assignable = (CHAR) expresion_simple ;
| assignable = read () ;
| assignable = ARRAY () ;

asig_array_inst → assignable accesos = expresion ;

accesos → accesos acceso
| acceso

acceso → [expresion_simple]

assignable → temporal

- | parametro
- | retorno
- | pila
- | ra
- | sp

expresion → expresion_simple + expresion_simple

- | expresion_simple - expresion_simple
- | expresion_simple * expresion_simple
- | expresion_simple / expresion_simple
- | expresion_simple % expresion_simple
- | expresion_simple && expresion_simple
- | expresion_simple || expresion_simple
- | expresion_simple XOR expresion_simple
- | expresion_simple & expresion_simple
- | expresion_simple | expresion_simple
- | expresion_simple ^ expresion_simple
- | expresion_simple << expresion_simple
- | expresion_simple >> expresion_simple
- | expresion_simple == expresion_simple
- | expresion_simple != expresion_simple
- | expresion_simple > expresion_simple
- | expresion_simple < expresion_simple
- | expresion_simple >= expresion_simple
- | expresion_simple <= expresion_simple
- | expresion_simple

expresion_simple → - expresion_simple

- | ABS (expresion_simple)
- | ! expresion_simple
- | ~ expresion_simple
- | assignable accesos
- | temporal
- | parametro
- | retorno
- | pila
- | ra
- | sp
- | entero
- | decimal
- | cadena

NO TERMINALES DESCENDENTE

	No terminal	Descripción
1	init	No terminal de inicio de la gramática.
2	labels	No terminal para declaración de labels
3	labels'	No terminal para eliminar recursividad por la izquierda y declarar una lista de uno o más labels.
4	label	No terminal que define la sintaxis de un label
5	instrucciones	No terminal para declaración de instrucciones.
6	instrucciones'	No terminal para eliminar recursividad por la izquierda y declarar una lista de uno o más instrucciones.
7	instruccion	No terminal que define todas las posibles opciones de una instrucción.
8	print_inst	No terminal que define la sintaxis de la instrucción print.
9	goto_inst	No terminal que define la sintaxis de la instrucción goto.
10	exit_inst	No terminal que define la sintaxis de la instrucción exit.
11	unset_inst	No terminal que define la sintaxis de la instrucción unset.
12	if_inst	No terminal que define la sintaxis de la instrucción if.
13	asig_inst	No terminal que define la sintaxis de la instrucción de asignación.
14	asig_valor	No terminal que define las posibles opciones del valor para asignación.
15	tipo	No terminal que define las posibles opciones de tipo de dato para la conversión de datos.
17	accesos	No terminal para una lista de accesos opcional.
18	acceso	No terminal que define la sintaxis de uno de los accesos.
19	asignable	No terminal que tiene como opción todos los registros asignables.
20	expresion	No terminal para definir expresiones.

21	expresion'	No terminal para eliminar ambigüedad de la gramática, ayuda a definir todas las posibles opciones de expresión lógica, bit a bit, relacional o aritmética.
22	expresion_simple	No terminal que define todas las expresiones simples; ya sea entero, decimal, cadena, temporal.

GRAMÁTICA DESCENDENTE

init → labels

labels → label labels'

labels' → label labels'
| epsilon

label → exp_label : instrucciones

instrucciones → instruccion instrucciones'

instrucciones' → instruccion instrucciones'
| epsilon

instruccion → print_inst
| goto_inst
| exit_inst
| unset_inst
| if_inst
| asig_inst

print_inst → PRINT (expresion) ;

goto_inst → GOTO exp_label ;

exit_inst → EXIT ;

unset_inst → UNSET (assignable) ;

if_inst → IF (expresion) instruccion

asig_inst → assignable accesos = asig_valor ;

asig_valor → expresion ;
| (tipo) ;
| read() ;
| array() ;

tipo → INT
| FLOAT
| CHAR

accesos → acceso accesos'
| epsilon

acceso → [expresion_simple]

asignable → temporal
| parametro
| retorno
| pila
| ra
| sp

expression → expresion_simple expresion'

expresion' → + expresion_simple
| - expresion_simple
| * expresion_simple
| / expresion_simple
| % expresion_simple
| && expresion_simple
| || expresion_simple
| XOR expresion_simple
| & expresion_simple
| | expresion_simple
| ^ expresion_simple
| << expresion_simple
| >> expresion_simple
| == expresion_simple
| != expresion_simple
| > expresion_simple
| >= expresion_simple
| <= expresion_simple
| epsilon

expresion_simple → - expresion_simple
| ABS (expresion_simple)
| ! expresion_simple
| ~ expresion_simple
| assignable accesos
| entero
| decimal
| cadena