

GRAMÁTICAS

Ruth Nohemy Ardón Lechuga

Carnet: 201602975

EXPRESIONES REGULARES

decimal = 'd+\.\d+'

entero = 'd+'

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

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

caracter = ('.*?\')

comentario multilínea = *(.\\n)*?*\\

comentario simple = \\/.*\\n

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	INT	27	{	53	^=
2	CHAR	28	}	54	!
3	DOUBLE	29	.	55	&&
4	FLOAT	30	=>	56	
5	PRINTF	31	++	57	~
6	STRUCT	32	--	58	&
7	IF	33	+	59	
8	ELSE	34	-	60	^
9	SWITCH	35	*	61	!=
10	CASE	36	/	62	?
11	DEFAULT	37	%		
12	WHILE	38	<<		
13	DO	39	>>		
14	FOR	40	<=		
15	BREAK	41	<		
16	CONTINUE	42	>		
17	RETURN	43	==		
18	VOID	44	+=		
19	SIZEOF	45	-=		
20	;	46	*=		
21	:	47	/=		
22	,	48	%=		
23	(49	<<=		
24)	50	>>=		
25	[51	&=		
26]	52	=		

NO TERMINALES

	No terminal	Descripción
1	init	No terminal de inicio de la gramática.
2	instrucciones_globales	No terminal para declarar una lista de instrucciones, este acepta uno o más instrucciones.
3	instruccion_global	No terminal que deriva en todas las posibles opciones de instrucción global
4	declaracion	No terminal que define la gramática de una instrucción declaracion.
5	lista_asignaciones_dec	No terminal para declarar una lista de asignaciones para una declaración.
6	asignacion_dec	No terminal que deriva en todas las posibles opciones de asignación para declaración.
7	sig_asig	No terminal que deriva en todas las posibles opciones de signo para asignación.
8	tipo	No terminal que deriva en todos los posibles tipos de datos.
9	accesos	No terminal para declarar una lista de accesos, acepta uno o más accesos.
10	acceso	No terminal que define la gramática de un acceso.
11	def_struct	No terminal que define la gramática de la definición de un struct.
12	struct_list_decl	No terminal para declarar una lista de declaraciones para la definición de un struct.
13	struct_decl	No terminal que define la gramática de la definición de la declaración para un struct.
14	lista_id_struct	No terminal para declarar una lista de identificador o accesos.
15	id_struct	No terminal que deriva en todas las posibles opciones para id de un struct.
17	declaracion_struct	No terminal que define la gramática de la definición de la declaración de un struct.
18	metodo	No terminal que define la gramática de la definición de la declaración para un método o función.
19	lista_parametros	No terminal para declarar una lista de parámetros.

20	parametro	No terminal que define la gramática de la definición de un parámetro.
21	instrucciones	No terminal para declarar una lista de instrucciones, este acepta uno o más instrucciones.
22	instruccion	No terminal que deriva en todas las posibles opciones de instrucción global
23	printf	No terminal que define la gramática de la definición de la instrucción print.
24	lista_param_printf	No terminal para declarar una lista de parámetros para la instrucción print.
25	asignacion	No terminal que define la gramática de la definición de la instrucción de asignación.
26	lista_punto	No terminal para declarar una lista de accesos con el operador punto.
27	valor	No terminal que deriva en todas las posibles opciones de valor para lista_punto.
28	if	No terminal que define la gramática de la definición de la instrucción if
29	switch	No terminal que define la gramática de la definición de la instrucción switch.
30	lista_cases	No terminal para declarar una lista de cases.
31	case	No terminal que define la gramática de la definición de un case.
32	default	No terminal que define la gramática de la definición del default
33	while	No terminal que define la gramática de la definición de la instrucción while.
34	do_while	No terminal que define la gramática de la definición de la instrucción do while
35	for	No terminal que define la gramática de la definición de la instrucción for
36	for_ini	No terminal que deriva en todas las posibles opciones del valor inicial para un for.
37	for_exp	No terminal que deriva en todas las posibles opciones para expresión.

38	for_inc	No terminal que deriva en todas las posibles opciones para el incremento del for.
39	inc_dec	No terminal que define la gramática de la definición de las instrucciones de incremento y decremento.
40	expresion	No terminal que define la gramática de la definición para las expresiones.
41	llamada	No terminal que define la gramática de una llamada a función o método.
42	lista_expresiones	No terminal para declarar una lista de expresiones, este acepta uno o más instrucciones.

GRAMÁTICA

init → instrucciones_globales

instrucciones_globales → instrucciones_globales instruccion_global
| instruccion_global

instruccion_global → declaracion ;
| def_struct
| declaracion_struct ;
| metodo

declaracion → tipo lista_asignaciones_dec

lista_asignaciones_dec → lista_asignaciones_dec , asignacion_dec
| asignacion_dec

asignacion_dec → IDENTIFICADOR
| IDENTIFICADOR sig_asig expresion
| IDENTIFICADOR accesos
| IDENTIFICADOR accesos sig_asig expresion

sig_asig → = | &= | <=<= | >>= | /= | += | -= | %= | |= | *= | ^=

tipo → int | float | double | char

accesos → accesos acceso
| acceso

acceso → [expresion]

acceso → []

def_struct → STRUCT IDENTIFICADOR { struct_list_decl }

struct_list_decl → struct_list_decl struct_decl
| struct_decl

struct_decl → tipo lista_id_struct ;

lista_id_struct → lista_id_struct COMA id_struct
| id_struct

id_struct → IDENTIFICADOR
| IDENTIFICADOR accesos

declaracion_struct → STRUCT IDENTIFICADOR IDENTIFICADOR
| STRUCT IDENTIFICADOR accesos IDENTIFICADOR

metodo → VOID IDENTIFICADOR (lista_parametros) { instrucciones }
| VOID IDENTIFICADOR () { instrucciones }
| tipo IDENTIFICADOR (lista_parametros) { instrucciones }
| tipo IDENTIFICADOR () { instrucciones }

lista_parametros → lista_parametros parametro
| parametro

parametro → tipo IDENTIFICADOR
| tipo IDENTIFICADOR accesos

instrucciones → instrucciones instruccion
| instruccion

instruccion → declaracion ;
| def_struct
| declaracion_struct ;
| printf ;
| asignacion ;
| if
| switch
| while
| do_while
| for
| break;
| continue;
| return
| llamada
| inc_dec;

printf → PRINTF (CADENA , lista_param_printf) ;
| PRINTF (CADENA) ;

lista_param_printf → lista_param_print COMA expresion
| expresion

asignacion → IDENTIFICADOR sig_asig expresion
| IDENTIFICADOR accesos sig_asig expresion
| IDENTIFICADOR lista_punto sig_asig expresion
| IDENTIFICADOR accesos lista_punto sig_asig expresion

lista_punto → lista_punto . valor
| .valor

valor → IDENTIFICADOR
| IDENTIFICADOR accesos

if → IF (expresion) { instrucciones }
| IF (expresion) { instrucciones } ELSE { instrucciones }
| IF (expresion) { instrucciones } ELSE if

switch → SWITCH (expresion) { lista_cases }
| SWITCH (expresion) { lista_cases default }

lista_cases → lista_cases case
| case

case → CASE expresion : instrucciones

default → DEFAULT : instrucciones

while → WHILE (expresion) { instrucciones }

do_while → DO { instrucciones } WHILE (expresion) ;

for → FOR (for_ini ; for_exp ; for_inc) { instrucciones }

for_ini → identificador
| declaracion
| asignacion
| epsilon

for_exp → expresion
| epsilon

for_inc → asignacion
| inc_dec
| epsilon

inc_dec → ++ IDENTIFICADOR
| -- IDENTIFICADOR
| IDENTIFICADOR ++
| IDENTIFICADOR --

expresion → expresion + expresion
| expresion - expresion
| expresion * expresion
| expresion / expresion
| expresion % expresion
| expresion && expresion

- | expresion || expresion
- | expresion & expresio
- | expresion | expresion
- | expresion ^ expresion
- | expresion << expresion
- | expresion >> expresion
- | expresion == expresion
- | expresion != expresion
- | expresion < expresion
- | expresion > expresion
- | expresion <= expresion
- | expresion >= expresion
- | ~expresion
- | !expresion
- | ++IDENTIFICADOR
- | --IDENTIIFCADOR
- | & expresion
- | IDENTIFICADOR ++
- | IDENTIFICADOR --
- | IDENTIFICADOR accesos
- | IDENTIFICADOR lista_punto
- | IDENTIFICADOR accesos lista_punto
- | ENTERO
- | CADENA
- | DECIMAL
- | CARACTER
- | IDENTIFICADOR
- | llamada
- | (expresion)
- | expresion ? expresion : expresion
- | sizeof (expresion)
- | sizeof (tipo)
- | (tipo) expresion

llamada → IDENTIFICADOR ()
 | IDENTIFICADOR (lista_expresiones)

lista_expresiones → lista_expresiones COMA expresion
 | expresion