

**CENTRO DE CIENCIAS BÁSICAS**  
**DEPARTAMENTO DE SISTEMAS ELECTRONICOS**  
**ACADEMIA DE ACADEMIA DE PROGRAMACION Y SW DE SISTEMAS**

<b>Nombre del Estudiante:</b>	Andrea Guadalupe Pérez Yáñez	<b>Fecha:</b>	mayo 2, 2022
<b>Materia:</b>	Compiladores I	<b>Carrera:</b>	ISC
<b>Profesor:</b>	M. en I.A. Eduardo Serna Perez	<b>Semestre:</b>	08
<b>Periodo:</b>	( X ) Enero – Junio ( ) Agosto - Diciembre	<b>Aciertos:</b>	
<b>Tipo de Examen:</b>	Parcial: 1° ( ) 2° ( X ) 3° ( ) Otro:	<b>Calificación:</b>	

<b>I. Instrucciones: Conteste lo que se indica</b>	<b>Valor: 10</b>
----------------------------------------------------	------------------

1. Dada la siguiente gramática que identifica declaraciones de variables con asignación, escriba la derivación por la izquierda y el árbol de análisis gramatical para la siguiente cadena de *tokens* **int x=0,y=x+1,z;**

```

decl -> tipo lid ;
lid -> lid , var_decl
    | var_decl
var_decl -> id
    | id = exp
tipo -> int | float
exp -> exp addop term | term
addop -> + | -
term -> term mulop factor | factor
mulop -> * | /
factor -> ( exp ) | numero

```

①

decl = tipo id;

=> int id;

=> int var\_decl, var\_decl;

=> int id = exp, var\_decl;

=> int x = exp, var\_decl;

=> int x = exp, id;

=> int x = exp, id = exp;

=> int x = term, id = exp;

=> int x = factor, id = exp;

=> int x = numero, id = exp;

=> int x = 0, id = exp;

=> int x = 0, factor = exp;

=> int x = 0, term = exp;

=> int x = 0, factor addop term = exp;

=> int x = 0, factor + term = exp;

=> int x = 0, id + term = exp;

=> int x = 0, y + term = exp;

=> int x = 0, y + factor = exp;

=> int x = 0, y + numero = exp;

=> int x = 0, y + 1 = exp;

=> int x = 0, exp = exp;

=> int x = 0, var\_decl;

=> int x = 0 id;

=> int x = 0 i z;

decl

tipo

int

id

var\_decl

var\_decl

id

exp

x

addop

term

exp + factor

y

term

term

1

factor

factor

id

id

z

x

## II. Instrucciones: Conteste lo que se indica

Valor: 20

2. Escriba una gramática regular para expresiones booleanas que incluya las constantes **true** y **false**, los operadores lógicos **and**, **or** y **not**, además de los paréntesis. Asegúrese de darle a **or** una precedencia mas baja que **and**, y a **and** una precedencia mas baja que a **not**, además de permitir la repetición del operador **not** como la expresión booleana **not not true**. Además que contemple a los operadores relacionales **==** y **!=**. Asegúrese también de que su gramática no sea ambigua.

②	$S \rightarrow \text{Expr} \mid \text{ParenExpr}$
	$\text{Expr} \rightarrow \text{constExpr} \mid \text{NotExpr} \mid \text{RelExpr} \mid \text{AndExpr} \mid \text{OrExpr}$
	$\text{constExpr} \rightarrow \text{true} \mid \text{false}$
	$\text{NotExpr} \rightarrow \text{not Expr}$
	$\text{RelExpr} \rightarrow \text{constExpr} \text{ RelOp } \text{constExpr}$
	$\text{RelOp} \rightarrow == \mid !=$
	$\text{AndExpr} \rightarrow \text{Expr} \text{ and } \text{Expr}$
	$\text{OrExpr} \rightarrow \text{Expr} \text{ or } \text{Expr}$
	$\text{ParenExpr} \rightarrow '(' \text{Expr} ')'$

### III. Instrucciones: Conteste lo que se indica

Valor: 20

Considere las gramáticas:

$S \rightarrow (L) \mid \text{id}$

$L \rightarrow L, S \mid S$

- Elimine la recursión por la izquierda
- Construya el conjunto primero y siguiente
- Construya la tabla de análisis sintáctico LL(1) para la gramática
- Describa las acciones de LL(1) para la siguiente entrada  $(m, x, y)$

③

a)  $S \rightarrow (L) \mid id$   
 $L \rightarrow SL'$   
 $L' \rightarrow , SL' \mid \epsilon$

b)  $\text{Primero}(S) = \{ (, id \}$   
 $\text{Primero}(L) = \{ (, id \}$   
 $\text{Primero}(L') = \{ , , \epsilon \}$   
 $\text{Siguiente}(S) = \{ ), \$ \}$   
 $\text{Siguiente}(L) = \{ , , ), \$ \}$   
 $\text{Siguiente}(L') = \{ , , ), \$ \}$

c)

	(	)	,	id	\$
S	$S \rightarrow (L)$		$S \rightarrow id$		
L	$L \rightarrow SL'$		$L' \rightarrow , SL'$	$L \rightarrow SL'$	
L'		$L' \rightarrow \epsilon$	$L' \rightarrow , SL'$		$L' \rightarrow \epsilon$



d)

Pila	Entrada	Acción
\$S	mxy\$	-
\$(L)S\$	mxy\$	$S \rightarrow (L)$
(L)	mxy\$	-
(S'L)	mxy\$	$L \rightarrow SL'$
(SL)	mxy\$	$S \rightarrow id$
(idL')	mxy\$	$L' \rightarrow , SL'$
,SL)	mxy\$	$S \rightarrow id$
(idL')	xy\$	$L' \rightarrow , SL'$
,SL)	xy\$	$S \rightarrow id$
(idL')	y\$	$L' \rightarrow \epsilon$
L')	y\$	-
,SL)	y\$	$S \rightarrow id$
L')	y\$	-
S	y\$	-
(L)	y\$	$S \rightarrow (L)$
(S'L)	y	

#### IV. Instrucciones: Conteste lo que se indica

Valor: 20

Considere la siguiente gramática (<lista> es el símbolo de inicio)

<lexp>  $\rightarrow$  <atomo> | <lista>

<atomo>  $\rightarrow$  numero | id

<lista>  $\rightarrow$  ( <lexp-sec> )

<lexp-sec>  $\rightarrow$  <lexp-sec> <lexp>  
| <lexp>

- Elimine la recursión por la izquierda
- Construya los conjuntos PIMERO y SIGUIENTE para los no terminales de la gramática
- Construya la tabla de análisis sintáctico LL(1) para la gramática resultante
- Describa las acciones de LL(1) para la siguiente entrada ( a ( m x y ) 23 )

4

a)

$\langle \text{exp-scc} \rangle \rightarrow \langle \text{exp} \rangle \langle \text{exp-scc}' \rangle$

$\langle \text{exp-scc}' \rangle \rightarrow \langle \text{exp} \rangle \langle \text{exp-scc}' \rangle \mid \epsilon$

$\langle \text{exp} \rangle \rightarrow \langle \text{atomo} \rangle \mid (\langle \text{exp-scc} \rangle)$

b)

$\text{Primero}(\langle \text{exp} \rangle) = \{ \text{numero}, \text{id}, '(' \}$

$\text{Primero}(\langle \text{atomo} \rangle) = \{ \text{numero}, \text{id} \}$

$\text{Primero}(\langle \text{listo} \rangle) = \{ '(' \}$

$\text{Primero}(\langle \text{exp-scc} \rangle) = \{ \text{numero}, \text{id}, '(' \}$

$\text{Siguiente}(\langle \text{exp} \rangle) = \{ ')', \text{fin de entrada} \}$

$\text{Siguiente}(\langle \text{atomo} \rangle) = \{ ')', \text{fin de entrada}, '(' , \text{numero}, \text{id} \}$

$\text{Siguiente}(\langle \text{listo} \rangle) = \{ ')', \text{fin de entrada} \}$

$\text{Siguiente}(\langle \text{exp-scc} \rangle) = \{ ')', \text{fin de entrada} \}$



c)  $\langle \text{no\_terminal, terminal} \rangle$  si hay ambigüedad no es LL(1)

	numero	id	'('	' )'	EOF
$\langle \text{lexp} \rangle$	$\langle \text{lexp} \rangle \rightarrow$ $\langle \text{atomo} \rangle$	$\langle \text{lexp} \rangle \rightarrow$ $\langle \text{atomo} \rangle$	$\langle \text{lexp} \rangle \rightarrow$ $\langle \text{lexp-sec} \rangle$		
$\langle \text{atomo} \rangle$	$\langle \text{atomo} \rangle \rightarrow$ numero	$\langle \text{atomo} \rangle \rightarrow$ id			
$\langle \text{listo} \rangle$			$\langle \text{listo} \rangle \rightarrow$ $\langle \text{lexp-sec} \rangle$		
$\langle \text{lexp-sec} \rangle$	$\langle \text{lexp-sec} \rangle \rightarrow$ $\langle \text{lexp} \rangle \langle \text{lexp-sec} \rangle$	$\langle \text{lexp-sec} \rangle \rightarrow$ $\langle \text{lexp} \rangle \langle \text{lexp-sec} \rangle$	$\langle \text{lexp-sec} \rangle \rightarrow$ $\langle \text{lexp} \rangle \langle \text{lexp-sec} \rangle$	$\langle \text{lexp-sec} \rangle \rightarrow$ $\langle \text{lexp} \rangle \langle \text{lexp-sec} \rangle$	$\langle \text{lexp-sec} \rangle \rightarrow$ $\langle \text{lexp} \rangle \langle \text{lexp-sec} \rangle$

d) Entrada (a1mxy123)

Pila	Entrada	Acción
\$	(a1mxy123)\$	Empujar ( $\langle \text{lexp-sec} \rangle$ )\$
$\langle \text{lexp-sec} \rangle$ \$	a1mxy123)\$	Empujar ( $\langle \text{lexp-sec} \rangle$ ) ( $\langle \text{lexp} \rangle$ )\$
) $\langle \text{lexp-sec} \rangle$ ( $\langle \text{lexp} \rangle$ )\$	a	Empujar ( $\langle \text{atomo} \rangle$ ) $\langle \text{lexp-sec} \rangle$ ( $\langle \text{lexp} \rangle$ )\$
$\langle \text{atomo} \rangle$ $\langle \text{lexp-sec} \rangle$ ( $\langle \text{lexp} \rangle$ )\$	a	Empujar id ( $\langle \text{lexp-sec} \rangle$ ) ( $\langle \text{lexp} \rangle$ )\$
id $\langle \text{lexp-sec} \rangle$ ( $\langle \text{lexp} \rangle$ )\$	(	Empujar ( $\langle \text{lexp-sec} \rangle$ ) $\langle \text{lexp-sec} \rangle$ ( $\langle \text{lexp} \rangle$ )\$
( $\langle \text{lexp-sec} \rangle$ )\$		