

Nombre:

1er Examen de Compiladores

Grupo:

Fecha:

10

1.-Defina **compilador**

2.-Que seccion de una especificacion de yacc se parece a un esquema de traduccion

- a) la de declaraciones b) la de codigo de soporte
c) la de reglas d) ninguna

(c) ✓

3.-Cual es la fase final de un **compilador** ?

- a) el análisis semantico b) el manejo de errores
c) la optimizacion de código d) la generacion de código

(d) ✓

4.-¿Cual es la **gramática** que tiene los siguientes cuatro componentes:

1. Un conjunto de **componentes léxicos**.
2. Un conjunto de **no terminales**.
3. Un conjunto de **producciones**, en el que cada producción consta de un no terminal, llamado lado izquierdo de la producción, una flecha y una secuencia de componentes léxicos y no terminales, o ambos, llamado lado derecho de la producción.
4. La denominación de uno de los no terminales como **símbolo inicial**.

- a) Gramática Asociativa por la izquierda b) Gramática *ambigua*
c) Gramática libre de contexto d) Gramática *recursiva*

(c) ✓

5.-Es una **gramática** donde existe una **cadena** (de componentes léxicos) que puede tener **más de un árbol de análisis sintáctico**.

- a) Gramática libre de contexto b) Gramática regular
c) Gramática ambigua d) Gramática Asociativa por la izquierda

(c) ✓

6.- **%token** sirve para indicar

- a) inicio de la sección de declaraciones d) los no terminales de la gramática
c) fin de la sección de declaraciones d) los terminales de la gramática

(d) ✓

7.-Como le indica el analizador léxico (**yyllex**) al analizador sintáctico (**yyparse**) que ya no hay mas tokens en la entrada

- a) retornando -1 b) retornando cero
c) almacenando -1 en yyval d) almacenando 0 en yyval

3.5 (b) 0.5 pts c/u

8.-Considere la siguiente gramática (los terminales se indican en negritas).

$P = \{ S \rightarrow (L), S \rightarrow a, L \rightarrow LS, L \rightarrow \epsilon \}$

- a) Mostrar una **derivación** de $(a)((a(a)))$.
b) Dibuje el **árbol de análisis sintáctico** para la entrada $(a)((a(a)))$.

0.5 pts

0.5 pts

9.-Considere la siguiente gramática de expresiones (los terminales se indican en negritas).

$E \rightarrow (E, E) \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

- a) Mostrar una **derivación** de $((4, 1, 6), 9)$
b) Dibuje el **árbol de análisis sintáctico** para la entrada

0.5 pts

0.5 pts

Análisis sintáctico predictivo descendente recursivo

10.-Para las gramáticas de los incisos de abajo (los terminales se indican en negritas). Escriba el **analizador sintáctico predictivo descendente recursivo**. Suponga que tiene las funciones error y para.

- a) $P = \{ S \rightarrow 111, (S) \}$ 1 pto
b) $P = \{ S \rightarrow cAd, A \rightarrow a \}$ 1.25 pts.
c) $P = \{ S \rightarrow aA, A \rightarrow bA, A \rightarrow cBd, B \rightarrow d \}$ 2 pts

13 = 0.3 + 1
14 = 0.3

Ambigüedad

11.-Demostrar que las siguientes **gramáticas** son **ambiguas**

- a) $E \rightarrow E \text{ and } E \mid E \text{ or } E \mid 0 \mid 1$
b) $S \rightarrow aS \mid aSbS \mid \epsilon$

0.75 pts
1 pto

PRODUCCIÓN	REGLA SEMÁNTICA	<p>12.- Considere la definición dirigida por la sintaxis de la izquierda (los terminales se indican en negritas).</p> <p>Dibuje el árbol de análisis sintáctico con anotaciones para la siguiente cadena</p> <p style="text-align: center;">c n e s s o o n n n e e e 1.25 pts</p>
$sec \rightarrow \text{comienza}$	$sec.x = 0$ $sec.y = 0$	
$sec \rightarrow sec_1 \text{ instr}$	$sec.x = sec_1.x + instr.dx$ $sec.y = sec_1.y + instr.dy$	
$instr \rightarrow \text{este}$	$instr.dx = 1$ $instr.dy = 0$	
$instr \rightarrow \text{norte}$	$instr.dx = 0$ $instr.dy = 1$	
$instr \rightarrow \text{oeste}$	$instr.dx = -1$ $instr.dy = 0$	
$instr \rightarrow \text{sur}$	$instr.dx = 0$ $instr.dy = -1$	

YACC

13.- Considere la siguiente gramática (los terminales se indican en negritas)

$der \rightarrow letra = der \mid letra$

$letra \rightarrow a \mid b \mid c \mid d \mid e \mid f$

0.75 pts.

Escriba la sección de reglas de la especificación de yacc para dicha gramática

14.- Considere la siguiente gramática (los terminales se indican en negritas)

$num \rightarrow 11 \mid 1001 \mid num \ 0$

a) Escriba la sección de reglas de la especificación de yacc para dicha gramática 0.75 pts.

b) Escriba las acciones gramaticales para convertir el numero de binario a decimal pts.

15.- Considere la siguiente gramática (los terminales se indican en negritas)

$lista \rightarrow lista, figura \mid figura$

$figura \rightarrow triangulo \mid cuadrilatero$

$triangulo \rightarrow lado \ lado \ lado$

$cuadrilatero \rightarrow lado \ lado \ lado \ lado$

Escriba la sección de reglas de la especificación de yacc para dicha gramática y las acciones semánticas respectivas para que se imprima si un triangulo es equilátero y si un cuadrilátero es un cuadrado
1 pto.

Análisis Sintáctico Predictivo no Recursivo

-Para las siguientes GLC construya la tabla Análisis Sintáctico Predictivo no Recursivo

-Use dicho análisis para analizar las cadenas propuestas:

-Muestre el contenido de la pila, la entrada y la acción a realizar

Problema 16.- Considere la gramática para generar paréntesis anidados

1) $A \rightarrow (A)$ 2) $A \rightarrow a$

Cadenas propuestas:

(a) ((a)) (((a))) ((((a))))

2 pts.

Problema 17.- Considere la siguiente gramática :

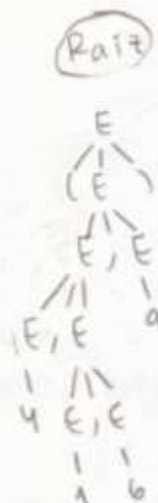
1) $S \rightarrow a$ 2) $S \rightarrow (S R$ 3) $R \rightarrow , S R$ 4) $R \rightarrow)$

Cadenas propuestas:

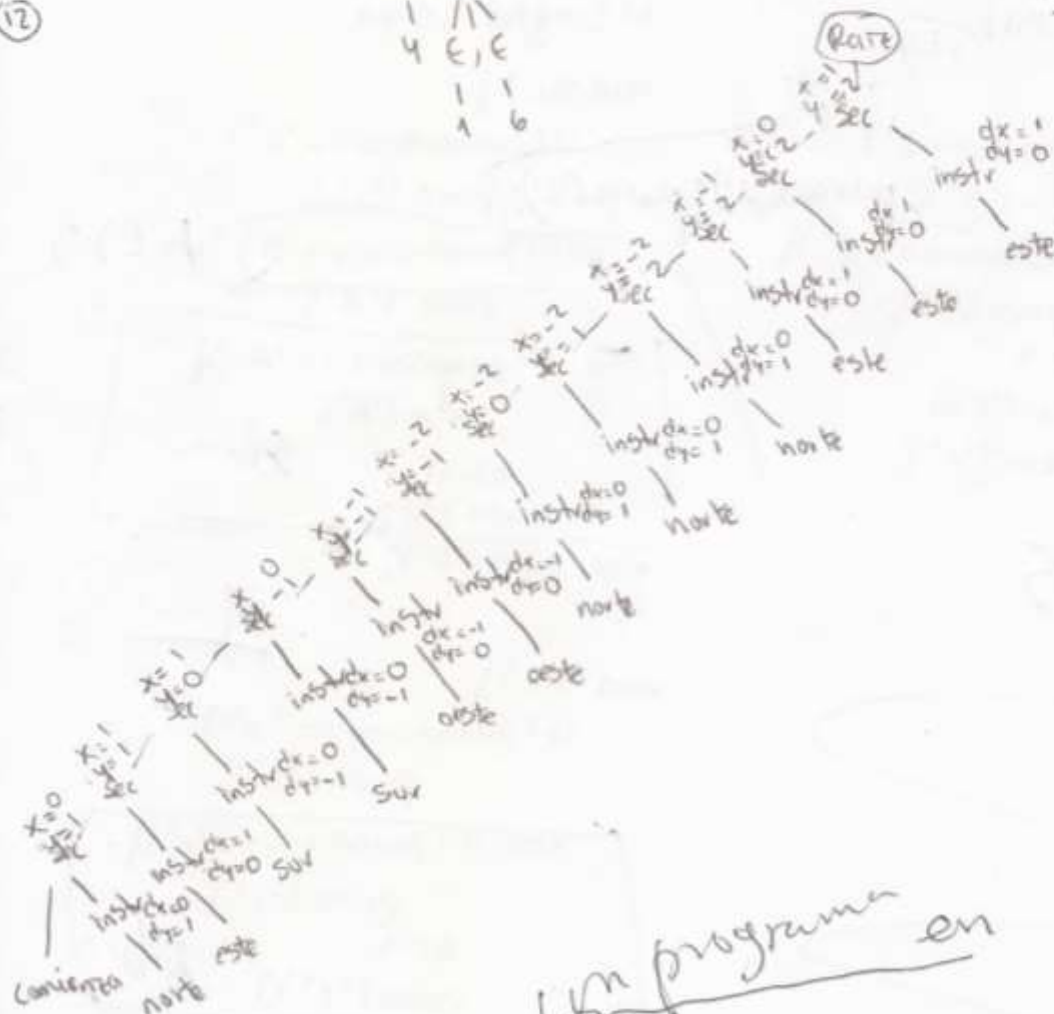
(a) (a , a) (a , a , a) (a , a , a , a)

2 pts.

⑨ $E \rightarrow (E) \rightarrow (E, E) \rightarrow ((E, E), E) \rightarrow (((E, E, E), E) \rightarrow ((4, 6), 9))$



⑫


$$\begin{matrix} x = 2 \\ y = 2 \end{matrix}$$

$x=1$
 $y=2$

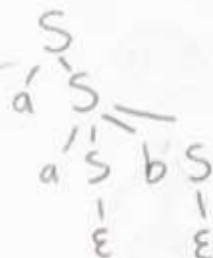
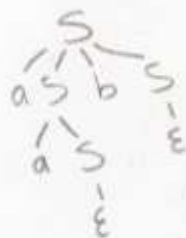
Un programme en

- ① Programa que recibe de entrada el código fuente y se encarga de traducirlo para que su salida sea en código objeto (lenguaje máquina) en programa equivalente en 1/4

④

$$\underline{b} \ S \rightarrow aS \mid aSbS \mid \epsilon$$

cadna aab



10

a) $P = \{S \rightarrow 111, \{S\}\}$

```
void sc() {
    if (preanalysis == '1')
        para('1'); para(
    else if (preanalysis == 'c') {
        para('c');
        sc()
        para('c')
    } else error('c');
```

 $0.\overline{75}$

b) $S \rightarrow cAd$, $A \rightarrow a$

```
void SL() {
    if (preanalysis == 'c')
        para('c');
    if (preanalysis == 'd')
        para('d');
    else if (preanalysis == 'A') {
        para('A');
        SL();
        para('A');
    }
    else error('A');
}
```

```
void AC(){
    if (prenatalis == 'a')
        para('a');
```

2/4

⑩

Σ $P = \{S \rightarrow aA, A \rightarrow bA, A \rightarrow CBd, B \rightarrow d\}$

```
void Sc() {
    if (preanalysis == 'a')
        para('a');
    else if (preanalysis == 'A') {
        para('A');
        Sc();
        para('A');
    }
    else error('A');
}
```

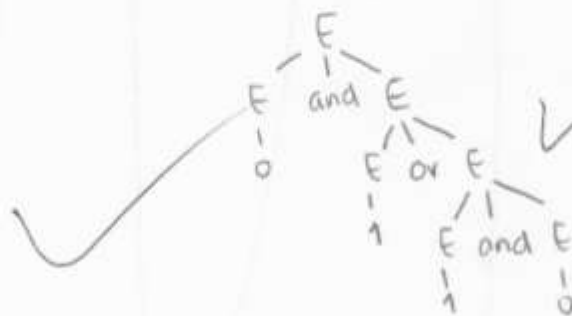
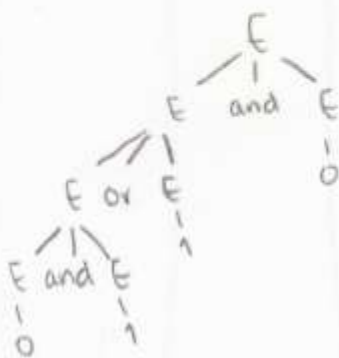
```
void A() {
    if (preanalysis == 'c' ||
        preanalysis == 'd')
        para(char);
    else if (preanalysis == 'B') {
        para('B');
        A();
        para('B');
    }
    else error('B');
}
```

```
void AC() {
    if (preanalysis == 'b')
        para('b');
    else if (preanalysis == 'A') {
        para('A');
        AC();
        para('A');
    }
    else error('A');
}
```

```
void BC() {
    if (preanalysis == 'd')
        para('d');
    else if (preanalysis == 'c') {
        para('c');
        BC();
        para('c');
    }
    else error('c');
}
```

⑪

Σ $E \rightarrow E \text{ and } E \mid E \text{ or } E \mid 0 \mid 1$ para 0 and 1 or 1 and 0



(14)

%o %o

num: '11' \$\$\$ = \$17

'1001' \$\$\$ = \$27

%o %o
num '0' \$\$\$ = \$37



11 11
11 0 0 1
num '0'

(13)

%o %o

der: letra '=' der {}

letra {}

letra: 'a' \$\$\$ = \$17

'b' \$\$\$ = \$27

'c' \$\$\$ = \$37

'd' \$\$\$ = \$47

'e' \$\$\$ = \$57

'f' \$\$\$ = \$67

%o %o

gi existe

falta ;



no existen