

# 3<sup>er</sup> Examen de Compiladores

90%

Nombre: Vargas Romero Erick Efraín

Grupo: 3CM 7

Fecha: 03/12/18

Falso o verdadero (F/V) 0.25 ptos. c/u

1.-el **arbol sintactico** no es una version condensada del **arbol de analisis sintactico** ( F )

2.-**Definición dirigida por la sintaxis** es una **GLC** en la que se encuentran intercalados, en los lados derechos de las producciones, **fragmentos de programa** llamados **acciones semánticas**. ( X )

3.-Un **atributo** es **heredado** si su valor depende de los valores de los atributos de su **padre** y/o de sus **hermanos**. ( ✓ )

4.-En lenguaje C las **variables locales** (no estáticas) se **crean** cuando se entra a una función y se **destruyen** cuando se sale de la función ( ✓ )

5.-No es posible definir **funciones recursivas** en hoc \_\_\_\_\_ ( F )

6.-En hoc no hay **variables locales** \_\_\_\_\_ ( V )

7.-En hoc los **parámetros reales** son listas de **expresiones** \_\_\_\_\_ ( VV )

8.-En hoc los **parámetros reales** se meten a la pila ( ✓ )

0.25 ptos. c/u

1.-Construir explicita o implicitamente el **grafo de dependencias**.

2.-Construir el **arbol de analisis sintactico** para la gramática y la entrada dadas.

3.-**Evaluar las reglas semanticas** de acuerdo con el **orden topologico**.

4.-Supuesto que el grafo de dependencias determina un **orden parcial** construir un **orden topologico** compatible con el orden parcial.

**Uno.**-Para la realizacion de una **Traduccion Dirigida por la Sintaxis** el orden seria

- a) 1, 2, 3 , 4      b) 2, 3 , 4 , 1      c) 2, 1 , 4, 3      d) 4, 3 , 2 , 1 ( C )

1-**Ejecutar** la función (poner el contador de programa igual a la dirección de su primera instrucción y ejecutar la instrucción a la que apunta el contador de programa) y meter el valor de retorno de la función en la pila.

2-**Meter** los **parametros** en la pila y **meter el marco de la funcion** en la pila de llamadas.

3-Poner el **contador de programa** igual a la **dirección de retorno** y saltar a la instrucción a la que apunta el contador de programa.

4-**Sacar parametros** de la pila y **sacar marco** de la pila de llamadas.

**Dos.**-De acuerdo al **mecanismo de llamada a funcion** cual es el orden correcto?

- a) 1, 2, 3 , 4      b) 2, 1 , 4, 3      c) 1, 3 , 4 ,2      d) 4, 3 , 2 , 1 ( B )

3.0

0.5 ptos. c/u

## PROBLEMAS

0.-Dibuje el **arbol sintactico** de la expresion **a := b + c \* d;**

Dada la gramatica:

LISTA -> LISTA + DIGITO | LISTA - DIGITO | DIGITO

DIGITO -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Suponga que la **Definición dirigida por la sintaxis es de la forma**

Produccion	Regla semantica
LISTA -> LISTA op DIGITO	LISTA.val := LISTA.val op DIGITO.val
LISTA -> DIGITO	LISTA.val := DIGITO.val
DIGITO -> 0	DIGITO.val := 0

1.-Dibuje el **arbol de analisis sint.** y el **grafo de dependencias** para las sig.cadena: 3+4-5+6

2a.-dibuja el arbol para la **expresiones de tipo** de la estructura de abajo :

struct agregado { int a, b, c; };

typedef struct agregado Agregado;

2b.-Cual es la **expresiones de tipo** para : int \*f(int \*,int \*, int);

a) int x pointer(int) x pointer(int) -> pointer (int)	b) int x pointer(int) x pointer (int)->int ✓ d )
c) int x int x int -> pointer (int)	d) pointer(int) x pointer(int) x int -> pointer (int)

3.-Cual es el **codigo de 3 direcciones** de:  $x - 25 * (y + z)$  ? ( b )

a)	tmp0 = x - 25 tmp1 = y + z tmp2 = tmp0 * tmp1	b)	tmp0 = y+z tmp1 = 25 tmp2 = tmp1 * tmp0 tmp3 = x - tmp2
c)	tmp0 = 25 * (y+z) tmp1 = x tmp2 = tmp2 - tmp1	d)	Ninguna de las anteriores

4.-Cual es el **codigo de 3 direcciones** de:  $((x / y) + k) - m * n$  ? ( a )

a)	t1 = x / y t2 = t1 + k t3 = m * n t4 = t2 - t3	b)	t1 = x / y + k t2 = t1 - m * n
c)	t1 = (x / y) + k t2 = t1 - (m * n)	d)	Ninguna de las anteriores

5.- a < b and c < d ( b )

6.- a < b or c < d ( a )

7.- a < b or c < d or e < f ( e )

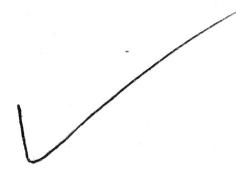
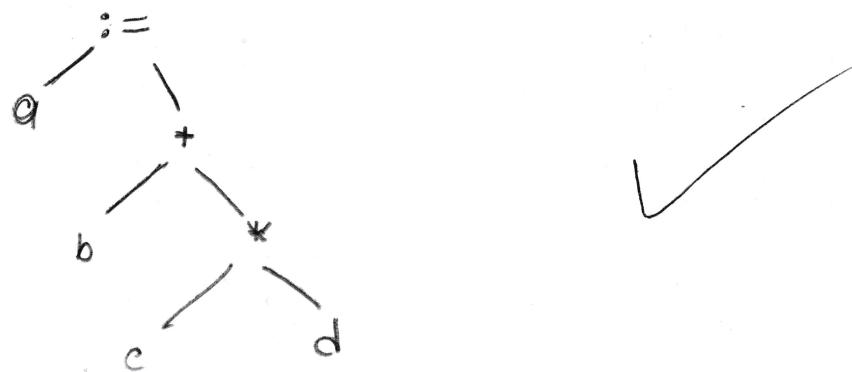
a)	if a < b goto Ltrue goto L1 L1: if c < d goto Ltrue goto Lfalse	b)	if a < b goto L1 goto Lfalse L1: if c < d goto Ltrue goto Lfalse
c)	if a < b goto L1 goto Lfalse L1: if c < d goto L2 goto Lfalse L2: if e < f goto Ltrue goto Lfalse	d)	if a < b goto Ltrue goto L1 L1: if c < d goto L2 goto Lfalse L2: if e < f goto Ltrue goto Lfalse
e)	if a < b goto Ltrue goto L1 L1: if c < d goto Ltrue goto L2 L2: if e < f goto Ltrue goto Lfalse	f)	if a < b goto Ltrue goto Lfalse L1: if c < d goto L2 goto Lfalse L2: if e < f goto Ltrue goto Lf

8.-Genere el codigo de 3 direcciones de : cta := 20 while ( cta <= 100) { cta := cta + 20 }

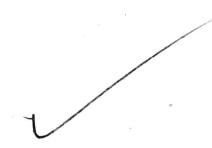
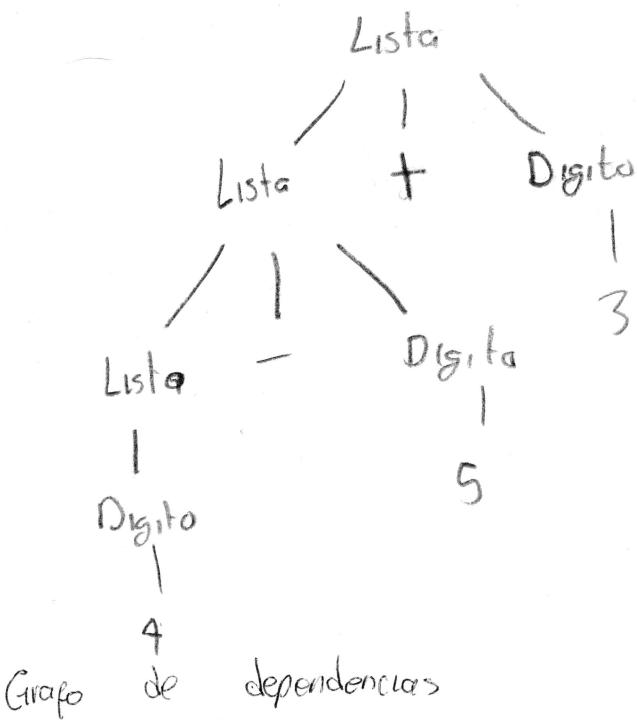
a)	La: Lb: Lc:	b)	La: Lb: Lc:
	t0 = 20 if t0 < 100 goto Lb goto Lc Lb: Lc: t0 = t0 + 20		t0 = 20 if t0 < 100 goto Lb goto Lc Lb: Lc: t0 = t0 + 20 goto La
c)	La: Lb: Lc:	d)	La: Lb: Lc:

Problemas

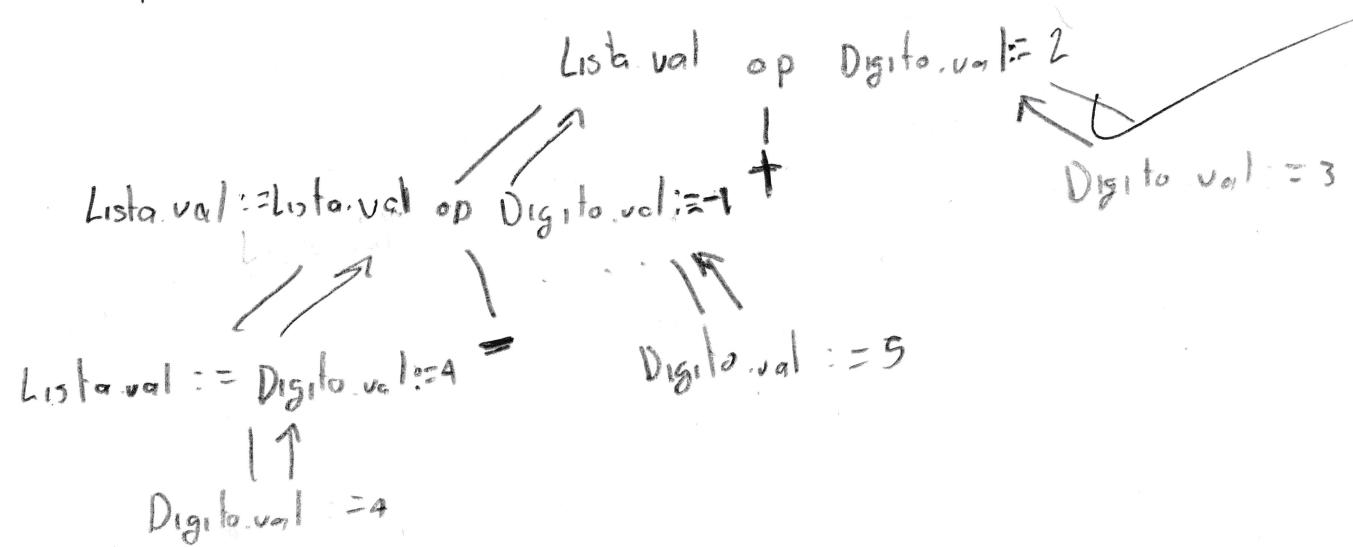
① Dibuja el árbol de análisis sintáctico de la expresión  $a := b + c * d$



② Árbol de análisis sintáctico



Grafo de dependencias



Digito.val = 3

1) Dibuja el árbol sintáctico y el grafo de dependencias para la siguiente cadena: 3+4-5+6

2) Dibuja el árbol para las expresiones de tipo de la estructura de abajo  
struct agregado { int a, b, c; }  
typedef struct agregado Agregado

