

Guía ETS Compiladores

(1)

- 1-Defina **compilador**
- 2-Cuales son las dos partes de la compilación
- 3-Describe las 6 **fases** de un **compilador**
- 4-Cuales son los 8 **modulos** de un **compilador**

- 1.-Para que sirve el **Análisis Léxico** ()
- a) Para generar el código en lenguaje objeto b) Nos dice si una cadena pertenece al lenguaje generado por una gramática
- c) Para dividir una cadena en tokens d) Los compiladores no lo necesitan nunca

- 2.-Es la **fase final** de un **compilador** ()
- a) análisis semántico b) manejo de errores
- c) optimización de código d) generación de código

- 3.-Es uno de los 8 **modulos** de un **compilador** ()
- a) análisis semántico b) manejo de errores
- c) optimización de código d) generación de código

- 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*

- 5.-Indica gráficamente cómo del **símbolo inicial** de una gramática deriva una cadena del lenguaje. ()
- a) árbol de análisis sintáctico con anotaciones b) árbol sintáctico
- c) árbol de análisis sintáctico d) Ninguno de los anteriores

- 6.-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

Falso o Verdadero (F/V)

- 1.-**Componente léxico** es sinónimo de no terminal ()
- 2.-**Token** es sinónimo de no terminal ()
- 3.-Una **gramática no** es **ambigua** si existe alguna cadena de terminales que pueda obtenerse mediante árboles de análisis sintáctico distintos (dos árboles distintos dan la misma cadena) ()
- 4.-Dos **gramáticas** son **equivalentes** si generan el mismo lenguaje. ()
- 4.1.-Dos **bloques básicos** son **equivalentes** si calculan el mismo conjunto de expresiones ()

- 4.2-La **eliminación** de **subexpresiones comunes** es una **transformación** que **preserva** la estructura en **bloques básicos**. ()
- 5.-En el **Análisis sintáctico ascendente** el árbol de análisis sintáctico la construcción se inicia en la raíz y avanza hacia las hojas ()
- 6.-En el **Análisis sintáctico descendente** se construye el árbol de análisis sintáctico de la cadena de desde las hojas y avanza hacia las raíz ()
- 7.-el **arbol sintactico** no es una version condensada del **arbol de analisis sintactico** ()
- 8.-Un **Esquema de traducción** asocia a cada símbolo de una **GLC** un conjunto de atributos y a cada producción, un conjunto de reglas semánticas para calcular los valores de los atributos asociados con los símbolos que aparecen en esa producción. ()
- 9.-**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. ()
- 10.-Los valores de los **atributos sintetizados** se calculan a partir de los valores de atributos de su nodo padre o sus nodos hermanos. ()
- 11.-Un **atributo** es **heredado** si su valor depende de los valores de los atributos de su padre y/o de sus hermanos. ()
- 12.-El **código de tres direcciones** consiste en una secuencia de instrucciones, cada una de las cuales tiene como máximo tres operandos. ()
- En lenguaje C los **parámetros formales** no tienen nombre ()
- 13.-En lenguaje C los **parámetros formales** son como **variables locales** que ya fueron inicializadas en el momento de la llamada a la función o procedimiento ()
- 14.-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 ()
- 15.-En hoc los **parámetros usados dentro de la definicion de las funciones** no tienen nombre ()
- 16.-No es posible definir **funciones recursivas** en hoc ()
- 17.-En hoc no hay **variables locales** ()
- 18.-En hoc cuando una función termina su ejecución se **saca** su **marco** de la **pila de llamadas**. ()
- 19.-En hoc los **parámetros reales** son listas de **expresiones** ()
- 20.-En hoc el código que ejecuta la **maquina virtual de pila** esta en prefijo (considere como se ejecuta una operacion de suma) ()
- 21.-En hoc el **tipo** de las **variables** es entero ()
- 22.-En hoc el **tipo** de las **variables** es doble ()
- 23.-En hoc los **parámetros reales** se meten a la pila ()
- 24.-Es imposible que la **pila de llamadas** de hoc se **desborde** (Stack Overflow) ()

Expresiones Regulares

- 1.-Describa el lenguaje correspondiente a la expresión regular $(0|1)^*0$
Escriba expresiones regulares para cada uno de lo siguientes.
- 2.-Las cadenas sobre el alfabeto $\{a, b, c\}$, donde la primera a precede a la primera b.
- 3.-Las cadenas sobre el alfabeto $\{a, b, c\}$ con un número par de a's.
- 4.-Los números binarios que son múltiplos de cuatro.
- 5.-Los números binarios que son mayores que **101.001**.
- 6.-Las cadenas sobre el alfabeto $\{a, b, c\}$ que no contienen la sub-cadena contigua baa.

Para cada una de las siguientes expresiones regulares , use la construcción de Thompson para derivar un autómata finito no determinista (NFA) que reconoce el mismo lenguaje.

- 1.- aaa
- 2.- $b^*a \mid bb$
- 3.- $(ab)^*ab$
- 4.- a^*bc^*d
- 5.- $(a|bc^*)a^*$
- 6.- $(a|b)^*$
- 7.- $a^*|b^*$
- 8.- $(a^*|b^*)^*$
- 9.- $((aa)^*(ab)^*(ba)^*(bb)^*)^*$
- 10.- $((b|a^*c)x)^*|x^*a$

Para cada una de la **NFA** de en el ejercicio anterior , utilizar la construcción de conjuntos para derivar un autómata finito determinista equivalente (**DFA**)

Para cada uno de los **DFA** en el ejercicio anterior, utilice el método de partición para derivar un mínimo **DFA** equivalente.

Arboles de análisis sintáctico y derivaciones

1.- Considere la siguiente gramática

$S \rightarrow 0S1 \mid 01$

- a) Mostrar una derivación de **00001111**
- b) Dibuje el árbol de análisis sintáctico para la entrada **00001111**

2.- Considere la siguiente gramática

$S \rightarrow bA$	$A \rightarrow bB$	$B \rightarrow bC$	$C \rightarrow \epsilon$
--------------------	--------------------	--------------------	--------------------------

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

3.- Considere la siguiente gramática

$S \rightarrow A$
 $A \rightarrow A+A \mid B++$
 $B \rightarrow y$

- a) Mostrar una derivación de **y + + + y + +**
- b) Dibuje el árbol de análisis sintáctico para la entrada **y + + + y + +**

4.- Considere la siguiente gramática

$l \rightarrow l, d \mid d$
 $d \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

- a) Mostrar una derivación de **9,8,7,6,5,4,3,2,1,0**
- b) Dibuje el árbol de análisis sintáctico para la entrada **9,8,7,6,5,4,3,2,1,0**

4

5.-Dada la gramática

$T = \{a, b, +, -, *, /, (,)\}$, $N = \{E, T, F\}$ $S = \{E\}$

$P = \{ E \rightarrow T \mid E+T \mid E-T$

$T \rightarrow F \mid T*F \mid T/F$

$F \rightarrow a \mid b \mid (E) \}$

y la cadena **(a+b)/b**

a) Obtenga una derivación de dicha cadena

b) Dibuje el árbol de análisis sintáctico que corresponde a la cadena mencionada

Análisis sintáctico predictivo descendente recursivo

Considere la siguiente gramática

$S \rightarrow a \mid (S)$

Escriba el analizador sintáctico predictivo descendente recursivo

Ambigüedad

1.-Demostrar que la siguiente gramática es ambigua

$A \rightarrow A x B \mid x$

$B \rightarrow x B \mid x$

usando la cadena **xxxxx**

2.-Demostrar que la siguiente gramática es ambigua

$S \rightarrow a S b S \mid b S a S \mid \epsilon$

usando la cadena **abab**

3.-Verificar si las siguientes gramáticas son ambiguas

$S \rightarrow S + S \mid S - S \mid a$

$S \rightarrow S S + \mid S S - \mid a$

Recursividad por la izquierda

Para eliminar la recursividad por la izquierda

$A \rightarrow Aa \mid b$

se transforma en

$A \rightarrow b \mid bR$

$R \rightarrow aR \mid \epsilon$

Ahora considere la siguiente gramática

$S \rightarrow (L) \mid a$

$L \rightarrow L , S \mid S$

Elimine la recursividad por la izquierda de dicha gramática.

Escriba el analizador sintáctico predictivo descendente recursivo

Escriba la sección de reglas de la especificación de **YACC** para dicha gramática

Definiciones dirigidas por la sintaxis

PRODUCCIÓN	REGLA SEMÁNTICA
$sec \rightarrow \text{comienza}$	$sec.x = 0$ $sec.y = 0$
$sec \rightarrow sec_1 instr$	$sec.x = sec_1.x + instr.dx$ $sec.y = sec_1.y + instr.dy$
$instr \rightarrow \text{este}$	$instr.dx = 1$ $instr.dy = 0$
$instr \rightarrow \text{norte}$	$instr.dx = 0$ $instr.dy = 1$
$instr \rightarrow \text{oeste}$	$instr.dx = -1$ $instr.dy = 0$
$instr \rightarrow \text{sur}$	$instr.dx = 0$ $instr.dy = -1$

Dibuje el **árbol de análisis sintáctico con anotaciones** para la siguiente cadena

c n e ss oo nnn eee ssss oooo

- 1.-Escribir la sección de reglas de la especificación de **YACC** para calcular la posición final del robot.
- 2.-Escriba una definición dirigida por la sintaxis para evaluar expresiones booleanas.

Esquemas de traducción

1.-Escriba un esquema de traducción para convertir una expresión en:

infijo a postfijo	postfijo a infijo
infijo a prefijo	prefijo a infijo

- 2.-Escriba un esquema de traducción para evaluar expresiones booleanas
- 3.-Para cada esquema de traducción de arriba escriba la sección de reglas de la especificación de **YACC**

Escritura de Gramaticas

1.-Escribir una gramática que genere todas las cadenas de longitud 4 formadas con los símbolos del alfabeto {a,b,c}

2.-Escribir una gramática que sirva para generar las siguientes cadenas

Especie perro	Especie gato	Especie perro	Especie gato
Edad 1	Edad 2	Edad 2	Edad 2
Sexo macho	Sexo macho	Sexo hembra	Sexo macho
Tamaño grande	Tamaño mediano	Tamaño pequeño	Tamaño grande
Colores negro , blanco	Colores negro , blanco ,	Colores canela , gris	Colores blanco
Soy rápido , activo,	café	Soy fuerte , alegre,	Soy listo , obediente
alegre	Soy tranquilo , sociable	activo.	Aficiones jugar,
Aficiones correr, comer	Aficiones dormir,	Aficiones aullar	haraganear
	parrandear, comer		

3.-Escribir una gramática que sirva para generar las siguientes cadenas

Etiquetado Nerd	Etiquetado Geek	Etiquetado Nerd	Etiquetado Freak
Nivel Junior	Nivel Senior	Nivel Junior	Nivel Senior
Sexo Hombre	Sexo Mujer	Sexo Mujer	Sexo Hombre
Lenguajes Java , C ,	Lenguajes Pascal ,	Lenguajes PHP , Perl,	Lenguajes
Logo	Prolog , SQL	Java	Ensamblador, C
Aficiones programar,	Aficiones chatear,	Aficiones hackear,	Aficiones gotcha,
videogames, comics,	videogames,	googlear, gotcha,	dormir, chatear,
hackear, googlear	programar	dormir	comics

YACC

1.-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

2.-Los %% se usan para indicar

- a) inicio de la sección de declaraciones
- b) inicio de la sección de reglas
- c) precedencia de los operadores
- d) fin del código de soporte

3.-%token sirve para indicar

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

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

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

5.-Una **acción gramatical** debe ir entre

- a) comillas
- b) paréntesis
- c) corchetes
- d) llaves

5.-Considere la producción (en la seccion de reglas de una especificacion de YACC)

$S : S 'a' S 'b'$

\$4 a cual de los miembros del lado derecho de la producción se refiere?

- a) la 'a'
- b) la 1er S
- c) la segunda S
- d) la 'b'

6.-Si el codigo de yylex es el siguiente

```
int yylex() { return getchar(); }
```

de cuantos caracteres son los tokens?

- a) 0
- b) 1
- c) 2
- d) la cantidad de caracteres del token varia

Problemas

1.-Escriba la sección de reglas de la especificación de yacc para la gramática de abajo

$L \rightarrow L, D \mid D$

$D \rightarrow 0 \mid 1$

2.-Escriba la sección de reglas de la especificación de yacc para la gramática de abajo

$S \rightarrow U \mid V$

$U \rightarrow TaU \mid TaT$
 $V \rightarrow TbV \mid TbT$
 $T \rightarrow aTbT \mid bTaT \mid \epsilon$

3.-Escriba las acciones gramaticales para que imprima el numero de b's en la cadena de entrada

```

%{
/* escriba el tipo de los elementos en la pila de yacc */
#define YYSTYPE
%}

```

```

%%
S : '(' B ')' { }
;
B : '(' B ')' { }
  | D { $$=$1; }
;
D : { }
  | 'b' D { }
;
%%

```

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

$lista \rightarrow lista , figura \mid figura$
 $figura \rightarrow triangulo \mid cuadrilatero$
 $triangulo \rightarrow \textbf{lado lado lado}$
 $cuadrilatero \rightarrow \textbf{lado lado lado lado}$

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

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 1.-Considere la gramática para generar paréntesis anidados

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

Cadenas propuestas:

(a)
 $((a))$
 $(((a)))$
 $((((a))))$

Problema 2.-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)

Problema 3.- Considere la siguiente gramática :

1) $S \rightarrow AaAb$	2) $S \rightarrow BbBa$	3) $A \rightarrow \epsilon$	4) $B \rightarrow \epsilon$
-------------------------	-------------------------	-----------------------------	-----------------------------

Cadenas propuestas:
ab y ba

Problema 4.- Considere la siguiente gramática :

$S \rightarrow A$	$A \rightarrow \epsilon$	$A \rightarrow bbA$
-------------------	--------------------------	---------------------

Cadena propuesta: **bbbb**

Problema 1.- Considere la siguiente gramática :

1) $S \rightarrow X$	2) $X \rightarrow aXc$	3) $X \rightarrow XX$	4) $X \rightarrow b$
----------------------	------------------------	-----------------------	----------------------

Calcule

cerradura($\{X \rightarrow X \cdot X\}$), cerradura($\{X \rightarrow XX \cdot\}$) e $ir_a(\{X \rightarrow a \cdot Xc\}, X)$

-Para el **Análisis LR** las gramáticas se muestran con sus producciones numeradas
Para cada gramática:

- Calcule los conjuntos PRIMERO y SIGUIENTE
- Construya la tabla Análisis Sintáctico Predictivo no Recursivo (LL(1))
- Construya la colección de conjuntos de elementos LR (0)
- Construya la tabla SLR

Problema 2

1) $A \rightarrow xA$	2) $A \rightarrow yA$	3) $A \rightarrow y$
-----------------------	-----------------------	----------------------

-Explicar porque esta gramatica no es LL(1).

$I1=ir_a(I0, x)$, $I2=ir_a(I0, y)$, $I3=ir_a(I0, A)$, $I4=ir_a(I1, A)$, $I5=ir_a(I2, A)$

Problema 3

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

$I1=ir_a(I0, a)$, $I2=ir_a(I0, ($) , $I3=ir_a(I0, S)$, $I4=ir_a(I2, S)$, $I5=ir_a(I4, ,)$, $I6=ir_a(I4,)$,
 $I7=ir_a(I4, R)$, $I8=ir_a(I5, S)$, $I9=ir_a(I8, R)$

Use ambos análisis para analizar las siguientes cadenas:

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

Problema 4

1) $S \rightarrow A$	2) $A \rightarrow \epsilon$	3) $A \rightarrow bbA$
----------------------	-----------------------------	------------------------

$I1=ir_a(I0, b)$, $I2=ir_a(I0, S)$, $I3=ir_a(I0, A)$, $I4=ir_a(I1, b)$, $I5=ir_a(I4, A)$

Use ambos análisis para analizar la siguiente cadena : **bbbb**

Análisis LR Para cada gramática:

- Calcule los conjuntos PRIMERO y SIGUIENTE
- Construya la colección de conjuntos de elementos LR (0)
- Construya la tabla SLR
- Use la tabla de análisis SLR para analizar la(s) cadena(s) propuesta(s)
- Construya la colección de conjuntos de elementos LR (1)
- Construya la tabla de Análisis Sintáctico LR canonico
- Use la tabla de análisis LR canonico para analizar la(s) cadena(s) propuesta(s)

Problema 5

- 1) $A \rightarrow A c$ 2) $A \rightarrow d$

$I1=ir_a(I0, b)$, $I2=ir_a(I0, A)$, $I3=ir_a(I2, a)$

cadenas	d	dc	dcc	dccc
---------	---	----	-----	------

Problema 5.1

- 1) $S \rightarrow S a$ 2) $S \rightarrow b$

$I1=ir_a(I0, b)$, $I2=ir_a(I0, S)$, $I3=ir_a(I2, a)$

cadenas	b	ba	baa	baaa
---------	---	----	-----	------

Problema 6

- 1) $S \rightarrow AA$ 2) $A \rightarrow a A$ 3) $A \rightarrow b$

$I1=ir_a(I0, a)$, $I2=ir_a(I0, b)$, $I3=ir_a(I0, S)$, $I4=ir_a(I0, A)$, $I5=ir_a(I1, A)$,
 $I6=ir_a(I4, A)$

cadenas: **abab** y **baab**

Problema 7

- 1) $S \rightarrow BB$ 2) $B \rightarrow b B$ 3) $B \rightarrow c$

$I1=ir_a(I0, b)$, $I2=ir_a(I0, c)$, $I3=ir_a(I0, S)$, $I4=ir_a(I0, B)$, $I5=ir_a(I1, B)$,
 $I6=ir_a(I4, B)$

cadenas: **bcbc** y **cbbc**

Problema 8

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

$I1=ir_a(I0, ()$, $I2=ir_a(I0, a)$, $I3=ir_a(I0, A)$, $I4=ir_a(I1, A)$, $I5=ir_a(I4,)$

cadenas	(a)	((a))	(((a)))	((((a))))
---------	-------	---------	------------	---------------

Problema 9

- | | |
|------------------------|----------------------|
| 1) $S \rightarrow (S)$ | 2) $S \rightarrow e$ |
|------------------------|----------------------|

$I1=ir_a(I0, (), I2=ir_a(I0, e), I3=ir_a(I0, S), I4=ir_a(I1, S), I5=ir_a(I4,))$

cadenas	(e)	((e))	(((e)))	((((e))))
---------	-------	---------	------------	--------------

Problema 10

- | | | |
|----------------------|---------------------------|----------------------|
| 1) $E \rightarrow n$ | 2) $E \rightarrow (E, E)$ | Donde n es un entero |
|----------------------|---------------------------|----------------------|

$I1=ir_a(I0, (), I2=ir_a(I0, n), I3=ir_a(I0, E), I4=ir_a(I1, E), I5=ir_a(I4, ,),$
 $I6=ir_a(I5, E), I7=ir_a(I6,))$
cadenas ((21 , 18) , 17) y (21 , (18 , 17))

Problema 11

- | | | | |
|------------------------|----------------------|-------------------------|----------------------|
| 1) $S \rightarrow [L]$ | 2) $S \rightarrow a$ | 3) $L \rightarrow L, S$ | 4) $L \rightarrow S$ |
|------------------------|----------------------|-------------------------|----------------------|

$I1=ir_a(I0, [), I2=ir_a(I0, a), I3=ir_a(I0, S), I4=ir_a(I1, S), I5=ir_a(I1, L), I6=ir_a(I5, [),$
 $I7=ir_a(I5, ,), I8=ir_a(I7, S)$

Problema 12

- | | | | |
|------------------------|------------------------|-----------------------|----------------------|
| 1) $S \rightarrow dca$ | 2) $S \rightarrow dAb$ | 3) $S \rightarrow Aa$ | 4) $A \rightarrow c$ |
|------------------------|------------------------|-----------------------|----------------------|

$I1=ir_a(I0, d), I2=ir_a(I0, c), I3=ir_a(I0, S), I4=ir_a(I0, A), I5=ir_a(I1, c),$
 $I6=ir_a(I1, A), I7=ir_a(I4, a), I8=ir_a(I5, a), I9=ir_a(I6, b)$

Problema 13

- | | | | | |
|-----------------------|------------------------|-----------------------|------------------------|----------------------|
| 1) $S \rightarrow Aa$ | 2) $S \rightarrow BAa$ | 3) $S \rightarrow dc$ | 4) $S \rightarrow bda$ | 5) $A \rightarrow d$ |
|-----------------------|------------------------|-----------------------|------------------------|----------------------|

$I1=ir_a(I0, b), I2=ir_a(I0, d), I3=ir_a(I0, S), I4=ir_a(I0, A), I5=ir_a(I1, d), I6=ir_a(I1,$
 $A),$
 $I7=ir_a(I2, c), I8=ir_a(I4, a), I9=ir_a(I5, a), I10=ir_a(I6, c)$

Problema 14

- | | | | | | |
|-------------------------|----------------------|-----------------------|----------------------|----------------------|----------------------|
| 1) $S \rightarrow cSAa$ | 2) $S \rightarrow d$ | 3) $A \rightarrow aB$ | 4) $A \rightarrow a$ | 5) $B \rightarrow a$ | 6) $B \rightarrow b$ |
|-------------------------|----------------------|-----------------------|----------------------|----------------------|----------------------|

$I1=ir_a(I0, c), I2=ir_a(I0, d), I3=ir_a(I0, S), I4=ir_a(I1, S), I5=ir_a(I4, a),$
 $I6=ir_a(I4, A), I7=ir_a(I5, a), I8=ir_a(I5, b), I9=ir_a(I5, B), I10=ir_a(I6, d)$

cadena: **cdad**

Problema 15

1) $R \rightarrow R \mid R$	2) $R \rightarrow RR$	3) $R \rightarrow R^*$	4) $R \rightarrow (R)$	5) $R \rightarrow a$	6) $R \rightarrow b$
-----------------------------	-----------------------	------------------------	------------------------	----------------------	----------------------

$I1=ir_a(I0, (), I2=ir_a(I0, a), I3=ir_a(I0, b), I4=ir_a(I0, R), I5=ir_a(I1, R), I6=ir_a(I5, |),$
 $I7=ir_a(I5, *), I8=ir_a(I5, R), I9=ir_a(I5,), I10=ir_a(I6, R)$

cadena: **aa*ba | b**

Problema 16

(1) $S \rightarrow L = R$	(2) $S \rightarrow R$	(3) $L \rightarrow * R$	(4) $L \rightarrow id$	(5) $R \rightarrow L$
---------------------------	-----------------------	-------------------------	------------------------	-----------------------

$I1=ir_a(I0, id), I2=ir_a(I0, *), I3=ir_a(I0, S), I4=ir_a(I0, L), I5=ir_a(I0, R),$
 $I6=ir_a(I0, L), I7=ir_a(I2, R), I8=ir_a(I4, =), I9=ir_a(I8, R)$
 cadena **id = * id**

Problema 17

1) $S \rightarrow A$	2) $A \rightarrow \epsilon$	3) $A \rightarrow Abb$
----------------------	-----------------------------	------------------------

$I1=ir_a(I0, S), I2=ir_a(I0, A), I3=ir_a(I2, b), I4=ir_a(I3, b)$

Problema 18.- Considere la siguiente gramática :

1) $S \rightarrow AaAb$	2) $S \rightarrow BbBa$	3) $A \rightarrow \epsilon$	4) $B \rightarrow \epsilon$
-------------------------	-------------------------	-----------------------------	-----------------------------

$I1=ir_a(I0, S), I2=ir_a(I0, A), I3=ir_a(I0, B), I4=ir_a(I2, a), I5=ir_a(I3, b),$
 $I6=ir_a(I4, A), I7=ir_a(I5, B), I8=ir_a(I6, b), I9=ir_a(I7, a)$

cadena: **ab y ba**

Problema 19.- Considere la siguiente gramática:

$S \rightarrow a S b S$	$S \rightarrow a$
-------------------------	-------------------

$I1=ir_a(I0, a), I2=ir_a(I0, S), I3=ir_a(I1, S), I4=ir_a(I3, b), I5=ir_a(I3, S),$

cadena:

Problema 20.- Considere la siguiente gramática:

1) $C \rightarrow AB$	2) $A \rightarrow a$	3) $B \rightarrow a$
-----------------------	----------------------	----------------------

$I1=ir_a(I0, a), I2=ir_a(I0, C), I3=ir_a(I0, A), I4=ir_a(I3, a), I5=ir_a(I3, B).$

cadena: **aa**

HOC

1.-En hoc 1 hay una **pila** cual es?

2.-A partir de hoc4 se usan dos etapas en hoc. Cuales son y que hacen ?

- I.-
- II.-

3.-En hoc 6 hay 3 **pilas** cuales son?

- I.-
- II.-
- III.-

1.-El **código de tres direcciones** se usa en

- a) El análisis sintáctico
- b) El análisis léxico
- c) Generación de código intermedio
- d) Generación de código

2.-Un _____ es $[A \rightarrow \alpha \cdot \beta, a]$ donde $A \rightarrow \alpha\beta$ es una producción y a es un terminal o $\$$.

- a) mango
- b) prefijo viable
- c) elemento LR (1)
- d) elemento LR (0)

3.-Es una producción de G con un punto en cierta posición del lado derecho.

- a) mango
- b) prefijo viable
- c) elemento LR (1)
- d) elemento LR (0)

4.-Son prefijos de las formas de frase derecha que pueden aparecer en la pila

- a) mango
- b) elemento LR (1)
- c) prefijo viable
- d) elemento LR (1)

5.-Un _____ de una forma de frase derecha γ es una producción $S \rightarrow \beta$ y una posición de γ donde la cadena β podría encontrarse y sustituirse por A para producir la forma de frase derecha previa en una derivación por la derecha de γ .

- a) prefijo viable
- b) mango
- c) elemento LR (0)
- d) elemento LR (1)

Traducción dirigida por la sintaxis

1.-Construir explícita o implícitamente el **grafo de dependencias**.

2.-Construir el **arbol de analisis sintactico** para la gramatica 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.

Para la realizacion de una **Traducción 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

Ambientes para el momento de la ejecucion

1.-Escriba 3 datos importantes que se almacenan usualmente en un **marco** (o registro de activacion) **de funcion**.

- I. _____
- II. _____
- III. _____

- 1-Ejecutar la funcion (poner el contador de programa igual a la direccion de su primera instruccion y ejecutar la instruccion a la que apunta el contador de programa) y meter el valor de retorno de la funcion 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 direccion de retorno y ejecutar la instruccion 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) 1, 3, 4, 2 c) 2, 1, 4, 3 d) 4, 3, 2, 1 ()

PROBLEMAS

Comprobacion de de tipos

Esciba las **expresiones de tipo** para :

- a) `char *function(double, double);` b) `struct agregado { char x; double y; };`

Generacion deCodigo Intermedio

Dada la gramatica

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

1.-Dibuje el **arbol de analisis sintactico**, el **arbol sintactico** y el **grafo de dependencias** para las siguientes cadenas: **1+2+3+4** y **2+4-6+8-10**

2.-Dibuje el **arbol sintactico** para las siguientes cadenas: **(2+3) * (10 - 5)** y **a / (b + c / (d + e))**

3.-Escriba el codigo de 3 direcciones de: **a * - (b+c)** y **a:= (b + c) * (e + f)**

4.-Escriba el codigo de 3 direcciones de: **(a * b + h) - j * k + 1**

5.-Hacer el codigo 3 direcciones de la expresion : **a > b + h or b == d**

6.- Traduzca a flujo de control (obtenga el codigo de tres direcciones) las siguientes expresiones booleanas

I) **a < b or c < d**

II) **a < b or c < d or e < f**

III) **a < b or c < d or e < f or g < h**

IV) **a < b and c < d**

V) **a < b and c < d and e < f**

VI) **a < b and c < d and e < f and g < h**

VIII) **a < b or c < d and e < f**

IX) **a < b and c < d or e < f**

7.-Hacer el código 3 direcciones de :

```
a := 0
while ( a <= 5) {
    a := a + 1
}
```

```
for ( i = 0; i < 5; i=i+1){
    A;
}
```

8.-Hacer el código 3 direcciones de :

```
a := 2 * x + 10;
while ( a <= p + 2 ) {
    a := p[4+i*2];
}
a := a + 1;
```

```
for ( i = 0; i < 5; i=i+1){
    a=p[2*i];
}
```

Generación de Código Objeto

9.-Si una instrucción de asignación de la forma $x=y+z$. Se traduce a:

Como se traducen

```
mov y, R0
add z, R0
mov R0, x
```

```
a = b + c;
d = a + c;
a = a + 1;
```

10.-Si una instrucción de asignación de la forma $a=a+1$. Se traduce a:

Como se traducen

```
Mov a, R0
add #1, R0
mov R0, a
```

```
a = c + 2
d = a + 3
```

11.-Para el siguiente código genere el código de 3 direcciones y divida el código generado en bloques básicos

```
w = 0;
x = x + y;
y=0;
if( x > z ){
    y = x;
    x++;
} else {
    y = z;
    z++;
}
w = x + z;
```