



Apellidos: Nombre:
DNI:

Instrucciones generales y valoración

- No se permitirá el uso de calculadora.
- Sobre la mesa únicamente el enunciado del examen, un bolígrafo y el carnet de estudiante.
- En los problemas no se evaluarán las respuesta que no estén debidamente justificadas. En otras palabras, además de la solución final, se tiene que mostrar el proceso seguido para obtenerla.
- No se podrá utilizar ninguna hoja aparte de las proporcionadas.
- En las preguntas de test de opción múltiple, la puntuación máxima sólo se conseguirá si se han seleccionado todas las opciones correctas y ninguna de las incorrectas, una opción mal elegida resta una opción correcta, pero no resta globalmente.
- En las preguntas de emparejar, algunas de las respuestas lo pueden ser de varias de las opciones, y también algunas de las respuestas puede que no lo sea de ninguna de las opciones.
- Para anular una respuesta rodearla con un círculo.
- Las preguntas de test de opción única restan *globalmente* un cuarto de pregunta correcta.

Hora de inicio: 16:00

Hora de fin: 18:45

Duración: 2 horas y tres cuartos

Valoración: La puntuación de cada pregunta es la mostrada en la siguiente tabla.

Pregunta:	1	2	3	4	5	6	7	Total:
Puntos:	30	6	10	16	16	6	16	100
Calificación:								

1. (30 puntos) Preguntas de test

- (a) Asocia los términos equivalentes (A: token, B: lexer, C: parser, D: front-end, E: back-end)

<input checked="" type="checkbox"/> A	Componente léxico.	<input checked="" type="checkbox"/> B	Analizador léxico.
<input checked="" type="checkbox"/> E	Etapa de síntesis.	<input checked="" type="checkbox"/> C	Analizador sintáctico.

- (b) El lenguaje de programación de alto nivel **FORTTRAN** fue el primero para el que se contó con un compilador.
- (c) Dados los lenguajes $L_1 = \{ab, a\}$ y $L_2 = \{c, bc\}$, ¿cuántas cadenas distintas hay en el lenguaje $L_1 \cdot L_2^2$? **6**
- (d) Dados los lenguajes $L_1 = \{ab, b, c\}$ y $L_2 = \{x, y\}$, ¿cuáles de las siguientes cadenas pertenecen al lenguaje $L_2^2 \cdot L_1$?

yyb yxab yxc xxabc xyaa

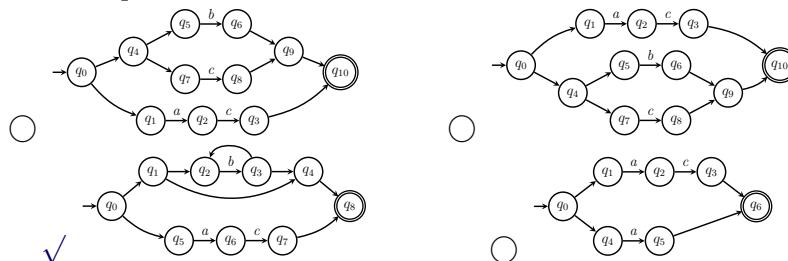
- (e) ¿Qué expresión regular no provocará ni warnings ni errores en flex?

[0-Z] [z-b] ECHO; +a{5,6}b* ECHO;
 (ab)*c ECHO; (a|b)+/(c*a) ECHO;

- (f) ¿Qué expresiones regulares no están bien construidas y devuelven error flex?

a{2,1} a{2, } a{5,} a{,} a{3,4}

- (g) Selecciona el autómata generado por la expresión regular $(b^*|(a \cdot c))$ siguiendo el método de Thompson.



- (h) ¿Cuál de los siguientes es el formato de la función de transición de un autómata finito no determinista?

$f: Q \times \Sigma \rightarrow \mathcal{P}(Q)$ $f: Q \times \Sigma \rightarrow Q$
 $f: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow Q$ $f: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$

- (i) Seleccionar las falsas

El estado inicial de un autómata no puede ser nunca final.
 La cardinalidad del alfabeto de entrada es siempre mayor que la cardinalidad del alfabeto de estados.
 La cardinalidad del conjunto de estados iniciales es siempre menor que la cardinalidad del conjunto de estados del autómata.

Un autómata finito como mínimo tiene un estado inicial.

- (j) Sea una gramática de producciones: $S \rightarrow BB$, $B \rightarrow CC$, $C \rightarrow 1|2|\epsilon$, ¿cuántas cadenas distintas genera? 11 12 15 16 31 63 64

- (k) Dada la gramática de producciones: $S \rightarrow A$, $A \rightarrow B|C$, $C \rightarrow BmC|D$, $D \rightarrow a|b$, ¿cuáles de las siguientes producciones harían a la gramática recursiva a izquierdas si fueran añadidas a las producciones de partida?

$A \rightarrow D$ $C \rightarrow 1C$ $B \rightarrow C$ $C \rightarrow CmB$ $D \rightarrow B$ $D \rightarrow A$

- (l) Dada la gramática $S \rightarrow bS$, $S \rightarrow a$, ¿qué puede decirse de ella?
- Es LL(1) Es SLR(1), pero no LL(1)
 Es SLR(1) Es LL(1) y SLR(1)
- (m) Dada la gramática $S \rightarrow aBc$; $S \rightarrow dA$; $S \rightarrow cCA$; $B \rightarrow b$; $B \rightarrow bC$; $C \rightarrow b$; $C \rightarrow c$; $A \rightarrow b$; $A \rightarrow bS$; ¿cuál sería el asidero de la forma sentencial $dbaBc$? **345** (utiliza un número en el que cada dígito indique el índice de los símbolo que forman parte del asidero, por ejemplo, si en la forma sentencia $abBac$ el asidero fuera la primera a , la respuesta sería 1, si fuera la subcadena bBa , la respuesta sería 234, si fuera la segunda a , la respuesta sería 4)
- (n) Asocia las siguientes descripciones con el concepto con que se corresponden (A: Atributo sintetizado, B: Gramática S-atribuida, C: Gramática L-atribuida , D: Atributo heredado)
- Atributo del antecedente que es función de los otros atributos del antecedente y de los atributos de los elementos del consecuente.
- Gramática con atributos que sólo usa atributos sintetizados.
- Gramática con atributos en la que los atributos heredados de un símbolo son sólo función de atributos de los símbolos a su izquierda.
- Atributo de un elemento del consecuente función de los otros atributos de ese elemento y de los atributos de los otros elementos del consecuente o del antecedente.
- (ñ) Dado el siguiente programa JavaCC, ¿cuál sería el resultado de procesar la entrada: 12 abc 34?

```

PARSER_BEGIN(test)
public class test {
    public static void main (String[] args) throws Exception {
        test parser = new test(System.in); parser.lst();
    }
}
PARSER_END(test)
TOKEN : { <N: (["0"-"9"])+> | <W: (["a"-"z"])+> }
SKIP: { <OTRO: ~[] > }
void lst():{Token t1, t2; String s;}{t1=<N> t2=<W> s=noterm(t2.image)
{System.out.println(t1.image+s+t2.image);} }
String noterm(String v):{Token t1;}{ t1=<N> { return v+t1.image; } }

```

- Daría un error sintáctico Imprimiría 12abc3434 No imprimiría nada
 Imprimiría 12abc34abc Imprimiría 12 abc 34 Imprimiría 1234abc
 Imprimiría 34abc1212abc Imprimiría abc123434abc
- (o) Asocia las siguientes definiciones con el término definido (A: Llamada o invocación a un procedimiento, B: Función, C: Definición de un procedimiento, D: Parámetros formales, E: Activación del procedimiento, F: Estado)
- C Una declaración que asocia un identificador con una porción de código.
- B Procedimiento que devuelve valores.
- F Función que transforma una posición de memoria en el valor allí almacenado (transforma el valor de lado izquierdo en un valor de lado derecho).

- (p) Asocia los siguientes conceptos con el tipo de noción de que se trata (A: Noción estática, B: Noción dinámica)
- B Duración de un enlace.
 - A Ámbito de una declaración.
 - A Definición de un procedimiento.
- (q) Asocia las siguientes afirmaciones con el tipo de análisis (A: análisis descendente, B: análisis ascendente)
- A Es como construir el árbol de análisis sintáctico empezando por la raíz.
 - B La idea es generar una forma sentencial a partir del axioma, reconstruyendo una derivación más a la derecha en orden inverso.
 - B Se conoce también como análisis LR(k).
 - A Los dos elementos de juicio son la porción de entrada por leer y el no terminal más a la izquierda de la forma sentencial.

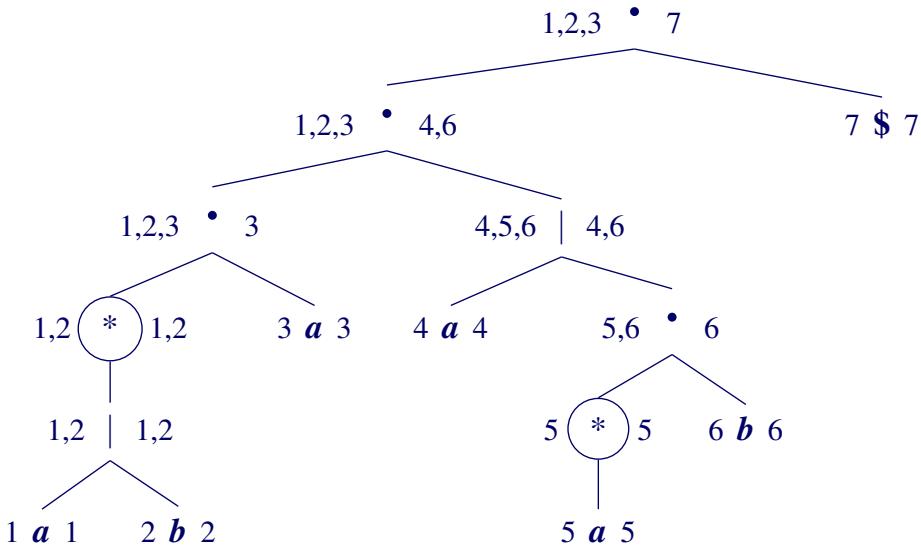
2. (6 puntos) Registros de activación Explicar en qué consiste un registro de activación, qué información se guarda en él, y cómo ésta está estructurada.

El registro de activación es un bloque de memoria donde se almacena la información necesaria para la ejecución de un procedimiento. Suele constar del siguiente conjunto de campos:

1. Los valores temporales, como los que surgen en la evaluación de expresiones, se almacenan en el campo para valores temporales.
2. El campo para datos locales, guarda los datos locales a una ejecución de un procedimiento.
3. El campo para el estado guardado de la máquina, contiene información sobre el estado de la máquina justo antes de que sea llamado el procedimiento. Esta información incluye los valores del contador del programa y los registros de la máquina que deben reponerse cuando el control regrese del procedimiento.
4. El *enlace de acceso* (opcional) que se utiliza para hacer referencia a los datos no locales guardados en otros registros de activación.
5. El *enlace de control* (opcional), también conocido como *vínculo de control* o *vínculo dinámico*, apunta al registro de activación del autor de la llamada.

Es habitual almacenar al principio del registro de activación el valor devuelto y los parámetros actuales, puesto que a esa información también tiene que tener acceso el procedimiento invocador. Y éste sabe el tipo del valor devuelto, el número y tipos de los parámetros, y por tanto le es posible calcular las posiciones de memoria donde se encuentran cada uno de ellos utilizando desplazamientos relativos respecto de su propio registro de activación. Los datos locales y datos temporales están al final del registro y son responsabilidad exclusiva del procedimiento llamado.

3. (10 puntos) **Transformación de expresión regular a AFD** Aplica el algoritmo de Aho-Sethi-Ullman y de minimización, para obtener el AFD mínimo capaz de reconocer el lenguaje definido por la siguiente expresión regular $(a|b)^*a(a|a^*b)$.



POS | STEPOS

-----+

1		1	2	3
2		1	2	3
3		4	5	6
4		7		
5		5	6	
6		7		
7		vacío		

 $P_0 = \{ \{A\}, \{B\}, \{C\}, \{D\} \}$

00	11	11	00
0		1	

-----+-----+-----+

A		B		A		1	2	3				
B		C		D		1	2	3	4	5	6	
(C)		C		D		1	2	3	4	5	6	7
(D)		B		A		1	2	3	7			

 $P_1 = \{ \{A\}, \{B\}, \{C\}, \{D\} \}$

4. (16 puntos) Análisis ascendente Completar los conjuntos de items LR(1), las tablas LR(1) y la traza de análisis de la cadena «**misie**» (sin las comillas), para la gramática que sigue (*S* es el axioma, se usa el símbolo ε para representar la cadena vacía):

$$\begin{array}{l} S \rightarrow mCe \mid s \\ C \rightarrow iMC \mid iM \\ M \rightarrow SM \mid \varepsilon \end{array}$$

- En la operación de lectura, la lectura de símbolos se tiene que hacer en el orden *S, C, M, m, e, s, i*.
- En la operación de cierre, las producciones se tienen que añadir en el mismo orden en el que aparecen en la gramática.

Conjuntos de items marcados

Conjunto 0

$@S \rightarrow \bullet S\$; \$$	S, 1
$S \rightarrow \bullet mCe ; \$$	m, 2
$S \rightarrow \bullet s ; \$$	s, 3

Conjunto 2

$S \rightarrow m \bullet Ce ; \$$	C, 4
$C \rightarrow \bullet iMC ; e$	i, 5
$C \rightarrow \bullet iM ; e$	

Conjunto 3

$S \rightarrow s \bullet ; \$$	e, 6
Conjunto 4	
$S \rightarrow mC \bullet e ; \$$	

Conjunto 1

$@S \rightarrow S \bullet \$; \$$

Conjunto 5

$C \rightarrow i \bullet MC ; e$	S, 7
$C \rightarrow i \bullet M ; e$	M, 8
$M \rightarrow \bullet SM ; i, e$	m, 9
$M \rightarrow \bullet ; i, e$	s, 10
$S \rightarrow \bullet mCe ; m, s, i, e$	
$S \rightarrow \bullet s ; m, s, i, e$	

Conjunto 6

$S \rightarrow mCe \bullet ; \$$	S, 7
Conjunto 7	
$M \rightarrow S \bullet M ; i, e$	M, 11
$M \rightarrow \bullet SM ; i, e$	m, 9
$M \rightarrow \bullet ; i, e$	s, 10
$S \rightarrow \bullet mCe ; m, s, i, e$	
$S \rightarrow \bullet s ; m, s, i, e$	

Conjunto 8

$C \rightarrow iM \bullet C ; e$	C, 12
$C \rightarrow iM \bullet ; e$	i, 5
$C \rightarrow \bullet iMC ; e$	
$C \rightarrow \bullet iM ; e$	

Conjunto 9

$S \rightarrow m \bullet Ce ; m, s, i, e$	C, 13
$C \rightarrow \bullet iMC ; e$	i, 5
$C \rightarrow \bullet iM ; e$	

Conjunto 10

$S \rightarrow s \bullet ; m, s, i, e$	e, 14
Conjunto 11	
$M \rightarrow SM \bullet ; i, e$	

Conjunto 12

$C \rightarrow iMC \bullet ; e$

Conjunto 13

$S \rightarrow mC \bullet e ; m, s, i, e$

Conjunto 14

$S \rightarrow mCe \bullet ; m, s, i, e$
--

Tabla Acción e ir a

Para especificar las acciones se utiliza una codificación similar a la vista en clase:

- dj: significa desplazar e ir al estado j.
- rj: significa reducir utilizando la producción j-ésima.
- A!: representa la acción de aceptar la cadena.
- -: se utiliza para indicar que la casilla está vacía (no tiene acción).
- En la parte de acción se especifica únicamente el estado al que pasa el autómata.

	<i>m</i>	<i>e</i>	<i>s</i>	<i>i</i>	\$	<i>S</i>	<i>C</i>	<i>M</i>
0	d2		d3			1		
1					A!			
2				d5			4	
3					r2			
4		d6						
5	d9	r6	d10	r6		7	8	
6					r1			
7	d9	r6	d10	r6		7	11	
8		r4		d5			12	
9				d5			13	
10	r2	r2	r2	r2				
11		r5		r5				
12		r3						
13		d14						
14	r1	r1	r1	r1				

Traza de análisis para la cadena «misie»

PILA	ENTRADA	SALIDA
0	<i>misie\$</i>	—
0 <i>m</i> 2	<i>isie\$</i>	—
0 <i>m</i> 2 <i>i</i> 5	<i>sie\$</i>	—
0 <i>m</i> 2 <i>i</i> 5 <i>s</i> 10	<i>ie\$</i>	<i>S</i> → <i>s</i>
0 <i>m</i> 2 <i>i</i> 5 <i>S</i> 7	<i>ie\$</i>	<i>M</i> → ε
0 <i>m</i> 2 <i>i</i> 5 <i>S</i> 7 <i>M</i> 11	<i>ie\$</i>	<i>M</i> → <i>SM</i>
0 <i>m</i> 2 <i>i</i> 5 <i>M</i> 8	<i>ie\$</i>	—
0 <i>m</i> 2 <i>i</i> 5 <i>M</i> 8 <i>i</i> 5	<i>e\$</i>	<i>M</i> → ε
0 <i>m</i> 2 <i>i</i> 5 <i>M</i> 8 <i>i</i> 5 <i>M</i> 8	<i>e\$</i>	<i>C</i> → <i>iM</i>
0 <i>m</i> 2 <i>i</i> 5 <i>M</i> 8 <i>C</i> 12	<i>e\$</i>	<i>C</i> → <i>iMC</i>
0 <i>m</i> 2 <i>C</i> 4	<i>e\$</i>	—
0 <i>m</i> 2 <i>C</i> 4 <i>e</i> 6	<i>\$</i>	<i>S</i> → <i>mCe</i>

5. (16 puntos) Análisis descendente Completar las tablas que siguen, dada la siguiente gramática (L es el axioma, y se usa el símbolo ε para representar la cadena vacía):

	PRIMERO	SIGUIENTE
$L \rightarrow dP \mid eP$	L	d, e
$P \rightarrow ODP \mid \varepsilon$	P	p, i, ε
$D \rightarrow iE$	D	i
$E \rightarrow wT \mid cT$	E	w, c
$T \rightarrow t \mid r$	T	t, r
$O \rightarrow p \mid \varepsilon$	O	p, ε
		$\$$
		$p, i, \$$
		$p, i, \$$
		i

(espacio para indicar el proceso de obtención del first y el follow)

Además de completar la tabla anterior, aquí habría que poner el proceso de obtención del FIRST y el FOLLOW.

TASP

	d	e	i	w	c	t	r	p	$\$$
L	$L \rightarrow dP$	$L \rightarrow eP$							
P			$P \rightarrow ODP$					$P \rightarrow ODP$	$P \rightarrow \varepsilon$
D			$D \rightarrow iE$						
E				$E \rightarrow wT$	$E \rightarrow cT$				
T						$T \rightarrow t$	$T \rightarrow r$		
O			$O \rightarrow \varepsilon$					$O \rightarrow p$	

Primeros pasos de la traza de análisis para la cadena $dict\$$

PILA	ENTRADA	SALIDA
$\$ L$	$d i c t \$$	$L \rightarrow dP$
$\$ P d$	$d i c t \$$	
$\$ P$	$i c t \$$	$P \rightarrow ODP$
$\$ P D O$	$i c t \$$	$O \rightarrow \varepsilon$
$\$ P D$	$i c t \$$	$D \rightarrow iE$
$\$ P E i$	$i c t \$$	
$\$ P E$	$c t \$$	$E \rightarrow cT$
$\$ P T c$	$c t \$$	
$\$ P T$	$t \$$	$T \rightarrow t$

6. (6 puntos) Eliminación de recursividad Dado el siguiente esquema de traducción.

$A \rightarrow A_1 Y$	{ $A.a = and(A_1.a, Y.a);$ $A.p = union(A_1.p, Y.p)$ }
$A \rightarrow X$	{ if $X.a > X.p$ then begin $A.p = X.p; A.a = X.a;$ end else begin $A.p = X.a; A.a = X.p;$ end }

Transformarlo en otro que no sea recursivo a izquierdas.

$A \rightarrow X$	{ if $X.a > X.p$ then begin $R.hp = X.p; R.ha = X.a;$ end else begin $R.hp = X.a; R.ha = X.p;$ end }
R	{ $A.a = R.sa; A.p = R.sp;$ }
$R \rightarrow Y$	{ $R_1.ha = and(R.ha, Y.a); R_1.hp = union(R.hp, Y.p);$ }
R_1	{ $R.sa = R_1.sa; R.sp = R_1.sp;$ }
$R \rightarrow \epsilon$	{ $R.sa = R.ha; R.sp = R.hp;$ }

7. (16 puntos) Generación de código intermedio Tradúzcase a código de tres direcciones las proposiciones ejecutables de la siguiente porción de código (proporcionar también el árbol de análisis sintáctico):

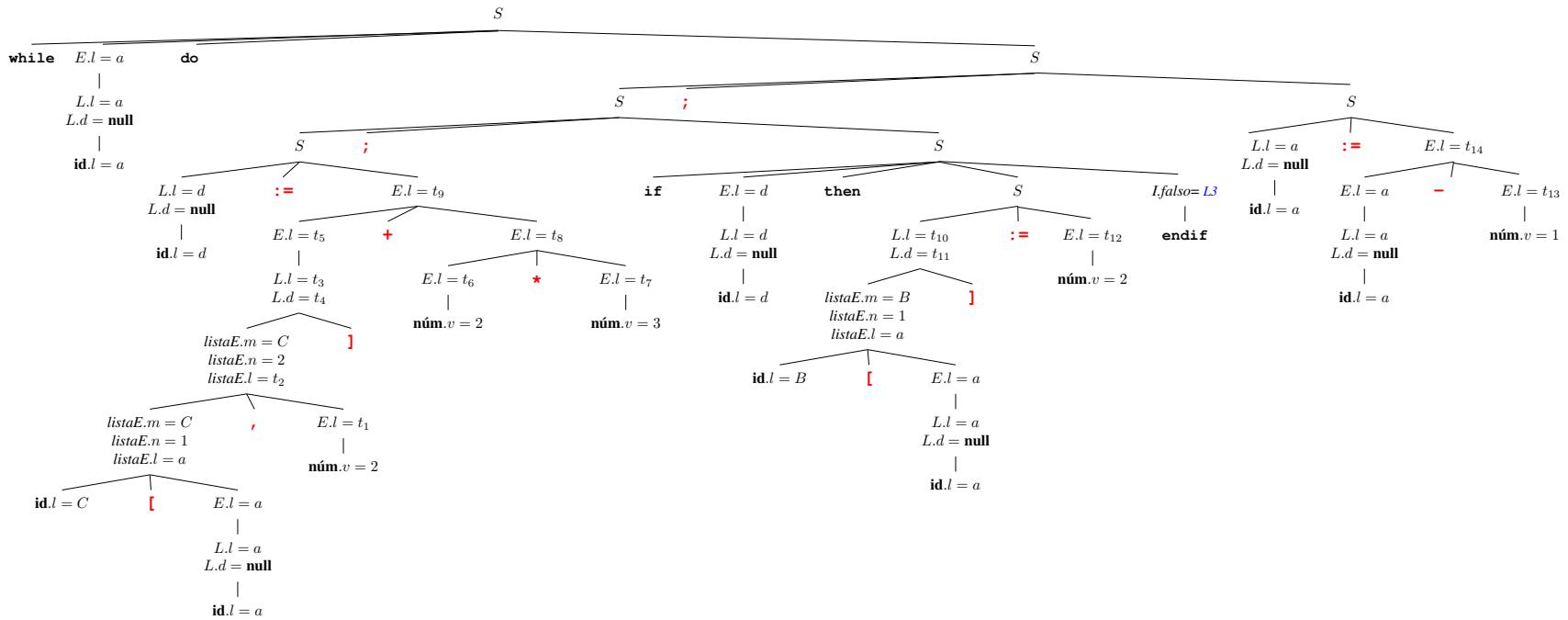
```

while (a) {
    d:=C[a,2]+2*3;
    if (d) B[a]:=2;
    a:=a-1;
}
```

$S \rightarrow L := E$	{ if $L.desplazamiento = null$ then $emite(L.lugar := E.lugar);$ else $emite(L.lugar ['L.desplazamiento '1] := E.lugar);$ }
$E \rightarrow L$	{ if $L.desplazamiento = null$ then $E.lugar := L.lugar$ else begin $E.lugar := tempnuevo;$ $emite(E.lugar := L.lugar ['L.desplazamiento '1]);$ end }
$L \rightarrow listaE]$	{ $L.lugar := tempnuevo;$ $L.desplazamiento := tempnuevo;$ $emite(L.lugar := c(listaE.matriz));$ $emite(L.desplazamiento := listaE.lugar * ancho(listaE.matriz));$ }
$L \rightarrow id$	{ $L.lugar := id.lugar;$ $L.desplazamiento := null$ }
$listaE \rightarrow listaE_1, E$	{ $t := tempnuevo;$ $m := listaE_1.ndim + 1;$ $emite(t := listaE_1.lugar * limite(listaE_1.matriz, m));$ $emite(t := E.lugar + t);$ $listaE.matriz := listaE_1.matriz;$ $listaE.lugar := t;$ $listaE.ndim := m$ }
$listaE \rightarrow id [E$	{ $listaE.matriz := id.lugar;$ $listaE.lugar := E.lugar;$ $listaE.ndim := 1$ }

```
lbl0: if a=0 goto lbl1
      t1:=2
      t2:=a*limite(C,2)
      t2:=t1+t2
      t3:=c(C)
      t4:=t2*ancho(C)
      t5:=t3[t4]
      t6:=2
      t7:=3
      t8:=t6*t7
      t9:=t5+t8
      d:=t9
      if d=0 goto lbl2
      t10:=c(B)
      t11:=a*ancho(B)
      t12:=2
      t10[t11]:=t12
      lbl2: t13:=1
             t14:=a-t13
             a:=t14
             goto lbl0
      lbl1:
```

Para conseguir la máxima calificación para este ejercicio, además de la secuencia anterior, sería necesario dibujar el árbol de análisis sintáctico anotado.



(página extra para anotaciones) (página extra para anotaciones)