



Apellidos, nombre:

DNI:

Instrucciones: Este enunciado y todos los folios usados deben entregarse al salir

Parte I: PREGUNTAS TIPO TEST. 30%. Cada dos respuestas incorrectas anulan una correcta.

1. Un lenguaje de programación en el que se requiera la declaración de variables previa a su uso **no** puede ser:
  - a) Libre de contexto.
  - b) Sensible al contexto.
  - c) De estructura de frase.
2. Una *máquina virtual*:
  - a) Traduce el código intermedio a código máquina.
  - b) Es un intérprete para un lenguaje de bajo nivel.
  - c) Es un intérprete para un lenguaje de alto nivel.
3. Elige la respuesta **incorrecta** acerca de los *compiladores interpretados*:
  - a) No generan código máquina ni ensamblador.
  - b) Generan una salida que se ejecutará a más velocidad que la generada por un compilador normal.
  - c) Generan una salida más portable que la generada por un compilador normal.
4. Dada una especificación léxica de Flex con la siguiente lista de expresiones regulares:  
(aa)\*  
ab+  
ba+  
¿Cuál de las siguientes cadenas **no** se podría procesar satisfactoriamente?
  - a) aaabbab
  - b) abbaabaa
  - c) abbbba
5. En un lenguaje con las palabras clave **if**, **while**, **for**, con identificadores que comiencen siempre por letra o dígito, números sólo naturales y los operadores +, -, \*, / y =, la expresión regular que describe en formato Flex las cadenas erróneas es:
  - a)  $[^a-zA-Z0-9+-=*/]+$ .
  - b)  $[^0-9+=a-zA-Z*/A-Z-]+$ .
  - c)  $[a-zA-Z0-9+=*/]+$ .
6. Considérese la siguiente gramática, recursiva por la izquierda:  
 $S \rightarrow X a \mid c$   
 $X \rightarrow S b$   
¿Cuál de las siguientes gramáticas es el resultado de eliminar correctamente la recursividad?
  - a)  $S \rightarrow X a \mid c$   
 $X \rightarrow c b \mid X a b$
  - b)  $S \rightarrow a X \mid c$   
 $X \rightarrow c b \mid c b X'$   
 $X' \rightarrow a b \mid \lambda$
  - c)  $S \rightarrow X a \mid c$   
 $X \rightarrow c b \mid c b X'$   
 $X' \rightarrow a b \mid a b X'$

7. La siguiente gramática:

$$\begin{array}{lcl} S & \rightarrow & A a y \\ A & \rightarrow & a x \\ & | & \lambda \end{array}$$

- a) Es LL(1).
- b) No es SLR(1).
- c) No es LL(1) pero sí SLR(1).

8. La gramática:

$$\begin{array}{lcl} S & \rightarrow & id := E \\ & | & if E = false then S \\ E & \rightarrow & id \\ & | & E \leq E \leq E \end{array}$$

- a) Es LL, recursiva por la izquierda y no propia.
- b) Es propia, no LL y no LR.
- c) Puede ser LR.

9. Una gramática **no** propia:

- a) Puede ser LL y LR.
- b) No puede ser LL, aunque sí SLR.
- c) No puede ser SLR, aunque sí LR-Canónica.

10. Supongamos que hemos calculado la colección LR(1) para la gramática:

$$\begin{array}{lcl} S & \rightarrow & aAd \mid bBd \mid aBe \mid bAe \\ A & \rightarrow & c \\ B & \rightarrow & c \end{array}$$

de modo que los conjuntos  $I_6$  e  $I_9$  contienen los siguientes items:

$$I_6 = \{[A \rightarrow c \bullet, d], [B \rightarrow c \bullet, e]\}$$

$$I_9 = \{[A \rightarrow c \bullet, e], [B \rightarrow c \bullet, d]\}$$

Sabiendo que la gramática es LR-canónica, indica la respuesta correcta:

- a) La gramática es LALR y SLR.
- b) La gramática es LALR aunque no tiene por qué ser SLR.
- c) La gramática no es LALR ni SLR.

11. Supongamos que estamos creando la colección LR(1) de una gramática, y tenemos que aplicar la operación de clausura al ítem  $[S \rightarrow A \cdot X Y Z, s]$ . La gramática tiene las siguientes reglas de producción:

$$\begin{array}{lcl} X & \rightarrow & x_1 \mid x_2 \\ Y & \rightarrow & y \mid \lambda \\ Z & \rightarrow & z \mid \lambda \end{array}$$

¿Cuáles de los siguientes ítems serían añadidos por aplicación de la operación clausura?

- a)  $[X \rightarrow \cdot x_1, y], [X \rightarrow \cdot x_2, y]$
- b)  $[X \rightarrow \cdot x_1, x_1/x_2], [X \rightarrow \cdot x_2, x_1/x_2]$
- c)  $[X \rightarrow \cdot x_1, y/z/s], [X \rightarrow \cdot x_2, y/z/s]$

12. Dada la siguiente gramática:

$$\begin{array}{lcl} E & \rightarrow & T * E \mid T \\ T & \rightarrow & n + T \mid n \mid (E) \end{array}$$

¿Cuál es el pivote de la forma sentencial derecha  $((n + n) * n)$ :

- a)  $((\underline{n} + n) * n)$
- b)  $((n + \underline{n}) * n)$
- c)  $((n + n) * \underline{n})$

13. Dada la gramática de la pregunta anterior, y la cadena de entrada  $((n + n) * n)$ , las tres primeras reducciones que realizaría un analizador LR serían:

- a) 1<sup>a</sup>)  $E \rightarrow T$ ; 2<sup>a</sup>)  $T \rightarrow n + T$ ; 3<sup>a</sup>)  $T \rightarrow n$ .
- b) 1<sup>a</sup>)  $T \rightarrow n + T$ ; 2<sup>a</sup>)  $T \rightarrow n$ ; 3<sup>a</sup>)  $E \rightarrow T$ .
- c) 1<sup>a</sup>)  $T \rightarrow n$ ; 2<sup>a</sup>)  $T \rightarrow n + T$ ; 3<sup>a</sup>)  $E \rightarrow T$ .

14. Dado el siguiente esquema de traducción para calcular el desplazamiento en memoria de las variables:

$$\begin{array}{ll}
 P & \rightarrow \quad \{desplazamiento = 0;\} \\
 & D \\
 D & \rightarrow T \text{ id}; \quad \{agregarTipo(id.entrada, T.tipo, desplazamiento); \\
 & \quad \quad \quad desplazamiento = desplazamiento + T.anchura;\} \\
 & D_1 \\
 D & \rightarrow \lambda \\
 T & \rightarrow \text{int} \quad \{T.tipo = integer; T.anchura = 4;\} \\
 T & \rightarrow \text{float} \quad \{T.tipo = float; T.anchura = 8;\}
 \end{array}$$

¿Cuál sería el desplazamiento de la variable **x** en el siguiente bloque de declaraciones?

```
int y; float z; int x;
```

- a) 12
- b) 8
- c) 4

15. El siguiente esquema de traducción:

$$\begin{array}{ll}
 S & \rightarrow A \{B.c = A.c\} B \\
 A & \rightarrow a A_1 \{A.c = A_1.c + 1\} \\
 A & \rightarrow a \{A.c = 1\} \\
 B & \rightarrow b \{B_1.c = B.c - 1\} B_1 \\
 B & \rightarrow b \{if (B.c - 1 = 0) printf(True); \\
 & \quad \quad \quad else printf(False);\}
 \end{array}$$

- a) Sólo usa atributos sintetizados y, por tanto, es S-Atribuida.
- b) Usa atributos tanto heredados como sintetizados, aunque no es L-Atribuida, pues algunos atributos heredan del hermano derecho.
- c) Usa atributos tanto heredados como sintetizados, y es L-Atribuida.

**Parte II: PROBLEMA. 70 %.**

La siguiente gramática  $G$  permite expresar operaciones aritméticas de suma, resta y factorial de números naturales expresados en notación postfija ( $V_T = \{+, -, !, 0, s, (, )\}$  y  $V_N = \{E, N\}$ ):

$$\begin{aligned} E &\rightarrow E E + \mid E E - \mid E ! \mid N \\ N &\rightarrow 0 \mid s(N) \end{aligned}$$

La representación de los números naturales se realiza empleando la notación  $s(N)$ , que permite expresar de forma recursiva el número siguiente de  $N$ . El caso base es 0. Por ejemplo, el número 3 se puede expresar como  $s(s(s(0)))$ .

1. (1 punto) Indica las características que tiene la gramática  $G$  que impiden que sea LL(1). Modifícala adecuadamente aplicando los algoritmos necesarios.
2. (1 punto) Deduce si la gramática obtenida en el apartado anterior es LL(1), calculando los conjuntos PRIMERO y SIGUIENTE, así como la tabla de análisis.
3. (1.5 puntos) Comprueba si la gramática  $G$  es SLR(1). Si no lo fuese, indica el modo de resolver los conflictos usando las precedencias adecuadas de los operadores.
4. (0.5 puntos) Indica de forma justificada si la gramática  $G$  es LR(1) y/o LALR(1), sin calcular ninguna colección de ítems adicional.
5. (0.75 puntos) Analiza la entrada  $s(0)+$  con el método ascendente predictivo usando la tabla obtenida en el apartado 3 y el método de tratamiento de errores en modo pánico.
6. (0.75 puntos) Comprueba si la gramática  $G$  es ambigua o bien da argumentos para demostrar que no lo es. ¿Sería necesario añadir a  $G$  expresiones entre paréntesis (E) como una opción adicional del no terminal E para poder realizar varias operaciones, teniendo en cuenta que  $!$  tiene más prioridad que  $+$  y  $-$ ?
7. (1.5 puntos)
  - a) Realiza una definición dirigida por la sintaxis (DDS) que permita obtener el valor numérico decimal que corresponde a la expresión analizada.
  - b) Decora el árbol de análisis para la siguiente entrada:  $s(0)s(0)+$
  - c) ¿La gramática atribuida obtenida es L-Atribuida? ¿Y S-Atribuida? Justifica las respuestas.