

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. ¿Cuál de los siguientes tipos de reglas de producción corresponden a las gramáticas que describen los componentes léxicos de un lenguaje de programación?

- a)  $A \rightarrow a, A \rightarrow aB$  o  $A \rightarrow \lambda$ , con  $A, B \in V_N$  y  $a \in V_T$ .
- b)  $A \rightarrow \alpha$ , con  $A \in V_N$  y  $\alpha \in (V_N \cup V_T)^*$ .
- c)  $\alpha A \beta \rightarrow \alpha \gamma \beta$ , con  $A \in V_N$ ,  $\alpha \beta \in (V_N \cup V_T)^*$  y  $\gamma \in (V_N \cup V_T)^+$ .

2. ¿Qué ventaja tiene usar centinelas al gestionar un buffer de dos mitades en un analizador léxico?

- a) Aumenta la longitud máxima de los lexemas que se pueden reconocer.
- b) Disminuye la cantidad de comprobaciones para verificar si es necesario recargar la siguiente mitad del buffer.
- c) Reduce la cantidad de lecturas de bloques de disco.

3. Dada la siguiente gramática:

$$\begin{aligned} A &\rightarrow B C A \mid a \\ B &\rightarrow b \mid \lambda \\ C &\rightarrow A B c \mid c \end{aligned}$$

¿cuál sería el conjunto SIGUIENTE( $B$ )?

- a) SIGUIENTE( $B$ ) =  $\{a, b, c\}$
- b) SIGUIENTE( $B$ ) =  $\{a, b, \$\}$
- c) SIGUIENTE( $B$ ) =  $\{b, \lambda\}$

4. La gramática anterior:

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

5. Dada la gramática  $G = (\{S\}, \{a, b\}, S, P = \{S \rightarrow abSba \mid aa\})$ , sin aplicarle transformaciones:

- a) Es ambigua.
- b) No es LL(1) pero sí es LL(2).
- c) Tiene ciclos.

6. Dada la siguiente gramática  $G$  con  $V_T = \{=, \vee, id, (, ), \neg, \Rightarrow\}$  y  $V_N = \{S, T\}$ , siendo  $P$ :

$$\begin{aligned} S &\rightarrow id = T \\ T &\rightarrow T \vee T \mid T \Rightarrow T \mid \neg T \mid ( T ) \mid id \end{aligned}$$

podemos afirmar

- a) la gramática es LL.
- b) la gramática ni es LL ni es LR.
- c) la gramática es SLR, pero no LL puesto que es recursiva por la izquierda.

7. Dada la gramática del ejercicio anterior, indica cual es el pivote de la forma sentencial

$$id = T \Rightarrow (T)$$

- a)  $id = \underline{T} \Rightarrow (T)$ .
- b)  $id = T \Rightarrow \underline{(T)}$ .
- c)  $id = T \Rightarrow (T)$ .

8. En una tabla de análisis LR, los conflictos desplaza/desplaza:

- a) Sólo se producen si la gramática es ambigua.
- b) Pueden producirse con gramáticas no ambiguas que no sean LR.
- c) No pueden producirse.

9. Dada la siguiente gramática:

$$\begin{aligned} X &\rightarrow Y Z \\ Y &\rightarrow a Y \mid \lambda \\ Z &\rightarrow T \mid \lambda \\ T &\rightarrow T a \mid a \end{aligned}$$

el pivote de la forma sentencial derecha  $w \equiv a$ :

- a) Es  $Y$ .
- b) Es  $T$ .
- c) No es único.

10. Dada la siguiente gramática:

$$\begin{aligned} X &\rightarrow Y Y T \\ Y &\rightarrow Z Z \\ Z &\rightarrow a a \\ T &\rightarrow T b \mid b \end{aligned}$$

y la colección  $LR(0)$  asociada a ella:

$$I_0 = \{ \begin{aligned} &X' \rightarrow \cdot X \\ &X \rightarrow \cdot Y Y T \\ &Y \rightarrow \cdot Z Z \\ &Z \rightarrow \cdot a a \end{aligned} \}$$

$$I_1 = \text{GOTO}(I_0, X) = \{ X' \rightarrow X \cdot \}$$

$$I_2 = \text{GOTO}(I_0, Y) = \{ \begin{aligned} &X \rightarrow Y \cdot Y T \\ &Y \rightarrow \cdot Z Z \\ &Z \rightarrow \cdot a a \end{aligned} \}$$

$$I_3 = \text{GOTO}(I_0, Z) = \{ \begin{aligned} &Y \rightarrow Z \cdot Z \\ &Z \rightarrow \cdot a a \end{aligned} \}$$

$$I_4 = \text{GOTO}(I_0, a) = \{ Z \rightarrow a \cdot a \}$$

$$I_5 = \text{GOTO}(I_2, Y) = \{ \begin{aligned} &X \rightarrow Y Y \cdot T \\ &T \rightarrow \cdot T b \\ &T \rightarrow \cdot b \end{aligned} \}$$

$$I_6 = \text{GOTO}(I_3, Z) = \{ Y \rightarrow Z Z \cdot \}$$

$$I_7 = \text{GOTO}(I_4, a) = \{ Z \rightarrow a a \cdot \}$$

$$I_8 = \text{GOTO}(I_5, T) = \{ \begin{aligned} &X \rightarrow Y Y T \cdot \\ &T \rightarrow T \cdot b \end{aligned} \}$$

$$I_9 = \text{GOTO}(I_5, b) = \{ T \rightarrow b \cdot \}$$

$$I_{10} = \text{GOTO}(I_8, b) = \{ T \rightarrow T b \cdot \}$$

- a) Es ambigua.
- b) Es LL y SLR.
- c) No es LL pero sí SLR.

11. Con respecto a un análisis descendente de una cadena derivada de la gramática anterior, si la pila del analizador contiene los símbolos  $\$X$ , y la entrada es  $\text{bbbbbbbbb}\$, ¿cómo se comportaría un analizador LL si tiene implementado un método de recuperación de errores en *modo pánico*?$

- a) Informaría del error, sacaría  $X$  de la pila y saltaría toda la entrada hasta el símbolo  $\$$ . Después, terminaría.
- b) Informaría del error, sacaría  $X$  de la pila, introduciría  $T$  y saltaría de la entrada los ocho primeros símbolos  $b$ . Después, continuaría hasta terminar correctamente.
- c) No daría mensaje. Simplemente sustituiría los ocho primeros símbolos  $b$  por ocho símbolos  $a$ .

12. Considerando las siguientes reglas de producción, ¿cuál sería el resultado de la factorización?:

$$A \rightarrow a b c d \mid a b c \mid a b$$

a)  $A \rightarrow a b A''$   
 $A'' \rightarrow c A' \mid \lambda$   
 $A' \rightarrow d \mid \lambda$

b)  $A \rightarrow a A'$   
 $A' \rightarrow b c d \mid b c \mid b$

c)  $A \rightarrow a b A'$   
 $A' \rightarrow c d \mid c \mid \lambda$

13. En relación con los esquemas de traducción y **bison**:

- a) **bison** sólo admite esquemas de traducción dirigidos por la sintaxis postfijos.
- b) **bison** sustituye cualquier acción en mitad del lado derecho por un no terminal ficticio con una  $\lambda$ -regla que tiene dicha acción al final.
- c) **bison** sólo puede procesar gramáticas S-atribuidas.

14. Dada la siguiente gramática  $G$  con  $V_T = \{\vee, \forall, \text{id}, (, ), ;\}$  y  $V_N = \{F, L\}$ , siendo  $P$ :

$$\begin{aligned} F &\rightarrow F \vee F \mid \forall \text{id} ( F ) \mid \text{id} ( L ) \\ L &\rightarrow \text{id} \mid \text{id} ; L \end{aligned}$$

en un análisis ascendente para el reconocimiento de la siguiente forma sentencial  $\forall \text{id}(\forall \text{id}(\text{id}(\text{id};\text{id})))$  indica cuales serían las tres primera reducciones:

- a)  $F \rightarrow \forall \text{id} ( F )$ ,  $F \rightarrow \forall \text{id} ( F )$ ,  $F \rightarrow \text{id} (L)$  .
- b)  $L \rightarrow \text{id} ; L$ ,  $L \rightarrow \text{id}$ ,  $F \rightarrow \forall \text{id} ( F )$  .
- c)  $L \rightarrow \text{id}$ ,  $L \rightarrow \text{id} ; L$ ,  $F \rightarrow \text{id} (L)$  .

15. Entre las siguientes tareas, señala cual NO se podría realizar en tiempo de compilación:

- a) Comprobación de tipos y conversión de tipos.
- b) Generación de GDA.
- c) Comprobación dinámica.

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

Considerar la siguiente gramática:

$$\begin{array}{lcl} E & \rightarrow & C \\ & | & C \cup E \\ & | & C \cap E \\ C & \rightarrow & \{ L \} \\ & | & \emptyset \\ L & \rightarrow & id \\ & | & L, L \end{array}$$

que genera un lenguaje para operar con conjuntos. Por ejemplo, la gramática podría generar la siguiente cadena:

$$\{a, g\} \cup \{a, b, c, d, e, f\} \cap \{c, e\}$$

que representaría al conjunto  $\{a, c, e, g\}$ .

1. (2 puntos) Decir si se trata de una gramática LL(1), justificando la respuesta. Si no se tratara de una gramática LL(1), intentar transformarla para conseguir que lo sea teniendo en cuenta que el operador  $,$  debe ser asociativo por la derecha. Calcular los conjuntos PRIMERO y SIGUIENTE para cada no terminal y *predict* para cada regla. Razonar con estos últimos conjuntos si la gramática es LL(1), sin construir la tabla de análisis.
2. (2 puntos) Calcular la colección LR(1) sobre la gramática original y la tabla de análisis para comprobar si se trata de una gramática LALR. En caso de que aparezcan conflictos en la tabla, eliminarlos teniendo en cuenta que el operador  $,$  es asociativo por la derecha.
3. (0.75 puntos) ¿Es la gramática SLR? ¿Y LR-Canónica?
4. (0.5 puntos) Simular el comportamiento de algoritmo ascendente predictivo para reconocer la cadena  $\{id, \emptyset\}$ , aplicando el método de recuperación de errores en modo pánico en caso de error.
5. (1.75 puntos) Supongamos que trabajamos con conjuntos de *id*, en los que no se permite la mezcla de elementos de diferentes tipos dentro de un mismo conjunto, ni operaciones ( $\cup$  o  $\cap$ ) con conjuntos de tipos diferentes. Supongamos también que en la tabla de símbolos se almacena el tipo de cada token *id*. Realizar una *definición dirigida por la sintaxis (DDS)* para comprobar cuál es el primer conjunto que incumple esta condición, teniendo en cuenta que el conjunto  $\emptyset$  no tiene tipo (puede operar con conjuntos de cualquier tipo). Es decir, si los tokens *x* e *y* son de tipo *int*, *a* de tipo *char* y *b* de tipo *float*, en la sentencia:

$$\emptyset \cup \{x, y\} \cup \{a\} \cap \{b, y\} \cup \emptyset$$

el primer conjunto erróneo sería el tercero. Se pide que, como resultado de la *DDS*, se almacene el índice (entero) del conjunto en el que falla la comprobación en un atributo del nodo raíz del árbol, o bien el valor 0 si no hay fallo en dicha comprobación.

Para ello definir los atributos que sean necesarios, indicando si son sintetizados o heredados. Decorar el árbol sintáctico correspondiente a la entrada anterior.

Decir si la gramática es S-Atribuida y/o L-Atribuida.