



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. La realización de un intérprete de bajo nivel:
 - a) No tiene sentido en ningún caso, puesto que los intérpretes suelen usarse en entornos interactivos para lenguajes como SQL o Caml.
 - b) Puede ser interesante en situaciones en las que deseemos una máquina no implementada en hardware.
 - c) No podría realizarse puesto que tendríamos que implementarlo en lenguaje máquina.
2. Dentro de la fase de análisis en el proceso de compilación:
 - a) El análisis sintáctico usa técnicas sistemáticas para construir la tabla de símbolos, que implementa el autómata de pila para guiar el análisis.
 - b) El análisis semántico no suele realizarse mediante la simulación de una máquina abstracta, sino con técnicas específicas que suelen usar una tabla de símbolos.
 - c) La fase de análisis léxico es la única en la que se accede a la tabla de símbolos.
3. Una de estas tres expresiones regulares *flex* reconoce el conjunto de cadenas no vacías formadas por letras minúsculas y no comenzando con tres a's seguidas:
 - a) `NotresAs (a|aa|a{4,})`
`%%`
`{NotresAs}[b-z]*`
 - b) `tresAs "aaa"`
`%%`
`[^{\tresAs}][b-z]*|([b-z]+[a-z]*)`
 - c) `(a|aa)([b-z][a-z]*)?[b-z][a-z]*`

4. La siguiente gramática:

$$\begin{aligned} S &\rightarrow T V \\ T &\rightarrow t \mid \lambda \\ V &\rightarrow v \mid t \mid \lambda \end{aligned}$$

- a) Es ambigua.
 - b) Es LL.
 - c) Es LR.
5. Dada la siguiente gramática G con $V_T = \{\vee, \forall, \text{id}, (,), ;\}$ y $V_N = \{F, L\}$, siendo P :

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

podemos afirmar:

- a) G no es propia.
 - b) G , una vez que se factorice y se elimine recursividad por la izquierda, es LL.
 - c) G no es LL ni LR.

6. Consideramos la gramática G siguiente:

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

a) G es LL(1), $\text{predict}(B \rightarrow \lambda) = \{c, \$\}$, y la tabla de análisis es

NO TERM	TERMINAL		
	<i>b</i>	<i>c</i>	<i>\$</i>
<i>A</i>	$A \rightarrow B C$	$A \rightarrow B C$	$A \rightarrow B C$
<i>B</i>	$B \rightarrow b c B$	$B \rightarrow \lambda$	$B \rightarrow \lambda$
<i>C</i>		$C \rightarrow c C$	$C \rightarrow \lambda$

b) G no es LL(1), $\text{predict}(B \rightarrow \lambda) = \{b, c, \$\}$, y la tabla de análisis es

NO TERM	TERMINAL		
	<i>b</i>	<i>c</i>	<i>\$</i>
<i>A</i>	$A \rightarrow B C$	$A \rightarrow B C$	$A \rightarrow B C$
<i>B</i>	$B \rightarrow b c B$ $B \rightarrow \lambda$	$B \rightarrow \lambda$	$B \rightarrow \lambda$
<i>C</i>		$C \rightarrow c C$	$C \rightarrow \lambda$

c) G no es LL(1) puesto que $\text{predict}(B \rightarrow \lambda) \cap \text{predict}(C \rightarrow \lambda) = \{\$\}$.

7. Consideremos la siguiente tabla LL

NO TERM	TERMINAL			
	<i>a</i>	<i>b</i>	<i>c</i>	<i>\$</i>
<i>S</i>	$S \rightarrow a$	$S \rightarrow b S c S b$		

para la gramática

$$S \rightarrow a \mid b S c S b$$

Con respecto a la simulación descendente predictiva, señalar la afirmación incorrecta:

- Si en la pila de análisis aparece $\$bScS$ y en la entrada $acab\$$, en el siguiente paso de cálculo, la pila contendría $\$bSca$ y la entrada $acab\$$.
- Si en la pila de análisis aparece $\$bScS$ y en la entrada $cab\$$, en el siguiente paso de cálculo, se daría un mensaje de error, la pila contendría $\$bSc$ y la entrada $cab\$$, suponiendo una recuperación de errores en modo pánico.
- Si en la pila de análisis aparece $\$bScS$ y en la entrada $cab\$$, en el siguiente paso de cálculo, se daría un mensaje de error, la pila contendría $\$bScS$ y la entrada $acab\$$, suponiendo una recuperación de errores en modo pánico.

8. La siguiente gramática:

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

genera el lenguaje:

- $\{a^n b^m c^r \mid n > 0, m > 0, r > n + m\}$
- $\{a^n b^m c^{m+n} \mid n > 0, m > 0\}$
- $\{a^n b^m c^{m+n} \mid n \geq 0, m \geq 0\}$

9. La gramática anterior:

- Es propia.
- Es LL, puesto que los predict asociados a reglas con el mismo no terminal a la izquierda, son disjuntos.
- No es SLR.

10. El conjunto I_0 de la colección de items LR(0) para la gramática anterior es:

- $\{[S' \rightarrow \bullet S], [S \rightarrow \bullet a S c], [S \rightarrow \bullet B], [S \rightarrow \bullet]\}$.
- $\{[S' \rightarrow \bullet S], [S \rightarrow \bullet a S c], [S \rightarrow \bullet B], [B \rightarrow \bullet b B c]\}$.
- $\{[S' \rightarrow \bullet S], [S \rightarrow \bullet a S c], [S \rightarrow \bullet B], [S \rightarrow \bullet], [B \rightarrow \bullet b B c], [B \rightarrow \bullet]\}$.

11. Supongamos que se realiza un análisis SLR de la gramática siguiente

$$L \rightarrow L \odot L \mid L \oslash L \mid \Delta$$

y uno de los conjuntos de items es el siguiente:

$$I_j = \{ [L \rightarrow L \odot L \bullet], [L \rightarrow L \bullet \odot L], [L \rightarrow L \bullet \oslash L] \}$$

de manera que, en la tabla de análisis, la fila correspondiente al estado j quedaría así:

ESTADO	ACCIÓN				IR-A
	\odot	\oslash	Δ	\$	L
	...				
j	r1/di	r1/dk		r1	
	...				

Si \odot y \oslash son asociativos por la izquierda, y \oslash tiene menor precedencia que \odot , para eliminar los conflictos:

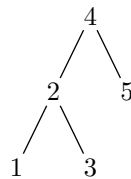
- Debemos elegir las reducciones en ambos los casos.
- Debemos elegir el desplazamiento en las casilla $[j, \odot]$ y la reducción en $[j, \oslash]$.
- Debemos elegir la reducción en la casilla $[j, \odot]$ y desplazamiento en $[j, \oslash]$.

12. Dada la siguiente gramática G con $V_T = \{\in, (,), \text{num}\}$ y $V_N = \{S, B\}$, siendo P :

$$\begin{aligned} S &\rightarrow \text{num} \in B \\ B &\rightarrow (B \text{ num } B) \mid \lambda \end{aligned}$$

podemos afirmar:

- El no terminal B es recursivo por la izquierda y por la derecha.
 - La gramática G no es LL(1) ni LR(1) porque es ambigua.
 - La gramática es LL(1).
13. Supongamos que, en la gramática anterior, B representa un árbol binario de búsqueda, y S representa una operación de búsqueda de un número en un árbol dado. Por ejemplo, $5 \in (((1)2(3))4(5))$ representa la búsqueda del valor 5 en el árbol:



La siguiente DDS describe una posible operación de búsqueda mediante reglas semánticas:

Regla de producción	Regla semántica
$S \rightarrow \text{num} \in B$	$B.v = \text{num}.v; S.res = B.res$
$B \rightarrow (B_1 \text{ num } B_2)$	$B_1.v = B.v; B_2.v = B.v;$ if $(\text{num}.v == B.v)$ $B.res = \text{true};$ else if $(\text{num}.v < B.v)$ $B.res = B_1.res;$ else $B.res = B_2.res;$
$B \rightarrow \lambda$	$B.res = \text{false};$

donde $\text{num}.v$ es un atributo que contiene el valor numérico del token num . Podemos afirmar:

- La gramática es S-atribuida.
- La gramática es L-atribuida.
- Ninguna de las anteriores, porque la gramática tiene atributos heredados por la derecha.

14. Dada la sentencia $5 \in (((1)2(3))4(5))$, el pivote¹ es:
- $5 \in (((\lambda)2(3))4(5))$
 - $5 \in (((1\lambda)2(3))4(5))$
 - $5 \in (((1)2(3))4(5))$
15. Si ampliamos la gramática G con las reglas $B \rightarrow B + \text{num}$ y $B \rightarrow B - \text{num}$ que representan, respectivamente, la operación de añadir y eliminar un valor del árbol binario, ¿qué algoritmo habría que aplicar en primer lugar para intentar obtener una gramática $LL(1)$ equivalente?
- Eliminar las λ reglas.
 - Eliminar la recursión inmediata de B .
 - Eliminar los factores comunes de las reglas de B .

Parte II: PROBLEMA. 70 %.

Sea G la gramática con $V_T = \{I, V, X, IV, IX, XL\}$ y $V_N = \{S, Decenas, Unidades, Dec, Unid\}$, con P :

S	$\rightarrow Decenas\ Unidades$
$Decenas$	$\rightarrow Dec\ \ XL$
Dec	$\rightarrow Dec\ X\ \ \lambda$
$Unidades$	$\rightarrow Unid\ \ IV\ \ V\ Unid\ \ IX$
$Unid$	$\rightarrow Unid\ I\ \ \lambda$

G genera el lenguaje de los números romanos menores que 50.

- (1 punto) Decir si se trata de una gramática $LL(1)$, justificando la respuesta. Calcular los conjuntos PRIMERO y SIGUIENTE para cada no terminal y *predict* para cada regla.
- (2.5 puntos) Calcular la colección $LR(1)$ y la tabla de análisis LR -Canónica para la gramática original. Decir si la gramática es LR -Canónica, $LALR$ y SLR , justificando las respuestas.
- (1 punto) Simular el comportamiento de algoritmo ascendente predictivo para reconocer la cadena **VIIIX**, aplicando el método de *recuperación de errores en modo pánico* en caso de error. Debemos tener en cuenta que el *analizador léxico* asociado a G , debería reconocer siempre los tokens asociados a los lexemas más largos.
- (1 punto) Supongamos que modificamos G para definir una nueva gramática G' , con $V'_T = \{I, V, X, L\}$ y $V_N = \{S, Decenas, Unidades, Dec, Unid\}$, y P' :

S	$\rightarrow Decenas\ Unidades$
$Decenas$	$\rightarrow Dec\ \ X\ L$
Dec	$\rightarrow Dec\ X\ \ \lambda$
$Unidades$	$\rightarrow Unid\ \ I\ V\ \ V\ Unid\ \ I\ X$
$Unid$	$\rightarrow Unid\ I\ \ \lambda$

Decir en este caso, si G' sería SLR , $LALR$ y/o LR -Canónica, sin construir ninguna tabla de análisis adicional.

- (1.5 puntos) Dar una definición dirigida por la sintaxis (DDS) para G , que permita obtener en un atributo sintetizado de S el valor decimal del número romano definido. Además, la DDS debe restringir el número de **X** en Dec y de **I** en $Unid$ a 3, es decir, que si en la entrada aparecen más de 3 veces consecutivas uno de estos símbolos, se ha de emitir el error semántico oportuno. Por ejemplo, para una cadena de entrada como **XXI** la traducción obtenida ha de ser **21** y para la cadena de entrada **VIIII** se ha de obtener un error semántico.

Toda la información debe pasarse a través de atributos y no es posible utilizar ninguna variable global. Se debe indicar claramente, además, de qué tipo es cada uno de los atributos utilizado.

Decorar el árbol de análisis para la sentencia **XXIV**.

Finalmente, decir si la gramática es S -atribuida y/o L -atribuida.

ACLARACIÓN:

Los números romanos cumplen las siguientes reglas:

- Si a la derecha de una cifra romana se escribe otra igual o menor, el valor de esta se suma a la anterior (ej. **XXI** es 21).
- La cifra **I** colocada delante de la **V** o la **X** les resta una unidad, la **X** precediendo a la **L** o la **C** les resta diez unidades y la **C** delante de la **D** o la **M** les resta cien unidades (ej. **IX** es 9).
- En ningún número se puede poner una misma letra más de tres veces seguidas.

¹ λ representa la cadena vacía.