

**EXAMEN DE COMPILADORES (2º Grado en Informática, final septiembre-2013)****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 gramática $G = (\{i, a, b\}, \{S, C\}, S, P)$, con el conjunto de reglas P siguiente:

$$S \rightarrow i C i$$

$$| i i$$

$$i C i \rightarrow i a i$$

$$| i b i$$

- a) Es una gramática regular para representar cadenas que comienzan y acaban por el símbolo i , aunque podría especificarse de forma más simple usando expresiones regulares.
- b) Es sensible al contexto y podría representar algunos aspectos sensibles al contexto de un lenguaje de programación, por ejemplo, la necesidad de declarar los identificadores.
- c) Está mal construida.
2. El preprocesador de C:
- a) Es un traductor.
- b) Es un intérprete.
- c) No es ni un traductor ni un intérprete.
3. El analizador léxico simula:
- a) Un Autómata Linealmente Acotado.
- b) Un Autómata de Pila.
- c) Un Autómata Finito.
4. Decir cuál de las siguientes afirmaciones es **falsa**:
- a) Los comentarios anidados pueden generarse con una gramática regular, aunque no suelen permitirse por cuestiones de eficiencia.
- b) Los comentarios anidados no pueden generarse con una gramática regular, por eso no suelen permitirse.
- c) Los comentarios anidados podrían reconocerse con la herramienta flex usando condiciones de arranque o de contexto.
5. Si partimos de una gramática libre de contexto y no LL, con un analizador descendente recursivo:
- a) No podría analizarse ninguna sentencia.
- b) Podría analizarse sin problemas cualquier sentencia, aunque sería más ineficiente que si tuviéramos una gramática LL.
- c) Podría entrar en bucles infinitos con algunas sentencias.

6. Sea G la gramática con las producciones:

$$S \rightarrow 0 S 1 \mid 0 1$$

Decir cuál de las siguientes es una gramática equivalente a G y LL(1).

- a) $S \rightarrow 0 S S' \mid 1$
 $S' \rightarrow 1 \mid \lambda$
- b) $S \rightarrow 0 S' 1$
 $S' \rightarrow 0 S' 1 \mid S''$
 $S'' \rightarrow 0 1 \mid \lambda$
- c) $S \rightarrow 0 S'$
 $S' \rightarrow S 1 \mid 1$

7. Sea G la gramática con las producciones:

$$S \rightarrow 0 S 1 \mid 0 1$$

El pivote de la sentencia '00001111' sería:

- a) 00001111
- b) 00001111
- c) 00001111

8. Sea G la gramática con las producciones:

$$S \rightarrow 0 S 1 \mid 0 1$$

Un analizador ascendente predictivo realizaría la siguiente secuencia de reducciones para reconocer la sentencia '000111':

- a) r2 r1 r1
- b) r1 r2 r1
- c) r2 r2 r1

9. Considera la siguiente gramática:

$$\begin{aligned} S &\rightarrow A (S) B \mid \lambda \\ A &\rightarrow S \mid S B \mid x \mid \lambda \\ B &\rightarrow S B \mid y \end{aligned}$$

¿Cuál es el conjunto de ítems I_0 de la colección LR(0) de la gramática de la pregunta anterior?

- a) $\{ [S' \rightarrow \cdot S], [S \rightarrow \cdot A (S) B], [S \rightarrow \cdot], [A \rightarrow \cdot S], [A \rightarrow \cdot S B], [A \rightarrow \cdot x], [A \rightarrow \cdot], [B \rightarrow \cdot S B], [B \rightarrow \cdot y] \}$
- b) $\{ [S' \rightarrow \cdot S], [S \rightarrow \cdot A (S) B], [S \rightarrow \cdot], [A \rightarrow \cdot S], [A \rightarrow \cdot S B], [A \rightarrow \cdot x], [A \rightarrow \cdot] \}$
- c) $\{ [S' \rightarrow \cdot S], [S \rightarrow \cdot A (S) B], [A \rightarrow \cdot S], [A \rightarrow \cdot S B], [A \rightarrow \cdot x] \}$

10. ¿Qué tipo de conflictos se producen en el conjunto I_0 de la pregunta anterior?

- a) Reduce-reduce.
- b) Desplaza-reduce.
- c) Desplaza-reduce y reduce-reduce.

11. Dada la siguiente gramática:

$stmt \rightarrow var \mid if_stmt$
 $if_stmt \rightarrow if\ var\ then\ stmt \mid if\ var\ then\ stmt\ else\ stmt$
 $var \rightarrow a \mid b \mid win \mid loss$

¿Qué tipo de conflictos se producen al analizar la entrada “if a then if b then win else loss”?

- a) Reduce-reduce.
- b) Desplaza-reduce.
- c) Ninguno.

12. Sea G la gramática con las producciones:

$S \rightarrow S (S) S \mid \lambda$

que genera el conjunto de paréntesis balanceados.

- a) G es propia, no LL y no LR-Canónica.
- b) G es no propia, no LL, no SLR, pero sí LR-Canónica.
- c) G es no propia, ambigua, no LL y no SLR.

13. Dada la siguiente gramática:

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid -TE' \mid \lambda$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid /FT' \mid \lambda$
 $F \rightarrow (E) \mid num$

Si construimos un analizador descendente predictivo recursivo, ¿cuántas veces entrará en el método del no terminal E' con la entrada $2+3*4$?

- a) 2
- b) 3
- c) 0

14. Para implementar una calculadora que interpretase expresiones aritméticas partiendo de la gramática anterior, ¿qué tipo de gramática atribuida habría que crear?

- a) Una que fuese sólo L-atribuida.
- b) Una que fuese sólo S-atribuida.
- c) Una que fuese tanto S-atribuida como L-atribuida.

15. Los lenguajes C y C++:

- a) Siempre usan equivalencia estructural.
- b) Siempre usan equivalencia de nombre.
- c) Usan equivalencia estructural, salvo en estructuras y uniones, para las que usan equivalencia de nombre.

Parte II: PREGUNTAS CORTAS. 10 %.

La siguiente gramática:

$baseNum \rightarrow num\ basechar$

$basechar \rightarrow octal$

$basechar \rightarrow decimal$

$num \rightarrow num\ dig$

$num \rightarrow dig$

$dig \rightarrow 0$

$dig \rightarrow 1$

$dig \rightarrow 2$

$dig \rightarrow 3$

$dig \rightarrow 4$

$dig \rightarrow 5$

$dig \rightarrow 6$

$dig \rightarrow 7$

$dig \rightarrow 8$

$dig \rightarrow 9$

es una gramática libre de contexto, expresada en notación BNF

1. ¿Crees que existe una gramática regular equivalente? Justifica la respuesta.
2. ¿Cuáles son los símbolos terminales o tokens de esta gramática?
3. Si estas reglas formaran parte de una gramática más general, que implementara una calculadora, por ejemplo, ¿descargarías parte del reconocimiento de los números en octal y decimal en el analizador léxico en lugar de usar estas reglas? En ese caso, ¿qué símbolos terminales o tokens definirías para que fueran devueltos por el analizador léxico?

Parte III: PROBLEMA. 60 %.

Considerar la siguiente gramática G:

$$\begin{array}{lcl} P & \rightarrow & [L] (num) \\ L & \rightarrow & E \\ & | & L , E \\ E & \rightarrow & num \\ & | & P \end{array}$$

que genera un lenguaje para evaluación de polinomios. Por ejemplo, la gramática podría generar la siguiente cadena:

$$[1, 3, [2, 1](2)](1)$$

donde los números de la lista entre corchetes representarían los coeficientes de un polinomio de una variable, y el número entre paréntesis, el valor que se quiere asignar a la variable para evaluar el polinomio. Por ejemplo, la frase $[1, 2, 6](5)$ representaría al polinomio $x^2 + 2x + 6$, que deberá evaluarse asignando a x el valor 5.

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. Calcular los conjuntos PRIMERO y SIGUIENTE para cada no terminal y *predict* para cada regla. Construir la tabla de análisis. Decir si la nueva gramática es LL(1).
2. (2.5 puntos) Construir la colección de items LR(1) para la gramática G original y comprobar si se trata de una gramática LR-Canónica y LALR. Razonar, también, sin construir la colección LR(0), si la gramática es SLR.
3. (1.5 puntos) Supongamos que queremos construir un traductor simple para evaluar polinomios de una variable. Es decir, para la entrada $[1, 3, [2, 1](2)](1)$ la salida debería ser 9. Realizar una definición dirigida por la sintaxis para traducir una entrada generada por la gramática anterior en el valor numérico correspondiente, de forma que el polinomio se vaya evaluando al tiempo que se hacen las reducciones en la lista de coeficientes. Definir los atributos que sean necesarios, con la restricción de que en la solución no se pueden emplear únicamente atributos sintetizados. Indicar de forma justificada si la gramática es L-atribuida. Decorar finalmente el árbol sintáctico correspondiente a la entrada $[1, 3, [2, 1](2)](1)$.