

EXAMEN DE COMPILADORES (2º Grado en Informática, final enero-2014)
--

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. El preprocesador y el ensamblador:
 - a) Son lenguajes necesarios para que se realice correctamente el proceso de traducción en algunos lenguajes, como C.
 - b) Son traductores, que generalmente complementan al compilador.
 - c) Son intérpretes.
2. El primer compilador:
 - a) Se implementó en los años 30, y traducía BASIC.
 - b) Se implementó en lenguaje de alto nivel.
 - c) Necesitó 18 años/persona para ser implementado.
3. En el contexto de la jerarquía de Chomsky:
 - a) No existen gramáticas para reconocer las restricciones de contexto, como el uso correcto de parámetros en una función. Por eso se usan las gramáticas atribuidas.
 - b) Existen gramáticas para reconocer las restricciones de contexto, como el uso correcto de parámetros en una función. Sin embargo, se usan las gramáticas atribuidas.
 - c) Usamos las gramáticas sensibles al contexto para reconocer las restricciones de contexto, como el uso correcto de parámetros en una función.
4. Dada la siguiente especificación léxica, usada para generar un analizador con Flex:
 $a(ba)^*$
 $b^*(ab)^*$
 abd
 d^+
¿cuál de las siguientes afirmaciones es correcta?
 - a) La cadena $ababdddd$ se descompondrá en tokens con lexemas $abab$ y $dddd$.
 - b) La cadena $ababddababa$ se descompondrá en tokens con lexemas ab , abd , d y $ababa$.
 - c) La cadena $dddabbabab$ se descompondrá en tokens con lexemas ddd , a y $bbabab$.
5. La siguiente gramática:
$$\begin{array}{lcl} S & \rightarrow & S A \\ & | & A \\ A & \rightarrow & id = L ; \\ L & \rightarrow & id \\ & | & L = L \end{array}$$
 - a) Es propia.
 - b) No es propia, pues es recursiva por la izquierda.
 - c) No es propia, pues es ambigua.
6. La gramática anterior:
 - a) Es LL(1) y SLR(1).
 - b) No es LL(1) aunque sí LR(1).
 - c) No es ni LL(1) ni SLR(1).

7. Si en la colección LR(0) de la gramática anterior obtenemos el estado

$$I_{10} \equiv \{ L \rightarrow L = L\bullet, L \rightarrow L\bullet = L \}$$

y la siguiente tabla:

ESTADO	accion				ir_a		
	id	=	;	\$	S	A	L
0	d3				1	2	
1	d3			aceptar		4	
2	r2			r2			
3		d5					
4	r1			r1			
5	d7						6
6		d9	d8				
7		r4	r4				
8	r3			r3			
9	d7						10
10		d9/r5	r5				

Para conseguir que el operador = sea asociativo por la derecha, en la casilla [10,=] debemos elegir la acción:

- a) r5
- b) d9
- c) Ninguna valdría.

8. Elige, de entre las opciones, una gramática no recursiva por la izquierda que NO sea equivalente a la siguiente:

$$\begin{aligned} A &\rightarrow (L) := (R) \\ L &\rightarrow \text{id} \\ &\quad | \quad L , \text{id} \\ R &\rightarrow \text{num} \\ &\quad | \quad R , \text{num} \end{aligned}$$

a)

$$\begin{aligned} A &\rightarrow (L) := (R) \\ L &\rightarrow \text{id} \\ &\quad | \quad \text{id} , L \\ R &\rightarrow \text{num} \\ &\quad | \quad \text{num} , R \end{aligned}$$

b)

$$\begin{aligned} A &\rightarrow (L) := (R) \\ L &\rightarrow \lambda \\ &\quad | \quad \text{id} , L \\ R &\rightarrow \lambda \\ &\quad | \quad \text{num} , R \end{aligned}$$

c)

$$\begin{aligned} A &\rightarrow (L) := (R) \\ L &\rightarrow \text{id} \mid \text{id} L' \\ L' &\rightarrow , \text{id} \mid , \text{id} L' \\ R &\rightarrow \text{num} \mid \text{num} R' \\ R' &\rightarrow , \text{num} \mid , \text{num} R' \end{aligned}$$

9. En la siguiente gramática:

$$\begin{aligned} \text{MATRIZ} &\rightarrow (\text{FILA} \text{ FILAS}) \\ \text{FILA} &\rightarrow \text{num} \text{ NUMEROS} ; \\ \text{FILAS} &\rightarrow \text{FILA} \text{ FILAS} \\ &\quad | \quad \lambda \\ \text{NUMEROS} &\rightarrow \text{num} \text{ NUMEROS} \\ &\quad | \quad \lambda \end{aligned}$$

- a) $\text{PRIMERO}(\text{FILAS}) = \{\text{num}, \lambda\}$, $\text{SIGUIENTE}(\text{FILAS}) = \{), \$\}$ y $\text{predict}(4) = \{), \$\}$
- b) $\text{PRIMERO}(\text{FILAS}) = \{\text{num}\}$, $\text{SIGUIENTE}(\text{FILAS}) = \{\}$ y $\text{predict}(4) = \{\text{num}\}$
- c) $\text{PRIMERO}(\text{FILAS}) = \{\text{num}, \lambda\}$, $\text{SIGUIENTE}(\text{FILAS}) = \{\}$ y $\text{predict}(4) = \{\}$

10. Sea G la gramática con las producciones:

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

que genera el conjunto de paréntesis balanceados. El conjunto I_0 de la colección LR(1) es:

- $\{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot S (S) S, \$] \}$
- $\{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot S (S) S, \$ (]], [S \rightarrow \cdot, \$ (]] \}$
- $\{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot S (S) S, \$ (]], [S \rightarrow \cdot, \$ (]] \}$

11. Si, para la misma gramática, calculamos la colección LR(0) y aparece el siguiente estado:

$$I_j = \{ [S \rightarrow S (S) S \cdot], [S \rightarrow S \cdot (S) S] \}$$

Entonces:

- Se produciría un conflicto deslaza-reduce en la casilla $[j, (]$ de la tabla.
- Se produciría un conflicto deslaza-reduce en la casilla $[j, \$]$ de la tabla.
- Se produciría un conflicto reduce-reduce en la casilla $[j, (]$ de la tabla.

12. Dada la siguiente gramática:

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid -TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid /FT' \mid \lambda$$

$$F \rightarrow (E) \mid \text{num}$$

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

- 1
- 2
- 3

13. La siguiente:

- (1) $MATRIZ \rightarrow (FILA FILAS) \quad \{ FILAS.PrimerColumnas = FILA.Columnas; \}$
- (2) $FILA \rightarrow \text{num NUMEROS} ; \quad \{ FILA.Columnas = NUMEROS.Num + 1; \}$
- (3) $FILAS \rightarrow FILA FILAS_1 \quad \{ \text{if}(FILAS.PrimerColumnas == FILA.Columnas) \\ FILAS_1.PrimerColumnas = FILAS.PrimerColumnas \\ \text{else error}() \}$
- (4) $\mid \lambda$
- (5) $NUMEROS \rightarrow \text{num NUMEROS}_1 \quad \{ NUMEROS.Num = NUMEROS_1.Num + 1; \}$
- (6) $\mid \lambda \quad \{ NUMEROS.Num = 0; \}$

- Es una gramática S-atribuida y L-Atribuida.
- Es una gramática S-atribuida y no L-Atribuida.
- Es una gramática L-Atribuida y no S-Atribuida.

14. La herramienta Bison realiza un análisis:

- SLR con dos tokens de anticipación.
- LALR usando un token de anticipación sólo cuando existen acciones diferentes en un mismo estado de la tabla.
- LR-Canónica con un token de anticipación.

15. La regla

if $f(x)$ es una expresión,
then para cierta α y β , f tiene el tipo $\alpha \rightarrow \beta$ and x tiene el tipo α .

- expresa la *inferencia de tipos* en funciones con un argumento.
- expresa la *síntesis de tipos* en funciones con un argumento.
- convierte el tipo del dominio de una función al de su rango.

Parte II: PROBLEMA. 70 %.

En el sistema formal denominado λ -Cálculo se definen los *números de Church* de la siguiente forma:

$$\begin{aligned}\underline{0} &\equiv \lambda f.(\lambda x.x) \\ \underline{1} &\equiv \lambda f.(\lambda x.(fx)) \\ \underline{2} &\equiv \lambda f.(\lambda x.(f(fx))) \\ \underline{3} &\equiv \lambda f.(\lambda x.(f(f(fx)))) \\ &\dots\end{aligned}$$

Considerar la siguiente gramática G para generar *números de Church*, con $V_T = \{\lambda, ., (,), f, x\}$, $V_N = \{N, F, X, C\}$, N el símbolo inicial, y P el siguiente conjunto de producciones¹:

$$\begin{array}{lll} N & \rightarrow & F . (X . C) \\ F & \rightarrow & \lambda f \\ X & \rightarrow & \lambda x \\ C & \rightarrow & (f C) \\ & | & x \end{array}$$

1. (1.25 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.
2. (0.75 puntos) Simular el algoritmo DPNR usando la tabla anterior, para la cadena de entrada $w \equiv \lambda f(x.x)$, aplicando el método de *recuperación de errores en modo pánico*, en caso de error.
3. (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.
4. (1.5 puntos) Supongamos que queremos construir un traductor simple que traduzca *enteros de Church* en números naturales. Es decir, para la entrada $\lambda f.(\lambda x.(f(fx)))$ la salida debería ser 2. Realizar una definición dirigida por la sintaxis para traducir una entrada generada por la gramática anterior en el valor numérico correspondiente. Definir los atributos que sean necesarios e indicar de forma justificada si la gramática es L-atribuida y/o S-atribuida. Decorar finalmente el árbol sintáctico correspondiente a la entrada $w \equiv \lambda f.(\lambda x.(f(fx)))$.
5. (1 punto) Simular el algoritmo *ascendente predictivo* con la entrada $w \equiv \lambda f.(\lambda x.(f(fx)))$, añadiendo al lado de cada reducción, la acción semántica asociada.

¹Observar que en esta gramática, tanto el símbolo λ como el símbolo $.$ forman parte de los tokens del lenguaje. Por tanto, no debéis confundir en este caso el símbolo λ con la cadena vacía.