

NOMBRE:

PROGRAMACIÓN DECLARATIVA

CURSO 2017-18

PARCIALILLO 1

30-11-2017

- Cada pregunta tiene (espero) una y solo una respuesta correcta. Marcad con un aspa la opción elegida.
- **Cada respuesta correcta suma un punto; cada respuesta incorrecta resta medio punto;** las respuestas en blanco ni suman ni restan. Estad ojo avizor y suerte. Está prohibidísimo copiar.

1. Considérense las expresiones, en las que 1, 2 se suponen de tipo Int:

$[1:[2]]$      $[1,2]:[]$      $1:[2]$      $1:2:[]$

¿Cuál de las siguientes afirmaciones es cierta?

- ☒ Dos son sintácticamente equivalentes entre sí y las otras dos también entre sí
- ☐ Hay tres sintácticamente equivalentes entre sí y una mal tipada
- ☐ Hay dos sintácticamente equivalentes entre sí y una mal tipada

2. Considérense las expresiones de tipo (que solo difieren en los paréntesis):

$\tau_1 = (a \rightarrow a) \rightarrow (a \rightarrow a \rightarrow a)$

$\tau_2 = a \rightarrow a \rightarrow ((a \rightarrow a) \rightarrow a)$

$\tau_3 = a \rightarrow a \rightarrow (a \rightarrow (a \rightarrow a))$

- ☐  $\tau_1 \neq \tau_2 \neq \tau_3 \neq \tau_1$     ☐  $\tau_1 \neq \tau_2 \equiv \tau_3$     ☒ Las dos anteriores son falsas.

3. Considérese el operador infix `4 **` y las expresiones (que solo difieren en los paréntesis):

$e_1 = f \ x \ y \ ** \ g \ x \ ** \ y$

$e_2 = (**) \ ((**) \ ((f \ x) \ y) \ (g \ x)) \ y$

$e_3 = (**) \ (f \ x \ y) \ ((** \ y) \ (g \ x))$

- ☒  $e_1 \equiv e_2 \neq e_3$     ☐  $e_1 \equiv e_3 \neq e_2$     ☐  $e_1 \neq e_2 \neq e_3 \neq e_1$

4. La evaluación de la expresión

`foldr (||) False [False,True,e]` da como resultado

- ☐ False, para alguna **expresión**  $e$  de tipo Bool
- ☒ True, para cualquier **expresión**  $e$  de tipo Bool
- ☐ Las dos anteriores son falsas.

5. La evaluación de la expresión

`foldl (||) False [False,True,e]` da como resultado

- ☐ False, para alguna **expresión**  $e$  de tipo Bool
- ☒ True, para cualquier **expresión**  $e$  de tipo Bool
- ☐ Las dos anteriores son falsas.

6. Sean  $e_1 = [x|x <- l, b]$ ,  $e_2 = [x|x <- l, b, b]$ ,  $e_3 = [x|x <- l, y <- l, b]$ ,

donde  $l :: [\text{Int}]$ ,  $b :: \text{Bool}$  son ciertas expresiones fijas (es decir, las mismas en las tres  $e_i$ ) de los tipos indicados,

y sea  $n_i = \text{length } e_i$  ( $i = 1 \dots 3$ ).

¿Cuál de las siguientes afirmaciones es cierta?

- ☒ Es seguro que  $n_1 = n_2$  y es posible pero no seguro que  $n_1 = n_3$
- ☐  $n_1 = n_2$  y  $n_1 = n_3$  son ambas posibles pero no seguras
- ☐ Puede que  $n_1$  sea par positivo y  $n_3$  sea impar

7. La reducción de la expresión  $(\lambda x \ y \ z \rightarrow y \ x \ (z \ x)) \ 3 \ (\lambda x \ y \rightarrow y \ - \ x) \ (\lambda x \rightarrow x+2)$  producirá el resultado

- ☒ 2    ☐ 1    ☐ Ninguno, porque hay un error de tipos

8. Sea  $f$  definida por  $f \ x \ y \ z = y \ x \ (z \ x)$ . El tipo de  $f$  es:

- ☒  $a \rightarrow (a \rightarrow (a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow c)$
- ☐  $a \rightarrow (a \rightarrow (a \rightarrow b) \rightarrow (a \rightarrow b) \rightarrow b)$
- ☐  $a \rightarrow (a \rightarrow b) \rightarrow (a \rightarrow c) \rightarrow c$

9. Sea  $f\ x\ y = x*x\ y$  y consideremos la expresión  $e \equiv f\ (2+1)\ (1+1)$

- ☒ La evaluación perezosa de  $e$  (teniendo en cuenta la *compartición*) requiere menos pasos que la evaluación impaciente
- ☐ La evaluación perezosa de  $e$  requiere los mismos pasos que la evaluación impaciente
- ☐ Las dos anteriores son falsas.
- 

10. Sea  $f$  definida por las siguientes ecuaciones:

$$\begin{array}{lcl} f\ x\ \text{True} & = & \text{not}\ x \\ f\ x\ y & = & x \end{array}$$

- ☒ La función es estricta en sus dos argumentos
- ☐ La función es estricta en el segundo pero no en el primer argumento
- ☐ Las dos anteriores son falsas.
- 

11. ¿Cuál de las siguientes funciones  $f$  hace que la expresión `zipWith f [(+ n) | n <- [1..]] [[1..n] | n <- [1..]]` esté bien tipada?

- ☐  $f \equiv \text{zip}$       ☒  $f \equiv \text{map}$       ☐  $f \equiv \text{iterate}$
- 

12. ¿Qué podemos afirmar de la evaluación de `takeWhile (> n) $ iterate (+ 1) m`, siendo  $n$  y  $m$  dos números concretos de tipo `Int`?

- ☐ La evaluación terminará, con independencia de  $n$  y  $m$
- ☐ La evaluación no terminará, con independencia de  $n$  y  $m$
- ☒ Las dos anteriores son falsas.
- 

13. Considérense las expresiones siguientes:

$$\begin{array}{ll} e_1 \equiv \backslash x \rightarrow ((\backslash y \rightarrow x+y)\ y) & e_2 \equiv \backslash x \rightarrow ((\backslash y \rightarrow x+y)\ x) \\ e_3 \equiv \text{let } x = y!!1 \text{ in let } y = [1,2,3] \text{ in } x * \text{head } y & e_4 \equiv \text{let } \{x = y!!1 ; y = [1,2,3]\} \text{ in } x * \text{head } y \\ e_5 \equiv [i+j \mid j <- [0..10], i <- [1..j], \text{mod } j\ i == 0] \text{ !! } i & \end{array}$$

¿Cuántas de ellas son sintácticamente erróneas por problemas de ámbito de variables?

- ☐ Exactamente una de ellas      ☒ Tres o más de ellas      ☐ Las dos anteriores son falsas.
- 

14. La evaluación de la expresión `let {y= 1:x ; x=y++[2]} in (x++y)!!2` produce como resultado:

- ☒ 1
- ☐ 2
- ☐ Un cómputo no terminante
- 

15. ¿Cuántas de las siguientes definiciones de tipos (independientes unas de otras) son correctas?

- `data Tip = A | B Int Tip | C (Int,True,Tip)`
- `data Tap = A | (Int,Bool) | C (Int,Int,Tap)`
- `data Top a b = A | C a | D a b`
- ☒ Una de las tres      ☐ Dos de las tres      ☐ Ninguna de las tres
- 

16. El tipo que inferirá Haskell, teniendo en cuenta clases de tipos, para una función  $f$  definida por

- $f\ x\ y\ z = \text{if } x \ \&\& \ y \text{ then } x == y \text{ else } z == 0$  será:
- ☒  $f :: (\text{Num } a, \text{Eq } a) \Rightarrow \text{Bool} \rightarrow \text{Bool} \rightarrow a \rightarrow \text{Bool}$
- ☐  $f :: \text{Eq } a \Rightarrow \text{Bool} \rightarrow \text{Bool} \rightarrow \text{Num } a \rightarrow \text{Bool}$
- ☐ La función  $f$  está mal tipada
-