

PROGRAMACIÓN DECLARATIVA

PRÁCTICA FINAL Curso 2020/2021

Objetivo de la práctica: Escribir un programa en Haskell que implemente una serie de funciones para trabajar con relaciones binarias sobre un conjunto cualquiera.

Descripción del problema: Una relación binaria R sobre un conjunto soporte A puede definirse como un subconjunto de $A \times A$. Las siguientes son ejemplos de relaciones sobre distintos conjuntos:

$$\begin{aligned} R_1 &= \{('D','8'), ('E','B'), ('C','B'), ('E','C'), ('8','D')\} \\ R_2 &= \{([1,2], []), ([2,-2], [3,3,1]), ([1,3], [0]), ([4],[4])\} \end{aligned}$$

Se pide:

1. Define la siguiente estructura de datos para representar en Haskell este tipo de relaciones:

```
data Rel a = R [(a,a)]
```


Define una función `esRelacion r` que vale `True` si `r` es una relación, es decir, la lista que representa el conjunto de pares no tiene repetidos; `False` en caso contrario.
2. El tipo `Rel` debe ser instancia de las clases `Read` y `Show`. También tiene que ser instancia de la clase `Eq`, en este caso hay que redefinir los métodos de la clase para que la igualdad en `Rel` coincida con la igualdad conjuntista.
3. Programa en Haskell las siguientes funciones para trabajar con expresiones del tipo `Rel a`. Puedes dar por hecho que los argumentos de las funciones son realmente funciones, o mejor aún, puedes hacer una comprobación previa. No olvides declarar el tipo de todas las funciones que definas, incluidos los de las funciones locales:

- `dominio r` = conjunto dominio de la relación `r`.
- `soporte r` = conjunto sobre el que está definida la relación `r`.
- `relEquivalencia r` = `True` si `r` es una relación de equivalencia; `False` en caso contrario.
- `conjCociente r` = conjunto cociente de la relación `r`, si esta es relación de equivalencia. Escribe un mensaje de error, en caso contrario.
- `generaDiv n m = r` donde `r` es la relación $\{(x,y) \mid n \leq x, y \leq m, x \text{ es divisor de } y\}$.
- `generaGE xs = r` donde `r` es la relación \geq sobre el conjunto de elementos de la lista `xs`.
- `cierreRST r` = cierre reflexivo, simétrico y transitivo de la relación `r`.
- `composicion r1 r2 = r1 o r2`. Observa que las relaciones `r1` y `r2` tienen que estar definidas sobre el mismo conjunto.

4. Implementa las siguientes acciones de E/S, declarando tipos adecuados:

- `introRel` lee una relación introducida por el usuario. Debe pedir al usuario que introduzca los pares de la relación uno a uno.
- `muestraRel` invoca a la función `introRel` y muestra la relación introducida en forma de tabla. La relación R_1 definida al principio se podría mostrar en forma de tabla, por ejemplo como:

	8	B	C	D	E
8				x	
B					
C				x	
D		x			
E			x	x	

Indicaciones:

- Observa que en muchas expresiones se van a utilizar listas para representar conjuntos. Asegúrate de no introducir duplicados.
- Se pueden usar todas las funciones de **Prelude**, es decir, las que se cargan con el sistema, pero no se puede importar ningún otro módulo, lo que quiere decir que, por ejemplo, si se usan operaciones con listas que no están en **Prelude** hay que programarlas.
- Se valorará muy positivamente el uso de funciones de orden superior predefinidas en Haskell y listas intensionales.
- Para poder comprobar el funcionamiento de cada función implementada y evaluar la práctica, deben incluirse varias relaciones concretas definidas mediante constantes (**r1,r2,r3,r4...**) del tipo definido para representar relaciones, sobre diferentes conjuntos soporte. Al menos una debe ser de equivalencia, otra no y dos deben ser iguales sin estar definidas de manera idéntica.

Entrega de la práctica y calificación:

- La entrega se realizará a través del Campus Virtual y consistirá en un solo fichero .hs no comprimido, en el que las explicaciones irán como comentarios Haskell. El nombre del autor debe aparecer como comentario en la primera línea.
- Fecha límite para la entrega: **7 de febrero de 2021**.
- La nota de la práctica supone el **10 % de la nota final (1 punto)**.

El trabajo es INDIVIDUAL. La copia de otros compañeros o de cualquier otra fuente, así como facilitar la copia a otros, será severamente castigado en la calificación global de la asignatura.
