

TECNOLOGÍA DE PROGRAMACIÓN

Guión de la práctica 6

1.- Objetivo de la práctica

Los objetivos de esta práctica son:

- Comprensión de definición y manejo de estructuras de datos en Haskell.
- Introducción a los mecanismos de herencia y programación genérica de Haskell.

2.- Árboles binarios

En esta práctica, vas a programar en Haskell una serie de funciones que trabajan sobre **árboles binarios de búsqueda**:

http://es.wikipedia.org/wiki/%C3%81rbol_binario

Deberás crear una estructura de datos que represente un árbol binario genérico (es decir, puede ser de enteros, de reales, de cadenas de texto o incluso de cualquier tipo de datos definido por el usuario). **Se deberán implementar las siguientes funciones:**

- `empty` – Esta función devolverá un árbol vacío, sin ningún elemento.
- `leaf x` – Esta función devolverá un árbol que consista en una única hoja que contenga el elemento `x`.
- `tree x lb rb` – Esta función devolverá un árbol que contenga en la raíz el elemento `x`, con hijo izquierdo `lb` e hijo derecho `rb`. Tanto el hijo izquierdo como el hijo derecho son nodos del árbol, no elementos del tipo de dato genérico.
- `size t` – Esta función devuelve el número de elementos del árbol.

También deberás permitir que los árboles puedan ser imprimidos por pantalla (con la alineación y formato que estimes oportuno) utilizando la función de Haskell **print**. Para ello deberás hacer que tu árbol binario herede de la clase **Show** e implemente la función **show**.

Deberás crear un módulo: `BinaryTree.hs`. Para probar tu módulo, puedes o bien utilizar el intérprete de Haskell **ghci** o bien hacer un programa principal. Prueba que el árbol es realmente genérico (es decir: prueba el árbol con diferentes tipos de datos).

3.- Recorrido de árboles

Hay tres tipos de recorrido que se pueden hacer sobre árboles binarios: en pre-orden, post-orden e in-orden (http://es.wikipedia.org/wiki/%C3%81rbol_binario)

Crea tres funciones (`preorder t`, `postorder t`, `inorder t`) que, dado un árbol `t`, devuelva una lista con todos los elementos del árbol en el orden correspondiente. Prueba que las tres funciones funcionan correctamente.

4.- Árboles de búsqueda

Deberás añadir nuevas funciones a tu árbol para añadir elementos a un árbol, de tal forma que se inserten ordenadamente (cada nodo siempre será mayor que todos los elementos del subárbol de la izquierda y menor que todos los elementos del subárbol de la derecha).

Para ello deberás implementar las siguientes dos funciones:

- `add t x` – Añade el elemento x al árbol t , devolviendo el resultado.
- `between t xmin xmax` – Busca en el árbol t y devuelve una lista con todos los elementos que están entre $xmin$ y $xmax$ (inclusive). No deberá recorrerse todos los elementos del árbol, ni devolver la lista (necesariamente) ordenada.

Date cuenta que para poder implementar estas funciones, necesitarás asegurar que el tipo de dato de x (dado que los árboles son genéricos) tenga los operadores de comparación. Esto se consigue añadiendo una restricción para que x sea instancia de un tipo **Ord**.

5.- Pruebas

Te proporcionamos un fichero `BinaryTreeTest.hs` que prueba todas las funciones. Además de las pruebas que tú mismo hagas con el intérprete de Haskell, tus funciones deberán compilar con dicho archivo de pruebas. De lo contrario la calificación de la práctica será de 0.

6.- Para entregar

Como resultado de esta práctica deberás entregar el siguiente archivo:

- `BinaryTree.hs` – que contenga la estructura de datos junto con todas las funciones arriba mencionadas. Todas las funciones deben tener **exactamente el mismo nombre** expresado en este guión.

Deberás incluir los archivos que te pedimos en un archivo comprimido con el siguiente nombre:

- `practica6_<NIP>.zip` si la práctica se realiza de forma individual (donde `<NIP>` representa los 6 dígitos del NIP del alumno).
- `practica6_<NIP1>_<NIP2>.zip` si la práctica se realiza por parejas (donde `<NIP1>` y `<NIP2>` representa los 6 dígitos del NIP de cada alumno de la pareja, respectivamente).

El código deberá compilar con el archivo de pruebas que te proporcionamos. De no hacerlo la calificación de la práctica será de 0.