

Paradigmas de Programación

Práctica 9

1. Considere la siguiente definición de tipo para representar árboles generales, y considere también la función que, dado un árbol de este tipo, devuelve la lista de nodos resultante de efectuar un recorrido por niveles sobre dicho árbol:

```
type 'a g_tree = Gt of 'a * 'a g_tree list;;

let rec breadth_first = function
  Gt (x, []) -> [x]
| Gt (x, (Gt (y, t2))::t1) -> x :: breadth_first (Gt (y, t1@t2));;
```

Se pide lo siguiente:

- Defina con nombre `breadth_first_t` una versión terminal de `breadth_first`.
- Defina un valor `t : int gtree` tal que no sea posible calcular `breadth_first t`, pero sí sea posible calcular `breadth_first_t t`.

2. Considere la siguiente definición de tipo para representar árboles binarios:

```
type 'a bin_tree = Empty | Node of 'a * 'a bin_tree * 'a bin_tree;;
```

Se pide lo siguiente:

- Defina una función `breadth_first : 'a bin_tree -> 'a list` que calcule el recorrido por niveles de un árbol binario.

3. (Ejercicio opcional) Defina las funciones `strict`, `perfect` y `complete`, todas ellas de tipo `'a bin_tree -> bool`, de forma que tengan el siguiente comportamiento:

- Un árbol binario es estricto si de cada nodo cuelgan dos ramas no vacías, o bien dos ramas vacías.
- Un árbol binario es perfecto si es estricto y si todas sus hojas están en el mismo nivel.
- Un árbol binario es completo si todo nivel está lleno (excepto quizás el último), y si todos los nodos del último nivel están lo más a la izquierda posible.

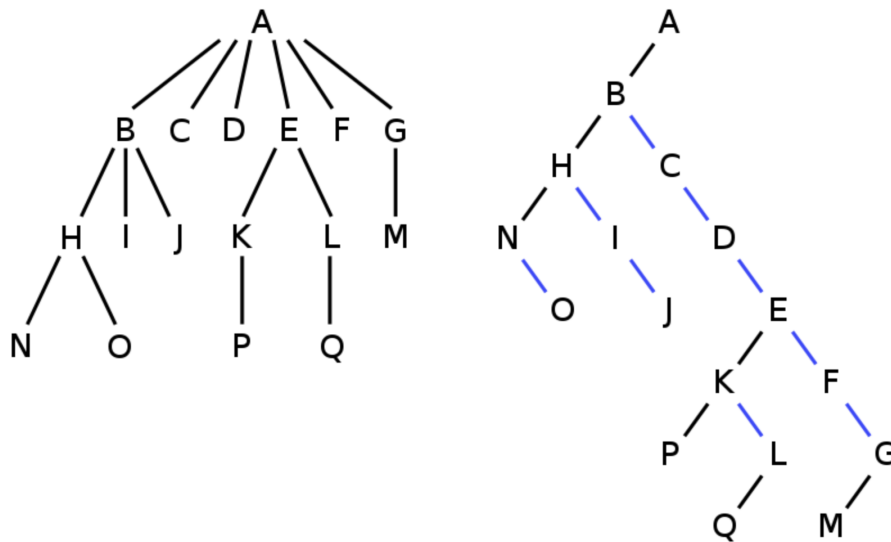
4. (Ejercicio opcional) Cualquier árbol general puede ser codificado como un árbol binario, utilizando la siguiente idea, tomada de este [enlace](#) de la Wikipedia:

***Codificación de árboles n-arios como árboles binarios.** Hay un mapeo uno a uno entre los árboles generales y los árboles binarios, el cual en particular es usado en Lisp para representar árboles generales como árboles binarios. Cada nodo N ordenado en el árbol corresponde a un nodo N' en el árbol binario; el hijo de la izquierda de N es el nodo correspondiente al primer hijo de N , y el hijo derecho de N' es el nodo correspondiente al siguiente hermano de N , es decir, el próximo nodo en orden entre los hijos de los padres de N .*

Esta representación como árbol binario de un árbol general se conoce a veces como un árbol binario primer hijo hermano, o como árbol doblemente encadenado.

Una manera de pensar acerca de esto es que los hijos de cada nodo estén en una lista enlazada, encadenados junto con el campo derecho, y el nodo solo tiene un puntero al comienzo o la cabeza de esta lista, a través de su campo izquierdo.

Por ejemplo, en el árbol de la izquierda, el nodo *A* tiene 6 hijos (*B*, *C*, *D*, *E*, *F* y *G*), y puede ser convertido en el árbol binario de la derecha.



El árbol binario puede ser pensado como el árbol original inclinado hacia los lados, con los bordes negros izquierdos representando el primer hijo y los azules representando los siguientes hermanos.

Se pide implementar las dos siguientes funciones:

- `codas_bin : 'a g_tree -> 'a bin_tree`
- `decod_from_bin : 'a bin_tree -> 'a g_tree`

de forma que sirvan para codificar y decodificar árboles generales como binarios.

NOTA: Realice las implementaciones de los ejercicios 1, 2, 3 y 4 en los ficheros `gtree.ml`, `bintree.ml`, `ej93.ml` y `ej94.ml`, respectivamente.

NOTA: Las implementaciones del fichero `ej93.ml` deben trabajar sobre el tipo `'a bin_tree`, el cual está definido en otro fichero, más concretamente, en `bintree.ml`. No repita la definición del tipo en el fichero `ej93.ml`. Lo que debe hacer es calificar los constructores del tipo (es decir, escriba `Bintree.Empty` y `Bintree.Node`), o bien escriba `open Bintree` al principio del fichero.

NOTA: Lo mismo ocurre en `ej94.ml` con el tipo `'a bin_tree` definido en `bintree.ml`, y también con el tipo `'a g_tree` definido en `gtree.ml`. De igual modo, no repita la definición de los tipos en `ej94.ml`. Califique los constructores (es decir, escriba `Bintree.Empty`, `Bintree.Node` y `Gtree.Gt`), o bien escriba `open Bintree` y `open Gtree` al principio del fichero.