

Programmation fonctionnelle - TD5

Licence 3

Stefano Guerrini

12 mars 2010

On considère le type suivant pour les arbres binaires étiquetés sur les nœuds :

```
type 'a bintree = Leaf 'a | Node of 'a * 'a bintree * 'a bintree
```

Écrire les suivantes fonctions.

1. `height : 'a bintree -> int`
Qui calcule la profondeur d'un arbre binaire.
2. `addlabels : int bintree -> int`
Qui calcule la somme des étiquettes d'un arbre binaire.
3. `preorder : 'a bintree -> 'a list`
Qui retourne la liste des étiquettes de tous les nœuds d'un arbre binaire en ordre préfixe (ou préordre).
4. `postorder : 'a bintree -> 'a list`
Qui retourne la liste des étiquettes de tous les nœuds d'un arbre binaire en ordre postfixe (ou postordre).
5. `inorder : 'a bintree -> 'a list`
Qui retourne la liste des étiquettes de tous les nœuds d'un arbre binaire en ordre infixé (ou inordre).
6. `list_of_leaves : 'a bintree -> 'a list`
Qui retourne la liste des feuilles d'un arbre binaire (de gauche à droite).
7. `list_of_nodes_depth : int * 'a bintree -> 'a list`
Qui retourne la liste des feuilles d'un arbre binaire à une profondeur donnée
8. `subtrees`
Qui calcule la liste des sous-arbres d'un arbre binaire.
9. `filter_bintree : ('a -> bool) -> 'a bintree -> 'a list`
Qui construit la liste en ordre préfixe des étiquettes d'un arbre binaire vérifiant un prédicat donné.
10. `map_bintree : ('a -> 'b) -> 'a bintree -> 'b bintree`
Qui applique une fonction à toutes les étiquettes d'un arbre binaire.
11. Un arbre binaire est dit saturé si toutes les feuilles sont à la même profondeur. Définir le prédicat `sature : 'a bintree -> bool` dont la valeur est `true` si et seulement si un arbre binaire donné est saturé. (Suggestion : utiliser une fonction qui retourne -1 si un arbre n'est pas saturé et $k \geq 0$ si l'arbre est saturé de profondeur k).