Lycée Chateaubriand
Informatique MP 2002/03
Mathieu Giraud

TP Caml n° 2
Arbres

Nous travaillons aujourd'hui sur les arbres, et plus particulièrement sur les arbres binaires de recherche. Vous pouvez récupérer le fichier `TP2.ml` qui contient quelques arbres pour tester vos programmes.

L'énoncé de ce TP est disponible à l'adresse `http://magiraud.free.fr/tpcaml`

1 Opérations de base sur les arbres binaires

On utilise ici un type d'arbre binaire comportant des valeurs seulement aux feuilles.

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

a. Écrire (toujours en donnant leur complexité...) les fonctions `nb_noeuds a`, `nb_feuilles a` et `hauteur a` calculant le nombre de noeuds, de feuilles et la hauteur d'un arbre.

b. Écrire la fonction `miroir` renvoyant l'image miroir d'un arbre.

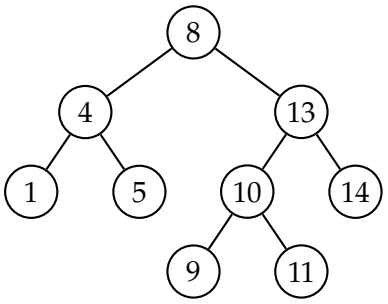
c. Dans le cas d'un arbre d'entiers, écrire la fonction `somme` renvoyant la somme de toutes les valeurs des feuilles.

2 Arbres binaires de recherche

Un arbre binaire de recherche (ABR) est un arbre binaire dont tous les noeuds sont étiquetés et tel que la valeur de chaque noeud est
- supérieure à toutes les valeurs apparaissant dans le fils gauche
- et inférieure à toutes les valeurs apparaissant dans le fils droit.

Le type de ces arbres sera ici

```
type arbre_br = Vide | Noeud of arbre_br * int * arbre_br
```




```
Noeud(Noeud(Noeud(Vide, 1, Vide),
            4,
            Noeud(Vide, 5, Vide)),
      8,
      Noeud(Noeud(Noeud(Vide, 9, Vide),
                  10,
                  Noeud(Vide, 11, Vide)),
            13,
            Noeud(Vide, 14, Vide)))
```

d. Écrire une fonction `recherche : int -> arbre_br -> bool` recherchant la présence d'un entier dans un ABR.

e. Écrire la fonction `insere : int -> arbre_br -> arbre_br` qui insère un entier dans un ABR.

f.* Écrire une fonction `retire : int -> arbre_br -> arbre_br` qui retire un entier d'un ABR (et qui renvoie une erreur si l'entier n'est pas présent). Attention, la solution est assez longue.

g. Comme d'habitude, vous avez donné la complexité "dans le cas le pire" de vos fonctions... mais quelle est leur complexité "en moyenne" ? D'où vient la différence ? Quels sont les pires cas et que peut-on faire pour améliorer cette situation ?

3 ABR et listes

h. Écrire une fonction `liste_of_arbre : arbre_br -> int list` qui crée une liste triée à partir d'un ABR. Cette fonction utilise un parcours infixe de l'arbre.

i. Écrire une fonction `arbre_of_liste : int list -> arbre_br` qui crée un ABR à partir d'une liste (non nécessairement triée). Que vient-on de réaliser ?

4 S'il vous reste du temps...

j.* Faire les dernières questions du premier TP (addition et multiplication de polynômes, traitement des entiers longs).

k.* Comment traiter des arbres n-aires ?

Sources

Ce TP a été inspiré par celui de l'année dernière ainsi que par un TP de Jean-Christophe Filliâtre (`http://caml.inria.fr/polycopies/index-fra.html`). N'hésitez pas à consulter cette adresse qui pointe vers de nombreux sujets intéressants.