

## TP D'OPTION INFORMATIQUE 4

### Arbres binaires de recherche

On rappelle qu'un **arbre binaire de recherche** (ABR) est un arbre pour lequel des clés sont associées aux noeuds, et vérifiant que pour chaque noeud  $x$ , les clés du fils gauche de  $x$  sont strictement inférieures à la clé de  $x$ , et les clés du fils droit de  $x$  sont strictement supérieures à la clé de  $x$ .

On utilise les types suivants :

```
type ('a,'b) arbre =  
  V  
| N of ('a,'b) arbre * 'a * 'b * ('a,'b) arbre;;
```

```
type ('a,'b) dico = ('a,'b) arbre ref;;
```

pour représenter un dictionnaire mutable dont les clés sont de type 'a et les valeurs associées de type 'b.

## 1 Premières opérations

1. Implémenter les opérations suivantes de la structure de dictionnaire :
  - `creer_dictionnaire` : `unit -> ('a, 'b) dico`
  - `ajouter_couple` : `('a, 'b) dico -> 'a -> 'b -> unit`
  - `cle_presente` : `('a, 'b) dico -> 'a -> bool`
  - `valeur_associee` : `('a, 'b) dico -> 'a -> 'b`
  - `modifier_valeur` : `('a, 'b) dico -> 'a -> 'b -> unit`
2. Déterminer la complexité de ces opérations ainsi implémentées.
3. Écrire une fonction `parcours_abr` : `('a -> 'b -> unit) -> ('a, 'b) arbre -> unit` prenant en entrée une procédure et un arbre binaire de recherche, et appliquant la procédure sur chaque clé et valeur de l'arbre, dans l'ordre croissant des clés.  
Par exemple, `parcours_abr (fun c v -> print_int c) a` doit afficher les clés entières de l'arbre `a` dans l'ordre croissant.

## 2 Suppression d'une clé

On souhaite à présent implémenter l'opération de suppression d'une clé et de sa valeur du dictionnaire. Pour être raisonnablement efficace et pour préserver la structure générale de l'arbre, on procède en remplaçant la clé à supprimer par la clé maximale dans son fils gauche.

1. Au papier, illustrer cette méthode sur un exemple.
2. Comment procéder si la clé à supprimer a un fils droit sans avoir de fils gauche ?
3. Écrire une fonction `('a, 'b) arbre -> ('a * 'b) * ('a, 'b) arbre` prenant en entrée un ABR `a` et renvoyant  $(c, v), a'$ , où  $c$  est la clé maximale de  $a$ ,  $v$  sa valeur associée, et  $a'$  est l'ABR obtenu à partir de  $a$  en y supprimant  $(c, v)$ .
4. En déduire la fonction `supprimer` : `('a, 'b) dico -> 'a -> unit` implémentant l'opération de suppression d'une clé et de sa valeur associée.
5. Déterminer la complexité de cette opération.

## 3 Test de recherche

1. Écrire une fonction prenant en argument un arbre binaire et testant si c'est un ABR.\*
2. Déterminer la complexité de cette fonction. Est-elle optimale ?