ARBRES BINAIRES DE RECHERCHE

Dans ce chapitre, il est sous-entendu que les arbres sont binaires.

1 Généralités

1.1 Quelques notations utilisées dans ce chapitre

Soit B un arbre sur E. Si s est un sommet, on note :
- clef(s) l'étiquette du sommet s,
- Fg(s) et Fd(s) les fils gauche et droit de s (éventuellement vides),
- $S(B)$ l'ensemble des sommets de B,
- $N(B)$ l'ensemble des nœuds (internes) de B.

1.2 Définition

Définition. Soit E un ensemble ordonné.

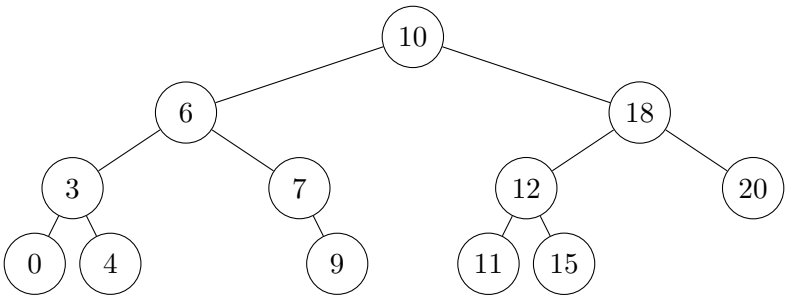
Un arbre binaire de recherche sur E (en abrégé dans ce cours ABR) est un arbre sur E tel que l'étiquette de tout nœud (interne) est strictement supérieur à l'étiquette de tout sommet de son fils gauche et inférieure ou égale à celle de tout sommet de son fils droit. Dans le cas des ABR, les étiquettes sont aussi appelées les clefs.

Soit B un arbre sur E.

B est un ABR si et seulement si pour tout $s \in N(B)$, pour tout $s' \in S(\text{Fg}(s))$ et $s'' \in S(\text{Fd}(s))$, on a

$$\text{clef}(s') < \text{clef}(s) \leqslant \text{clef}(s'')$$

Exemple.



Proposition 1 *Un arbre est un ABR si et seulement si la suite des clefs obtenues par le parcours infixe de l'arbre est croissante.*

2 Opérations sur les ABR

2.1 Représentation CAML

On représente les arbres par le type générique

```
type 'a arbre = V | S of 'a * 'a arbre * 'a arbre;;
```

On code les opérations suivantes sur les ABR (on supposera que les clefs sont distinctes) :
- recherche dans un ABR (un objet appartient-il à un ABR ?) ;
- insertion dans un ABR (sans fioritures) ;
- création d'un ABR à partir d'une liste d'objets ;
- vérifier si un arbre est un ABR ;

Dans tous les cas, en notant h la hauteur de l'ABR, la complexité de ces opérations sur les ABR est en $O($ $)$.

2.2 Structure persistante de dictionnaire

Un dictionnaire est un type abstrait qui permet d'organiser des couples (clef, valeur) (penser à un dictionnaire papier : les clefs sont les mots, les valeurs sont les définitions des mots) et de permettre de récupérer, si possible rapidement, la valeur associée à un clef lors d'une recherche.

Un dictionnaire peut être concrètement réalisé de diverses manières :

- par une simple liste (ou tableau) de couples (clef, valeur) : recherche peu efficace en complexité $O(\ )$ où n est le nombre de couples ;
- par un tableau de couples (clef, valeur) **triée sur les clefs** : la recherche peut être faite en complexité $O(\ \ \ \ )$;
- par un ABR dont les étiquettes sont les couples et ordonné par les clefs : à condition que l'arbre soit « bien équilibré », la recherche peut être faite en complexité $O(\ \ \ \ \ \ )$, sinon dans le pire cas, elle est de complexité $O(\ \ \ \ \ \ )$;

Si on s'en tient à l'opération de recherche, le tableau trié semble être ce qu'il y a de mieux. Mais si on ajoute des contraintes sur la suppression ou l'insertion de couples, alors la structure d'ABR est plus intéressante, à condition de produire un ABR équilibré (voir TD).