

UE ALGO5 — TD2 — Séance 7 : Arbres binaires

Objectifs

À la fin de cette séance, vous devriez être capable de :

- manipuler et concevoir des arbres binaires comme des structures abstraites ;
- réfléchir aux propriétés des arbres binaires ;
- proposer des implémentations d'arbres binaires cohérentes avec les spécifications choisies en utilisant des structures sous-jacentes adaptées.

On donne un exemple de spécification d'un type abstrait `Arbre` construit sur un type `Élément`.

```

1 ArbreVide {
   Données : aucun
3 Résultat : un Arbre vide
   Post-condition : renvoie un Arbre vide
5 Effet de bord : aucun
}
7 NouveauNœud(G,x,D) {
9 Données : un Arbre G, un Élément x, un Arbre D
  Résultat : un Arbre constitué du nœud (G, x, D)
11 Effet de bord : un nouveau nœud a été créé
}
13 EstArbreVide(A) {
15 Donnée : un Arbre A
  Résultat : un booléen vrai ssi A est un Arbre vide
17 }
19 Racine(A) {
  Donnée : un Arbre A
21 Résultat : l'Élément associé à la racine de A
  Pré-condition : A non vide
23 }
25 FilsDroit(A) {
  Donnée : un Arbre A
27 Résultat : un Arbre, renvoie le fils droit associé à la
  racine de A
  Pré-condition : A non vide
29 }
31 FilsGauche(A) {
  Donnée : un Arbre A
33 Résultat : un Arbre, renvoie le fils gauche associé à
  la racine de A
  Pré-condition : A non vide
35 }
37 Libérer(A) {
  Donnée : un Arbre A
39 Pré-condition : A non vide
  Post-condition : la mémoire associée au nœud racine
  de A est libérée
41 Effet de bord : La racine de A est détruite, les fils
  gauche et droit sont inchangés
}
43 InsérerGauche(A,G) {
45 Donnée-résultat : un Arbre A
  Donnée : un Arbre G
47 Pré-condition : A non vide
  Post-condition : le fils gauche de A est remplacé par
  l'Arbre G.
49 Effet de bord : l'Arbre A est modifié, son ancien fils
  gauche est «détruit»
}
51 InsérerDroit(A,D) {
53 Donnée-résultat : un Arbre A
  Donnée : un Arbre D
55 Pré-condition : A non vide
  Post-condition : le fils droit de A est remplacé par
  l'Arbre D.
57 Effet de bord : l'Arbre A est modifié, son ancien fils
  droit est «détruit»
}

```

Exercice 1. Utilisation du type abstrait `Arbre`

Q 1. Dessiner un arbre binaire de hauteur 3 comportant 5 feuilles

En n'utilisant que des primitives du type abstrait :

Q 2. Écrivez une séquence d'instructions permettant de construire l'arbre binaire proposé à la question 1.

Q 3. Écrivez une fonction qui prend en paramètre un arbre binaire et renvoie le nombre de feuilles de cet arbre.

- Q 4. Écrivez une fonction qui prend en paramètre un arbre binaire et renvoie la hauteur de cet arbre.
- Q 5. Écrivez une procédure permettant de «supprimer» un arbre entier, en libérant la mémoire de chacun de ses nœuds.

Comment procéderiez-vous sans récursivité?

Exercice 2. Implémentation du type abstrait **Arbre**

- Q 1. Proposez une implémentation du type Arbre à l'aide d'un tableau de taille N alloué statiquement (on suppose donc ici que le nombre de nœuds de tout arbre binaire sera limité à N). Écrivez le code de chaque primitive pour cette implémentation.
- Q 2. Proposez maintenant une implémentation du type Arbre à l'aide de cellules mémoires allouées dynamiquement et chaînées par pointeurs.