UFR IM²AG

UNIVERSITÉ Grenoble Alpes

L3 Informatique
Année 2018 – 2019

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.

ArbreVide {
Données : aucun
Résultat : un Arbre vide
Post-condition : renvoie un Arbre vide
Effet de bord : aucun
}

NouveauNœud(G,x,D) {
Données : un Arbre G, un Élément x, un Arbre D
Résultat : un Arbre constitué du nœud (G, x, D)
Effet de bord : un nouveau nœud a été créé
}

EstArbreVide(A) {
Donnée : un Arbre A
Résultat : un booléen vrai ssi A est un Arbre vide
}

Racine(A) {
Donnée : un Arbre A
Résultat : l'Élément associé à la racine de A
Pré-condition : A non vide
}

FilsDroit(A) {
Donnée : un Arbre A
Résultat : un Arbre, renvoie le fils droit associé à la racine de A
Pré-condition : A non vide
}

FilsGauche(A) {
Donnée : un Arbre A
Résultat : un Arbre, renvoie le fils gauche associé à la racine de A
Pré-condition : A non vide
}

Libérer(A) {
Donnée : un Arbre A
Pré-condition : A non vide
Post-condition : la mémoire associée au nœud racine de A est libérée
Effet de bord : La racine de A est détruite, les fils gauche et droit sont inchangés
}

InsérerGauche(A,G) {
Donnée-résultat : un Arbre A
Donnée : un Arbre G
Pré-condition : A non vide
Post-condition : le fils gauche de A est remplacé par l'Arbre G.
Effet de bord : l'Arbre A est modifié, son ancien fils gauche est «détruit»
}

InsérerDroit(A,D) {
Donnée-résultat : un Arbre A
Donnée : un Arbre D
Pré-condition : A non vide
Post-condition : le fils droit de A est remplacé par l'Arbre D.
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.

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.