

UE ALGO5 — TD2 — Séance 8 : Arbres n-aires

Objectifs

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

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

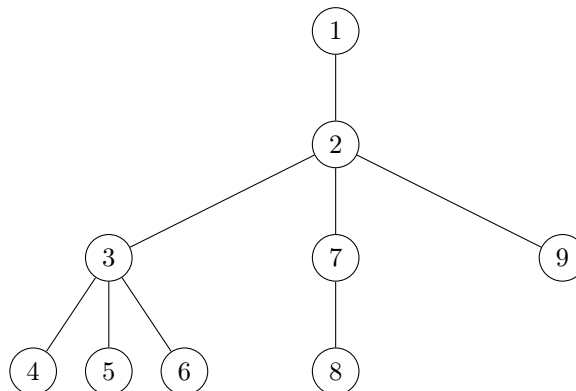
On s'intéresse ici à l'implémentation d'un type abstrait «Arbre n-aire», en utilisant le type «Arbre binaire» vu lors de la séance précédente.

Un arbre n-aire est ici implémenté comme suit :

- le «fils aîné» d'un nœud n-aire est implémenté par le «fils gauche» du nœud binaire du père ;
- le «frère cadet» d'un nœud n-aire est implémenté par le «fils droit» du nœud binaire du frère précédent.

Exercice 1.

Q1. Traduisez l'arbre ci-dessous dans sa représentation binaire :



Exercice 2.

Q2. Implémentez les primitives du type abstrait «arbre n-aire», telles que définies ci-dessous.

- 1 Élément : un **type**
- Arbre : un **type**
- 3 ListeArbre : un **type**

- 5 ArbreVide
- { **Données** : *aucun*
- 7 **Résultat** : *un Arbre vide*}

- 9 NouveauNœud
- { **Données** : *un Élément x, une ListeArbre L*
- 11 **Résultat** : *un Arbre constitué du nœud x, dont les fils sont les éléments de L*
- Effet de bord** : *un nouveau nœud a été créé* }
- 13
- EstArbreVide
- 15 { **Données** : *un Arbre A*
- Résultat** : *un booléen vrai ssi A est un Arbre vide* }

17

Elem

19 { **Données** : un Arbre A

Résultat : l'Élément associé à la racine de A

21 **Pré-condition** : A est non vide }

23 ListeFils

{ **Données** : un Arbre A

25 **Résultat** : une ListeArbre

description : A doit être non vide, renvoie la liste des fils associée à la racine de A }

27

{ Manipulation des listes d'arbres }

29

ListeVide

31 { **Données** : aucun

Résultat : une ListeArbre vide }

33

Cons

35 { **Données** : un Arbre A, une ListeArbre L

Résultat : une ListeArbre constituée de l'arbre A, suivi de la liste L. }

37

EstListeVide

39 { **Données** : une ListeArbre L

Résultat : un booléen vrai ssi L est vide. }

41

Premier

43 { **Données** : une ListeArbre L

Résultat : un Arbre, renvoie le premier arbre de la liste L

45 **Pré-condition** : L non vide }

47 Suivants

{ **Données** : une ListeArbre L

49 **Résultat** : une ListeArbre, renvoie la liste des arbres suivants le premier.

Pré-condition : L non vide }