

Méthodes de conception de programmes

Devoir 2 : 1, 2, 3... Arbres !

Alexandre GOBEAUX^a, Louis NAVARRE^a, Gilles PEIFFER^a

^a*École Polytechnique, Université catholique de Louvain, Place de l'Université 1, 1348 Ottignies-Louvain-la-Neuve, Belgique*

Abstract

Ce papier donne les invariants de représentation, la fonction d'abstraction et les spécifications des fonctions `insert` et `join` pour une implémentation des arbres 2-3 basée sur `?` et `?`.

1. Invariant de représentation

Commençons par définir quelques fonctions auxiliaires :

- `size(T)` : donne le nombre de nœuds d'un arbre T ;
- `height(T)` : donne la hauteur d'un arbre T ;
- `type(T)` : donne le nombre de sous-arbres du nœud source de l'arbre T .

Afin d'alléger la notation de l'invariant de représentation `ok(T)`, voici quelques fonctions supplémentaires.

- Si le nœud source de T est un 2-nœud, alors L et R dénotent respectivement le sous-arbre de gauche et de droite de T , alors que a dénote la donnée de son nœud source.
- Si le nœud source de T est un 3-nœud, alors L , M et R dénotent respectivement le sous-arbre de gauche, du milieu et de droite de T , alors que $a < b$ sont les données du nœud source.

$$f(T) = \left(\text{size}(L) > 0 \wedge \text{size}(R) > 0 \right) \wedge \left(\text{height}(L) = \text{height}(R) \right) \wedge \left(\forall \lambda \in L, \varrho \in R : \lambda < a \leq \varrho \right), \quad (1)$$

$$g(T) = \left(\text{size}(L) > 0 \wedge \text{size}(M) > 0 \wedge \text{size}(R) > 0 \right) \wedge \left(\text{height}(L) = \text{height}(M) = \text{height}(R) \right) \wedge \left(\forall \lambda \in L, \mu \in M, \varrho \in R : \lambda < a \leq \mu < b \leq \varrho \right). \quad (2)$$

L'invariant de représentation est alors donné par

$$\text{ok}(T) \equiv \text{size}(T) = 0 \vee (\text{type}(T) = 2 \wedge f(T)) \vee (\text{type}(T) = 3 \wedge g(T)). \quad (3)$$

2. Fonction d'abstraction

La fonction d'abstraction `abs()` est donnée par

$$\text{abs}() = . \quad (4)$$

3. Spécifications formelles

3.1. Spécification de `insert`

3.1.1. Précondition

3.1.2. Modifies

3.1.3. Postcondition

3.2. Spécification de `join`

3.2.1. Précondition

3.2.2. Modifies

3.2.3. Postcondition

Références