Démonstrations en informatique théorique

OPTION INFORMATIQUE - TP n° 2.6 - Olivier Reynet

A De l'intérêt des ensembles inductifs

A1. Écrire une fonction de signature `concat : 'a list -> 'a list -> 'a list` qui renvoie la concaténation de deux listes.

A2. Prouver la terminaison de la fonction `concat`.

A3. Prouver la correction de la fonction `concat`

A4. Quelle est la complexité de la fonction `concat` ?

R D'une manière plus générale, si on dispose d'un ensemble inductif X défini de manière non ambigüe et d'une fonction f récursive sur X qui utilise la structure inductive de l'ensemble dans sa définition, alors :

- la terminaison de f est évidente car l'ordre induit sur X est bien fondé,
- la correction de f se montre par induction structurelle,
- la complexité s'analyse de la même manière qu'une fonction récursive quelconque.

B Arbres AVL

Afin de réduire au maximum les complexités des opérations, on cherche à minimiser la hauteur des arbres binaires de recherche utilisés pour les dictionnaires.

■ **Définition 1 — Arbre AVL.** Un arbre binaire de recherche est un arbre AVL[a] si, pour tout nœud, la différence entre la hauteur du sous-arbre gauche et la hauteur du sous-arbre droit est $-1, 0$ ou 1 .

a. du nom de leur deux inventeurs : Adelson-Velsky et Landis

Un AVL garantit (cf. questions ci-dessous) que la hauteur de l'arbre est en $\Theta(\log n)$ où n est le nombre de nœuds.

■ **Définition 2 — Facteur d'équilibrage.** Le facteur d'équilibrage d'un nœud d'un arbre binaire de recherche est la différence entre la hauteur du sous-arbre gauche et la hauteur du sous-arbre droit.

Ainsi, pour un arbre AVL, le facteur d'équilibrage de chaque nœud est -1 , 0 ou 1.

On souhaite montrer que la hauteur d'un arbre AVL est de l'ordre de $\log(n)$ où n est le nombre de nœuds. Pour cela, on note $(f_n)_{n\in\mathbb{N}}$ la suite de Fibonacci définie par :

$$\begin{cases} f_0 = 1, f_1 = 1 \\ f_{n+2} = f_{n+1} + f_n \quad \text{pour tout } n \geqslant 0 \end{cases} \tag{1}$$

B1. Donner un exemple d'AVL pour les facteurs d'équilibrage 0, 1 et -1.

B2. Soit $h \in \{-1\} \cup \mathbb{N}$. Montrer qu'un arbre AVL de hauteur h possède au moins $f_{h+1} - 1$ nœuds.

Soit $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.62$ le nombre d'or. On rappelle que φ est racine du polynôme $X^2 - X - 1$.

B3. Montrer que $f_n \geqslant \varphi^{n-1}$ pour tout $n \in \mathbb{N}$.

B4. On considère un arbre binaire de recherche de hauteur h avec n nœuds. Montrer que si cet arbre est un AVL, alors $h \leqslant \log_\varphi(n+1)$.

B5. La réciproque de la proposition précédente est-elle vraie ?

B6. Montrer que pour tout arbre binaire, $h \geqslant \log_2(n+1) - 1$.

B7. En conclure que la hauteur d'un arbre AVL est en $\Theta(\log n)$.

C Arbres de Braun

Si t est un arbre, on note $|t|$ sa taille.

■ **Définition 3 — Arbre de Braun.** Un arbre de Braun est un arbre binaire qui est :

- soit l'arbre vide.
- soit de la forme $N(r, g, d)$ avec r une étiquette et g et d deux arbres de Braun vérifiant

$$|d| \leq |g| \leq |d| + 1$$

On se limite dans ce sujet sujet au cas où les étiquettes des arbres de Braun sont des entiers et on représente de tels arbres à l'aide du type suivant en OCaml :

```
type braun = Vide | Noeud of int * braun * braun
```

C1. Pour tout $n \in [\![1, 6]\!]$, dessiner les squelettes des arbres de Braun de taille n. Que remarque-t-on ?

C2. Montrer que pour tout entier naturel n, il existe un arbre de Braun unique de taille n.

On dispose de la fonction suivante :

```
let rec add_braun x t =
    match t with
    |Vide -> Noeud(x, Vide, Vide)
    |Noeud(y,g,d) -> Noeud(y,add_braun x d,g);;
```

C3. Montrer que la fonction `add_braun` est correcte, c'est-à-dire qu'elle renvoie un arbre de Braun.