Matthieu Boyer
Planche A

Génération Fonctionnelle

On s'intéresse dans cet exercice à l'utilisation des séries génératrices pour calculer des longueur de chemin moyennes.

1. Rappelez l'algorithme de quicksort dont le pivot est choisi pour être le premier élément du tableau.

On va s'intéresser au nombre moyen de comparaison de quicksort sur tous les arrangements possibles des nombres dans le tableau d'entrée. On pose t_n le nombre moyen de comparaisons de quicksort sur n nombres.

1. Donnez une relation de récurrence sur t_n.
2. En introduisant la série génératrice des t_n, calculez t_n.
3. Déduisez en une fonction simple f telle que $t_n = \Theta\left(f(n)\right)$ i.e. $t = \mathcal{O}(f)$ et $f = \mathcal{O}(t)$.
4. Démontrez d'une manière similaire que la longueur moyenne d'un chemin dans un arbre avec N noeuds internes est $\frac{N}{2}\left(\sqrt{\pi N} - 1\right) + \mathcal{O}\left(\sqrt{N}\right)$.

Solution: Génération Fonctionnelle

Voir `https://www3.cs.stonybrook.edu/~rezaul/Fall-2017/CSE548/CSE548-lecture-7.pdf` et `https://aofa.cs.princeton.edu/60trees/`.

Matthieu Boyer
Planche B

Foncteurs

On définit une (petite) catégorie $\mathcal{C}$ comme une classe d'objets $\mathrm{Obj}\,\mathcal{C}$ et, pour chaque paire (A, B) d'objets un ensemble de flèches $\mathrm{Hom}_{\mathcal{C}}(A, B)$ telles que :

- Si $f : A \to B$ et $g : B \to C$ alors il existe $h \in \mathrm{Hom}_{\mathcal{C}}(A, C)$ et on note $h = g \circ f$.
- Il existe $\mathrm{id}_A \in \mathrm{Hom}_{\mathcal{C}}(A, A)$ telle que $f \circ \mathrm{id}_A = f$ et $\mathrm{id}_B \circ f = f$.

Un exemple de catégorie est la catégorie dont les objets sont les ensembles et les flèches sont les fonctions entres ensembles.

1. Définissez une catégorie dont les objets sont les types d'OCaml (munis d'un opérateur $\to$ pour les types des fonctions et d'un type $+$ pour les types somme). On ne considèrera pas les types enregistrements.

Un foncteur $F : \mathcal{A} \to \mathcal{B}$ associe à un objet de la catégorie $\mathcal{A}$ un objet de la catégorie $\mathcal{B}$ et qui préserve la composition des flèches.

2. Montrez que dans la catégorie précédemment définie, un constructeur de types F est un foncteur s'il existe une fonction polymorphe `fmap` : $(A \to B) \to (FA) \to (FB)$ qui vérifie les lois suivantes :

$$\texttt{fmap}\ \mathrm{id} = \mathrm{id} \text{ et } \texttt{fmap}\,(f \circ g) = (\texttt{fmap}\ \ f) \circ (\texttt{fmap}\ \ g)$$

3. Définissez un constructeur de type `option` tel qu'un élément de `option A` est soit vide soit un élément de type `A`. Montrez que `option` et `list` sont des foncteurs.

On dit qu'un foncteur M est une *monade* s'il existe deux fonctions polymorphes `return` : $A \to (MA)$ et `bind` : $(MA) \to (A \to (MB)) \to (MB)$ telles que :

- $\texttt{bind}\,(\texttt{return}\,x)\,f = fx$
- $\texttt{bind}\,x\,\texttt{return} = x$
- $\texttt{bind}\,(\texttt{bind}\,x f)\,g = \texttt{bind}\,x\,(y \mapsto (\texttt{bind}\,(fy)\,g))$

4. Montrez que `option` et `liste` sont des monades.

Pour tout type S on définit $(\texttt{etat}\,S)\,A$ comme le type polymorphe (S, A). On suppose l'existence d'un type `memoire` représentant un dictionnaire entre `string` et `int` avec les fonctions : `trouve` : `string` $\to$ `memoire` $\to$ (`optionint`) et `maj` : `string` $\to$ `int` $\to$ `memoire` $\to$ `memoire`.

5. Montrer que pour tout S, $\texttt{etat}\,S$ est une monade.
6. En déduire l'écriture avec `bind` d'une procédure qui prend en entrée trois clefs, récupère successivements deux valeurs d'une mémoire à partir des premières clefs puis associe dans la mémoire la somme des deux valeurs récupérée à la troisième clef.

On propose une définition alternative des monades. Au lieu de définir `bind` on définit `join` : $M\,(MA) \to (MA)$ qui vérifient les lois suivantes :

- $\texttt{join} \circ \texttt{return} = \mathrm{id}$
- $\texttt{join} \circ (\texttt{fmap}\,\texttt{return}) = \mathrm{id}$
- $\texttt{join} \circ \texttt{join} = \texttt{join} \circ (\texttt{fmap}\,\texttt{join})$.

7. Montrez que les deux définitions sont équivalentes.
8. Donnez une condition suffisante pour que $(M_1 \circ M_2)$ soit une monade.

Matthieu Boyer
Planche B : Corrigés

Khôlle MPI
DATE

Matthieu Boyer
Planche C

Polynômialisation

Dans la suite, on représentera un polynôme de degré n par un tableau de taille $n+1$ tel que a_0 soit le premier élément du tableau.

1. Donnez un algorithme qui évalue un polynôme en $\mathcal{O}(n)$.
2. Rappelez les racines n-èmes de l'unité dans $\mathbb{C}$. Quelle est leur somme ? Quel est leur produit ?
3. Montrez que les matrices de Vandermonde des racines de l'unité parcourue en sens trigonométrique $1, \omega, \omega^2, \dots$ et en sens horaire sont inverses l'une de l'autre.
4. Montrez qu'un polynôme de degré n peut se représenter de manière unique en n points. Donnez une méthode pour multiplier deux polynômes.
5. En déduire un algorithme permettant de représenter efficacement un polynôme.

Solution: Polynômialisation

For $P = \sum_{i=0}^{n-1} a_i x^i$, we define :

$$\begin{aligned} P_{odd}(x) &= a_{n-1}x^{n/2-1} + a_{n-3}x^{n/2-3} + \ldots + a_1 x \\ P_{even}(x) &= a_{n-2}x^{n/2-2} + a_{n-4}x^{n/2-4} + \ldots + a_2 x^{2/2} + a_0 \end{aligned}$$

1. We have : $P = xP_{odd}(x^2) + P_{even}(x^2)$
2. We evaluate P_{odd}, P_{even} at $(\omega_n^i)^2$ recursively by the halving property.
3. We combine the result.