

Rapport de TP 4MMAOD : Génération d'ABR optimal

SAMY AMRAOUI

MATTHIEU CHARLY-DESROCHES

samy.amraoui@ensimag.grenoble-inp.fr

matthieu-charly-desroches@ensimag.grenoble-inp.fr

23 mars 2017

1 Équation de Bellman

Question 1

Considérons l'ensemble d'éléments $e_0 < e_1 < \dots < e_{n-1}$ associés aux probabilités $(p_i)_{0 \leq i \leq n-1}$. Soit A un ABR *optimal* pour les (e_i) , i.e. minimisant la quantité $\sum_{i=0}^{n-1} p_i \Delta_A(e_i)$. Soit B un sous-arbre de A qui contient les noeuds $e_0 < e_1 < \dots < e_{m-1}$ avec $m \leq n$, et supposons que ce dernier n'est pas optimal. Comme B est un sous-arbre, il existe une constance $\delta \in \mathbb{N}$ telle que pour tout $i \in \llbracket 0, m-1 \rrbracket$, $\Delta_B(e_i) = \Delta_A(e_i) - \delta$. Or B n'est pas optimal, donc il existe un arbre C reprenant les noeuds de B tel que $\sum_{i=0}^{m-1} p_i \Delta_C(e_i) < \sum_{i=0}^{m-1} p_i \Delta_B(e_i)$. On a alors :

$$\sum_{i=0}^{m-1} p_i \Delta_C(e_i) + \delta \left(\sum_{i=0}^{m-1} p_i \right) + \sum_{i=m}^{n-1} p_i \Delta_A(e_i) < \sum_{i=0}^{m-1} p_i \Delta_A(e_i) + \sum_{i=m}^{n-1} p_i \Delta_A(e_i)$$

En remplaçant le sous-arbre B par C dans A , on en déduit que $\forall i \in \llbracket 1, m-1 \rrbracket$, $\Delta_C(e_i) + \delta = \Delta_{A'}(e_i)$ où A' est le nouvel arbre formé, donc on a :

$$\begin{aligned} \sum_{i=0}^{m-1} p_i \Delta_{A'}(e_i) + \sum_{i=m}^{n-1} p_i \Delta_A(e_i) &< \sum_{i=0}^{m-1} p_i \Delta_A(e_i) + \sum_{i=m}^{n-1} p_i \Delta_A(e_i) \\ \sum_{i=0}^{m-1} p_i \Delta_{A'}(e_i) &< \sum_{i=0}^{m-1} p_i \Delta_A(e_i) \end{aligned}$$

ce qui est contradictoire avec le fait que A est un ABR *optimal*.

Pour calculer le coût de recherche d'un élément e_i de manière récursive en partant de la racine de l'ABR, on remarque qu'il suffit de sommer p_i à chaque descente dans les sous-arbres gauches et droits tant qu'on ne l'a pas trouvé. On peut alors obtenir récursivement le produit $p_i \Delta_A(e_i)$. Soit $C_{i,j}$ le coût de calcul de l'ABR optimal pour les éléments $e_i < \dots < e_j$. Considérons que e_r où $r \in \llbracket i, j \rrbracket$ soit le noeud racine de cet ABR. On en déduit alors que :

$$C_{i,j} = \min_{r=i \dots j} \left\{ \underbrace{C_{i,r-1} + \sum_{k=i}^{r-1} p_k}_{\text{sous-arbre gauche}} + \underbrace{p_r + C_{r+1,j}}_{\text{sous-arbre droit}} \right\} = \sum_{k=i}^j p_k + \min_{r=i \dots j} \{C_{i,r-1} + C_{r+1,j}\}$$

avec comme conditions aux bords : $\forall i \in \llbracket 0, n-1 \rrbracket$, $C_{i,i} = p_i$ et $\forall i > j$, $C_{i,j} = 0$.

Question 2

Les calculs des sommes $\sum_{k=i}^j p_k$ pour $0 \leq i \leq j \leq n-1$ peut se faire en $\Theta(n^2)$ additions en stockant en mémoire les sous-sommes successives de la forme $\sum_{k=i}^j p_k = \sum_{k=i}^{j-1} p_k + p_j$. Pour le calcul récursif du coût minimum des sous-arbres, on utilise une boucle **for** sur tous les couples (i, j) avec $0 \leq i \leq j \leq n-1$ soit $\Theta(n^2)$ opérations à l'intérieur desquelles on va itérer sur l'indice r entre i et j qui sera retenu comme minimum, il s'agit d'une boucle **for** interne en $\Theta(n)$. On en déduit au total un coût en $\Theta(n^3)$ opérations.

Au niveau du coût en mémoire, l'algorithme nécessite de stocker les valeurs des $C_{i,j}$ où $0 \leq i \leq j \leq n-1$ dans une matrice triangulaire de taille $n \times n$ et les $\sum_{k=i}^j p_k$ où $0 \leq i \leq j \leq n-1$ de même dans une autre matrice triangulaire de taille $n \times n$, soit un coût total en mémoire en $\Theta(n^2)$.