Preuve de l'optimalité du codage de Huffman

Cette démonstration est tout à fait en dehors du programme de Terminale.

1. Tout d'abord on montre ce que ça fait dans un arbre d'échanger deux feuilles. Mettons qu'on a une lettre de poids 5 à la profondeur 3 et qu'on l'échange avec une lettre de poids 2 à la profondeur 6.
Leur coût avant (la place que cela prenait pour coder 5 fois la lettre de code de taille 3 + 2 fois la lettre dont le code est de longueur 6) était 5 x 3 + 2 x 6 = 15 + 12 = 27, et après le coût devient 5 x 6 + 2 x 3 = 36.
La différence "après - avant" est donc 5 (=le coût de la lettre qui s'éloigne de l'entrée)- 2 (=le coût de celle qui se rapproche), le tout x 3 (l'écart de profondeur entre les deux positions), ici (5 - 2) x 3 = 9.

2. En se basant sur cela, on peut prouver que, si notre ensemble contient au moins deux lettres, il existe un arbre optimal pour lequel les deux lettres de poids minimum (ou deux parmi les lettres de poids minimum) l1 et l2 sont reliées ensemble. Si on a un arbre optimal pour lequel ce n'est pas le cas, prenons une lettre l3 qui est le plus loin de la racine.

On a alors deux cas :

- soit l3 est reliée à une autre lettre l4 à la même hauteur
- soit l3 n'est reliée avec personne (il n'y a aucune lettre à l'autre bout de son couloir), auquel cas on peut enlever le couloir et raccrocher la lettre l3 une place plus haut dans l'arbre. C'est impossible car ça donnerait un arbre de poids total plus petit et on a supposé que notre arbre était optimal.

Du coup l4 existe et si on échange l1 et l3 puis l2 et l4 (remarque : on n'est pas sûrs que l1 et l2 soient tous différents de l3 et l4 mais peu importe), comme l1 et l2 sont soit à la même profondeur soit moins loin de la racine, et que l1 et l2 ont un poids inférieur ou égal à l3 et l4, au pire le poids de l'arbre ne change pas, au mieux il diminue. Le nouvel arbre est donc optimal et l1 et l2 sont reliées ensemble.

3. Ensuite il faut se convaincre que, quand on a deux lettres l1 et l2 de poids respectif p1 et p2 qui sont reliées ensemble dans un arbre A, le poids de l'arbre (on le note poids(A)) est le même que poids(A 0) + p1 + p2, où A 0 est construit en enlevant à A les lettres l1 et l2 avec le couloir qui les relie, et en rajoutant au bout du couloir devenu vide une lettre de poids p1 + p2. En effet, si les lettres l1 et l2 sont à une profondeur n de la racine, le poids de A est égal à la somme du poids des autres lettres + n × p1 + n × p2 et le poids de A' est égal à la somme du poids des autres lettres + (n - 1) × (p1 + p2).

4. Enfin on va montrer qu'un arbre créé grâce à l'algorithme de Huffman est optimal.

Pour cela on le fait par induction, à savoir qu'on va montrer que c'est vrai pour un arbre à une lettre, puis prouver que si c'est vrai pour tout arbre à strictement moins de k lettres alors c'est aussi vrai pour les arbres de k lettres.

- tout d'abord à une lettre c'est simple, on la met directement à la racine de l'arbre, pas besoin de couloir et le poids total est 0. C'est du coup optimal (mais d'un point de vue codage de texte ça n'a pas vraiment de sens). Avec deux lettres pas de choix non plus, on les relie ensemble et c'est réglé.
- on suppose ensuite que la propriété est vraie pour tous les arbres jusqu'à k -1 lettres (on appelle cela notre hypothèse d'induction). Prenons maintenant un ensemble de k lettres et appelons A H un arbre obtenu à partir de ces lettres en utilisant l'arbre de Huffman. Comme il a été créé en commençant par relier entre elles deux lettres se poids minimum (mettons l 1 et l 2), on va appeler maintenant A un arbre optimal ayant l1 et l2 reliés entre eux, vu qu'on a montré qu'il existe. D'après ce qu'il y a ci-dessus on sait que le coût de A est égal au coût de A 0 défini ci-dessus + p1 + p2. De même pour A H son coût est égal à celui de A 0 H (à savoir A H à qui on a fait la même transformation que de A vers A 0)+p1 + p2. Or A 0 H et A 0 sont deux arbres, de taille strictement plus petite que k, construits avec

les mêmes lettres l'un et l'autre. Comme A 0 H a été créé en suivant l'algorithme de Huffman (qui nous dit précisément qu'une fois deux lettres reliées ensemble on les considère comme une seule lettre dont le poids est la somme des deux poids de départ) et est de taille $<$ k, notre hypothèse d'induction nous dit que A 0 H est optimal, et a donc un coût inférieur ou égal à celui de A 0 . On en déduit :

$$\text{poids(A H)}= \text{poids(A 0 H)} + \text{p 1} + \text{p 2} \leqslant \text{poids(A 0)} + \text{p 1} + \text{p 2} \leqslant \text{poids(A)}$$

On a donc le poids de A H qui est inférieur où égal à celui de A. Comme A est par définition optimal, A H l'est aussi et la preuve est finie.

source : Marie Duflot-Kremer