Université de Bordeaux
L2 Informatique/Maths-Informatique

Probabilités, Statistiques, Combinatoire : TD3

Combinatoire : codage des arbres binaires

`https://moodle1.u-bordeaux.fr/course/view.php?id=7792`

Exercice 3.1

Le premier arbre a pour codage *aaababbbaabababbab*.

Pour simplifier le calcul du codage du second arbre, on peut remarquer que ses sous-abres gauche et droit sont les mêmes, ils ont donc le même codage.

L'arbre gauche a pour codage, $w = abaababb$. Le codage de l'arbre entier est donc $awbw$, soit *aabaababbbabaababb*.

Exercice 3.2

1. Pas de dessins, mais quand on calcule les mots de codage pour les peignes à 4, 5 ou 6 nœuds internes, on trouve :

n	Gauche	Droit
4	*aaaabbbb*	*abababab*
5	*aaaaabbbbb*	*ababababab*
6	*aaaaaabbbbbb*	*abababababab*

   On peut assez facilement proposer une formule pour le codage du peigne gauche à n nœuds internes : $a^n b^n$; et pour le codage du peigne droit à n nœuds internes : $(ab)^n$. Ces deux formules sont assez faciles à prouver (par récurrence sur n), une fois qu'on repère qu'on peut définir le peigne gauche G_n comme ayant comme sous-arbre gauche le peigne gauche à $n-1$ nœuds internes, et comme sous-arbre droit, l'arbre réduit à une racine ; et le peigne droit à n nœuds internes, D_n, comme l'arbre ayant comme sous-arbre gauche l'arbre réduit à une racine, et comme sous-arbre droit, le peigne droit D_{n-1}. En appliquant la formule récursive pour le codage d'un arbre en fonction des codages de ses sous-arbres, on a ainsi, en notant w_n pour le codage de G_n, et w'_n pour le codage de D_n : $w_1 = w'_1 = ab$, et pour $n \geq 2$,

$$w_n = a \cdot w_{n-1} \cdot b$$
$$w'_n = ab \cdot w'_{n-1}$$

   ce qui permet facilement de prouver les formules, par récurrence immédiate sur n.

2. Pas de dessins non plus pour les arbres parfaits. On trouve, pour l'arbre parfait de hauteur 2, un mot de codage *aabbab* ; et pour l'arbre parfait de hauteur 3, *aaabbabbaabbab*. On ne voit pas de "formule" simple pour le mot codant l'arbre parfait de hauteur h. On peut, dans la même veine, écrire une formule récursive : si w''_n désigne le mot codant l'arbre parfait de hauteur h, on a $w''_1 = ab$, et pour $h \geq 2$, $w''_h = aw''_{h-1}bw''_{h-1}$.

Exercice 3.3

Impossible de donner une correction pour cet exercice, dont l'intérêt réside dans le fait de l'expérimenter.

Exercice 3.4

Si l'on prend le peigne droit à 5 noeuds internes, il y a 6 façons de l'obtenir comme résultat de la transformation de Rémy (autant qu'il y a de façons de choisir une de ses feuilles pour être la feuille marquée).

Si l'on fait le calcul inverse à partir des 6 peignes droits avec feuille marquée, on trouve les résultats suivants :

— si on marque une des 5 feuilles gauches, le résultat est le peigne droit à 4 noeuds internes, avec un des noeuds de sa branche droite marqué (l'un des 4 noeuds internes, ou la feuille au bout de la branche droite), et la direction Gauche.

— si on marque l'unique feuille droite, le résultat est encore le peigne droit à 4 noeuds internes, avec comme noeud marqué la feuille droite (au bout de la branche droite), mais comme direction, la direction Droite.

Exercice 3.5

1. En expérimentant un peu, on arrive assez facilement à la conjecture selon laquelle la longueur de la première montée du mot de Dyck qui code un arbre t, correspond systématiquement à la longueur de la branche gauche de cet arbre (où l'on compte la profondeur de la feuille qui termine cette branche).

   Il n'est pas très difficile de prouver cette conjecture : en effet, quand on calcule le mot de codage de l'arbre par un parcours préfixe, les premiers noeuds du parcours sont précisément ceux de la branche gauche. Comme chaque nœud interne est codé par un a, et la feuille qui termine la branche, par un b, on aura autant de a précédant le premier b qu'il y a de noeuds internes dans cette branche.

2. Pour la deuxième question, il est plus facile de passer par le codage récursif. Si le sous-arbre gauche de l'arbre t comporte k noeuds internes (donc $2k+1$ noeuds au total), le mot de codage de t va commencer par a, puis les $2k$ lettres du mot de codage w_g de l'arbre gauche, puis b. Or, il est facile de voir que ce préfixe aw_gb du mot de codage de t, est le plus court préfixe (à part le mot vide) du mot de codage de t qui soit un mot de Dyck. En effet, vu que w_g est un mot de Dyck, c'est un mot équilibré, et donc aw_gb est aussi un mot équilibré ; et si on prend un préfixe plus court mais non vide, il est de la forme au, où u est un préfixe de w_g ; comme w_g est un mot de Dyck, u a au moins autant de a que de b, et donc au a strictement plus de a que de b, et au n'est donc pas un mot de Dyck.

   Donc, au final, la taille du sous-arbre gauche est liée à la longueur du plus court préfixe non vide qui soit un mot de Dyck : si cet arbre est de taille k, le préfixe est de taille $2k+2$.

3. D'après le raisonnement précédent, le premier retour à 0 correspond, dans le parcours de l'arbre, à la fin du parcours du sous-arbre gauche. En fait, le nombre total de retours à 0 correspond exactement à la longueur de la branche droite de l'arbre (on ne donne pas ici de preuve de cette affirmation).

4. D'après les questions précédentes, le nombre de mots de Dyck de longueur 100 et commençant par $a^{12}b$ est exactement le nombre d'arbres binaires complets de taille 50 et dont la branche gauche est de longueur 12 ; et le nombre de mots de Dyck de longueur 100 et ayant 12 retours à 0, est exactement le nombre d'arbres binaires complets de taille 50 et dont la branche droite est de longueur 12.

   Or, il y a une symétrie assez évidente entre les branches droites et gauche : si t est n'importe quel arbre, et t' l'arbre symétrique (obtenu récursivement en échangeant, dans chaque noeud, la gauche et la droite), la longueur de la branche gauche de t est aussi la longueur de la branche droite de t'. Cette opération de symétrie est une bijection (sa bijection inverse est elle-même : c'est ce qu'on appelle une *involution* sur l'ensemble des bijections de l'ensemble des arbres dans lui-même) ; par conséquent, pour tout n et pour tout k, il y a autant d'arbres binaires complets de taille n et de branche gauche de longueur k, que d'arbres binaires complets de taille n et de branche **droite** de longueur k.

   Bref, on ne sait pas dire simplement **combien** d'arbres il y a avec ces conditions, mais on peut dire qu'il y en a autant d'un type que de l'autre.

   Et donc, il y a autant de mots de Dyck de longueur 200 avec 12 retours à zéro, que de mots de Dyck de longueur 200 avec une première montée de longueur 12.

Exercice 3.6
Principe de réflexion étendu

1. Les chemins de $(0, k)$ à $(n, n+k)$ font exactement, et dans n'importe quel ordre, n pas Nord et n pas Est. Il y en a autant que de chemins de $(0, 0)$ à (n, n), soit $\binom{2n}{n}$.
2. Un chemin de $(0, k)$ à $(n, n + k)$ fait n pas Est, et n pas Nord. Comme il commence à k au-dessus de la diagonale, pour passer strictement en-dessous, il faut qu'après un certain nombre de pas, il ait fait $k + 1$ pas Est de plus que de pas Nord. Pour cela, il faut que l'on ait $n \geq k+1$, soit $n > k$. Et à partir de $n = k+1$, on a le mot $E^{k+1}N^{k+1}$, dont on vérifie aisément qu'il passe en-dessous de la diagonale.
3. Prenons un chemin (codé par un mot w) de $(0, k)$ à $(n, n + k)$ qui n'est pas un chemin positif. Soit A le premier point du chemin qui se trouve strictement en dessous de la diagonale : c'est donc un point dont les coordonnées sont de la forme $(i, i - 1)$, avec $i - 1 \geq k$ donc $i > k$. Le mot w est donc de la forme $w = uv$, avec $|u|_E = i$ et $|u|_N = i - 1 - k$, et donc $|v|_E = n - i$ et $|v|_N = n + k - i + 1$ (le point A est le point atteint après la partie du chemin codée par le mot u).

   Appliquons une symétrie par rapport à la sous-diagonale (droite d'équation $y = x-1$) à la partie du chemin située après le point A. Cela revient à remplacer le mot v par le mot v', obtenu en changeant les N en E et les E en N. Le nouveau mot $w' = u \cdot v'$ code donc un chemin qui fait $i+(n+k-i+1) = n+k+1$ pas Est, et $(i-1-k)+(n-i) = n-k-1$ pas Nord ; en d'autres termes, ce nouveau chemin (qui coïncide avec le chemin de départ du point $(0, k)$ au point $A = (i, i - 1)$) va de $(0, k)$ à $(n + k + 1, n - 1)$ – on remarque que le point final ne dépend pas de i.

   La transformation du chemin d'origine est inversible, à partir du chemin de $(0, k)$ à $(n + k + 1, n - 1)$, on peut retrouver le chemin non positif de $(0, k)$ à $(n, n + k)$: le premier point sous la diagonale du nouveau chemin est le même point A, et on retrouve le chemin d'origine en appliquant la même symétrie à la partir après A. Par

conséquent, il y a autant de chemins non positifs de $(0,k)$ à $(n,n+k)$ que de chemins de $(0,k)$ à $(n+k+1,n-1)$, soit $\binom{2n}{n+k+1}$.
Par soustraction, on obtient une formule pour le nombre de chemins positifs de $(0,k)$ à $(n,n+k)$: il y a $\binom{2n}{n}$ chemins au total, dont $\binom{2n}{n+k+1}$ chemins non positifs, donc il y a $\binom{2n}{n} - \binom{2n}{n+k+1}$ chemins positifs.

4. En appliquant la formule précédente, pour $k = 1$ on trouve le nombre de chemins positifs de $(0,1)$ à $(n,n+1)$: $\binom{2n}{n} - \binom{2n}{n+2}$. On triture un peu la formule :

$$\begin{aligned}
\binom{2n}{n} - \binom{2n}{n+2} &= \frac{(2n)!}{n!n!} - \left(\frac{(2n)!}{(n+2)!(n-2)!}\right) \\
&= \frac{(n+2)(n+1)(2n)!}{(n+2)!n!} - \frac{n(n-1)(2n)!}{(n+2)!n!} \\
&= \frac{(2n)!(n^2+3n+2-n^2+n)}{(n+2)!n!} \\
&= \frac{2(2n+1)(2n)!}{(n+2)!n!} \\
&= \frac{(2n+2)(2n+1)(2n)!}{(n+1)(n+2)!n!} \\
&= \frac{(2n+2)!}{(n+2)!(n+1)!} \\
&= C_{n+1}.
\end{aligned}$$

(Ces calculs se vérifient facilement, il s'agit de simplifications de fractions. La difficulté est de **trouver** la formule que l'on cherche à prouver, mais une fois qu'on a la formule précédente avec une différence de deux coefficients binomiaux il suffit de la programmer et de regarder les premières valeurs pour repérer le début de la suite de Catalan décalée.)

En d'autres termes, il y a autant de chemins positifs de $(0,1)$ à $(n,n+1)$ que de chemins de Dyck de longueur $2n+2$ (ou, de manière équivalente, de chemins positifs de $(0,0)$ à $(n+1,n+1)$. Pour le coup, il y a une bijection très simple qui passe de l'un à l'autre : pour passer d'un chemin positif de $(0,1)$ à $(n,n+1)$ à un chemin positif de $(0,0)$ à $(n+1,n+1)$, il suffit d'ajouter un pas Nord avant, et un pas Est après ; comme chaque chemin positif de $(0,0)$ à $(n+1,n+1)$ commence forcément par un pas Nord et finit forcément par un pas Est, la transformation inverse consiste simplement à "effacer" le premier et le dernier pas.

Exercice 3.7

Arbres ternaires complets

Un **arbre ternaire complet** est un arbre dans lequel chaque noeud interne (noeud autre qu'une feuille) a exactement trois enfants ; comme pour les arbres binaires, on distingue un ordre sur les enfants. La **taille** d'un arbre ternaire complet est son nombre de noeuds internes.

1. Toujours pas de dessins, mais on trouve 1 arbre à 0 noeud interne (l'arbre réduit à une feuille), 1 arbre à 1 nœud interne, 3 arbres à 2 nœuds internes, et 12 arbres à 3 nœuds internes. La suite de comptage commence donc par $(1,1,3,12)$.

2. Sur les dessins, on constate que l'arbre sans nœud interne a 1 feuille, l'arbre à 1 nœud interne a 3 feuilles, les arbres à 2 nœuds internes ont 5 feuilles, et les arbres à 3 nœuds internes ont 7 feuilles. Cela donne envie de proposer une formule générale : un arbre à n nœuds internes a $2n+1$ feuilles.

   Ce n'est pas bien compliqué à prouver, par récurrence (forte) sur le nombre de nœuds internes. Soit P_n la propriété : "tout arbre ternaire complet à $k \leq n$ nœuds internes, a exactement $2k+1$ feuilles".

   — **Initialisation :** pour $n = 0$, le seul entier $k \leq n$ est $k = 0$, et le seul arbre ternaire complet à $k = 0$ nœuds internes a bien 1 feuille. Donc P_0 est vraie.

   — **Hérédité :** soit $n \geq 0$ un entier tel que P_n soit vraie, soit $k \leq n+1$ un entier, et t un arbre ternaire à k nœuds internes.

     — si $k < n+1$, alors $k \leq n$, et d'après P_n, t a $2k+1$ feuilles.

     — sinon, $k = n+1$. Appelons t_1, t_2 et t_3 les trois sous-arbres (gauche, milieu, droit) de t, et k_1, k_2 et k_3 leurs nombres respectifs de nœuds internes. Chacun de ces trois arbres a au plus $k-1 \leq n$ nœuds internes, donc P_n nous assure que t_i a $2k_i+1$ feuilles. On peut donc calculer le nombre toal de feuilles de t : chaque feuille de t est soit dans t_1, soit dans t_2, soit dans t_3, et donc le nombre de feuilles de t est $(2k_1+1)+(2k_2+1)+(2k_3+1) = 2(k_1+k_2+k_3+1)+1$. Or, $k_1+k_2+k_3+1$ n'est autre que le nombre total de nœuds internes de t (le $+1$ vient de la racine), c'est donc $k+1$, et t a bien $2(k+1)+1$ feuilles.

     Comme on a pris k quelconque et t quelconque, on a bien prouvé P_{n+1}. En d'autres termes, on a $P_n \implies P_{n+1}$.

   — Donc, par le principe de récurrence, P_n est vraie pour tout n, et donc tout arbre ternaire complet a un nombre de feuilles qui est égal à son nombre de nœuds internes, plus 1.

3. Le codage suivant fonctionne, et s'inspire du codage récursif des arbres binaires complets : l'arbre réduit à sa racine est codé par le mot vide ϵ, et pour n'importe quel a autre arbre ternaire complet, si ses trois sous-arbres t_1, t_2 et t_3 sont codés par les mots w_1, w_2 et w_3, l'arbre entier est codé par $a \cdot w_1 \cdot b \cdot w_2 \cdot b \cdot w_3$.

   Une définition alternative, utilisant un parcours préfixe, consiste à coder les nœuds de l'arbre (sauf la dernière feuille) dans l'ordre du parcours : a pour les nœuds internes, et b pour les feuilles. On peut montrer que cela définit le même codage.

4. Pas de dessins ici, mais on constate, en dessinant les chemins à pas Nord et Est correspondant aux arbres de petite taille, que les chemins semblent aller du point $(0,0)$ à un point de la forme $(k, 2k)$ (cette partie est claire, au vu de la formule donnant le nombre de feuilles en fonction du nombre de nœuds internes), tout en ne passant jamais "sous" la droite d'équation $y = x/2$ qui joint le point de départ au point d'arrivée ; et de plus, il semble bien que les chemins correspondant aux mots de codage, correspondent **exactement** à tous les chemins de cett forme.

   On propose donc cela comme conjecture : les chemins correspondant aux mots de codage des arbres ternaires complets sont exactement les chemins (à pas Nord et Est) allant du point $(0,0)$ à un point de la droite $y = x/2$, sans jamais passer sous cette droite.

   (C'est bien le cas, et cela peut se prouver avec une adaptation du lemme cyclique ; on ne le prouve pas ici)