ASPECTS ALGORITHMIQUES DE LA COMBINATOIRE
Examen du 23 février 2006 – durée : 2 heures 30

Notes de cours autorisées

PROBLÈME 1 : Génération aléatoire d'arbres non enracinés

Soit A un *arbre*, *i.e.* un graphe connexe sans cycle. Un *chemin* de longueur k dans A est une suite alternée $(s_0, a_1, s_1 \ldots, a_k, s_k)$ de $k+1$ sommets et k arêtes de A telle que pour tout $i \in [\![1,k]\!]$, $a_i = \{s_{i-1}, s_i\}$; s_0 et s_k sont les *extrémités* du chemin, et son *milieu* est son $k+1^{\text{e}}$ élément (qu'il s'agisse d'un sommet ou d'une arête).

La *distance* entre deux sommets est la longueur d'un plus court chemin dont ils sont extrémités. Le *diamètre* $d(A)$ de A est la distance maximale entre deux de ses sommets. Les *chemins maximaux* de A sont les chemins de longueur $d(A)$ dans A.

Question 1.

1. Montrer que l'intersection des chemins maximaux de A est un chemin non vide. On appelle *centre de* A son milieu.
2. Proposer un algorithme pour déterminer le centre d'un arbre.

On considère désormais uniquement des arbres *plans* (non enracinés) et des arbres plans *plantés*, *i.e.* enracinés sur une de leurs feuilles. On dit qu'un arbre plan à k feuilles est *symétrique d'ordre* d si le nombre d'arbres plantés correspondants est exactement k/d.

3. Soit A un arbre plan ; décrire les symétries possibles de A en fonction de son centre.

Question 2. On appelle *arbre binaire* tout arbre plan dont les sommets sont soit des nœuds de degré 3, soit des feuilles. On note $\mathcal{B}_n$ et $\mathcal{B}_n^{\perp}$ l'ensemble des arbres binaires (respectivement des arbres binaires plantés) à n nœuds, et b_n et $b_n^{\perp}$ leurs cardinaux.

1. Exprimer $(n+2)b_n$ en fonction de la famille $(b_k^{\perp})_{k\leqslant n}$.
2. En déduire la valeur de b_n.
3. Donner un algorithme de génération aléatoire uniforme pour $\mathcal{B}_n^{\perp}$, et en déduire un algorithme pour $\mathcal{B}_n$. Quelle est sa complexité ?

Question 3. On note $b_n^{(d)}$ le nombre d'arbres binaires à n nœuds et d'ordre de symétrie d.

1. Exprimer $b_n^{\perp}$ en fonction de la famille $\left(b_n^{(d)}\right)_{d\geqslant 1}$.
2. En tirant uniformément dans $\mathcal{B}_n^{\perp}$ puis en oubliant l'enracinement, avec quelle probabilité tire-t-on un élément de $\mathcal{B}_n$ fixé ?
3. En déduire un nouvel algorithme de génération aléatoire uniforme pour $\mathcal{B}_n$. Quelle est sa complexité ?

Question 4. Plus généralement, considérons une famille $\mathcal{A}$ d'arbres plans, et ses avatars suivants :

- la famille $\mathcal{A}^{\bullet}$ des arbres de $\mathcal{A}$ avec un sommet marqué ;
- la famille $\mathcal{A}^{-}$ des arbres de $\mathcal{A}$ avec une arête marquée ;
- la famille $\mathcal{A}^{\bullet\!-}$ des arbres de $\mathcal{A}$ avec une arête orientée ;
- la famille $\mathcal{A}^{\perp}$ des arbres de $\mathcal{A}$ plantés.

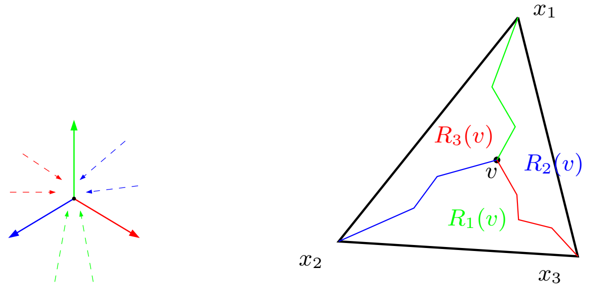
1. À l'aide de la notion de centre, montrer l'isomorphisme suivant (où la somme représente l'union disjointe) :
$$\mathcal{A} + \mathcal{A}^{\bullet\!-} \simeq \mathcal{A}^{\bullet} + \mathcal{A}^{-}.$$
2. En déduire que $\mathcal{A}$ peut s'exprimer en fonction de $\mathcal{A}^{\perp}$ à l'aide de constructions analogues aux constructions admissibles vues en cours.

PROBLÈME 2 : Triangulations, forêts de Schnyder et mots de Dyck

On considère des triangulations planaires dont la face extérieure est un triangle (x_1, x_2, x_3) de sens direct. L'ensemble I des sommets internes de T est constitué des sommets distincts des x_i. Une arête est interne si elle est incidente à un sommet interne.

Une *réalisation de Schnyder* $(T; T_1, T_2, T_3)$ est la donnée d'une triangulation planaire T munie d'une forêt de Schnyder (T_1, T_2, T_3) enracinée en (x_1, x_2, x_3). On rappelle qu'une forêt de Schnyder est une partition des arêtes internes en trois parties (T_1, T_2, T_3) (indiquées par les couleurs verte, bleue et rouge) telles que :

- les arêtes de T_i forment un arbre couvrant des sommets $I \cup \{x_i\}$,
- l'orientation des arêtes de chacun des arbres T_i vers le sommet x_i correspondant vérifie la condition locale de Schnyder donnée à la figure 1(a).



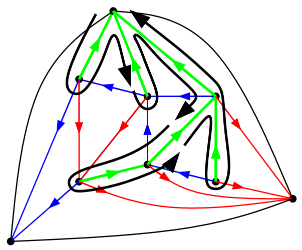
(a) Règle locale. (b) Régions $R_i(v)$.

FIG. 1 – Réalisations de Schnyder.

Étant donné une réalisation de Schnyder $(T; T_1, T_2, T_3)$, on définit pour chaque sommet interne v les trois chemins $(P_i(v))_{1\leqslant i\leqslant 3}$ d'extrémités v et x_i dans T_i, et les régions $(R_i(v))_{1\leqslant i\leqslant 3}$ délimitées par ces trois chemins. Plus précisément, on considèrera que ces régions contiennent leur bord, de sorte que $R_i(v) \cap R_{i+1}(v) = P_{i+2}(v)$.

Question 1. Un couple (w, w') de mots de Dyck sur l'alphabet $\{a, b\}$ de longueur $2n$ est dit *ordonné* si, pour tout $i \in [\![0, 2n]\!]$, les deux factorisations $w = uv$ et $w' = u'v'$ avec $|u| = |u'| = i$ vérifient $|u'|_a \geqslant |u|_a$.

Exprimer le fait qu'un couple de mots de Dyck est ordonné en fonction des chemins de Dyck correspondants.



$w = ababaaabbabb$
$w' = aaaabbaabbbb$

FIG. 2 – Une réalisation et le couple de mots associés.

Le problème est constitué de deux parties essentiellement indépendantes : la première montre que les réalisations de Schnyder à n sommets internes sont en bijection avec les couples ordonnés de mots de Dyck de longueur $2n$; la seconde s'intéresse à l'énumération et à la génération aléatoire de ces couples ordonnés de mots de Dyck.

Partie I. Une bijection

On considère l'application φ qui à toute réalisation $(T; T_1, T_2, T_3)$ associe un couple de mots (w, w'), obtenus au fur et à mesure d'un parcours du bord de l'arbre T_1 en sens direct à partir de sa racine x_1 dans la face infinie (voir figure 2) de la façon suivante :

- pour le mot w, au fur et à mesure du parcours, on écrit la lettre a lorsqu'on longe une arête de T_1 pour la première fois et b lorsqu'on la longe pour la seconde fois ;
- pour le mot w', on écrit la lettre a lorsqu'on coupe une arête sortante de T_2 et b lorsqu'on coupe une arête entrante de T_3, puis on ajoute autant de lettres b que nécessaire pour que $|w'| = |w|$.

Question 2. On veut tout d'abord montrer que l'application φ construit bien un couple de mots de Dyck ordonnés.

1. Décrire en terme des régions $R_i(v)$ l'ensemble des sommets déjà visités au moment où un nouveau sommet v est découvert durant le tour de l'arbre T_1.
2. Montrer que le mot w' est un mot de Dyck.
3. Montrer que le couple (w, w') est un couple de Dyck ordonné.

Question 3. Maintenant on s'intéresse à la réciproque.

1. Décrire l'application réciproque ψ. (Indication : à partir d'une paire (w, w'), on pourra commencer par reconstruire T_1 puis T_3 et enfin T_2.)
2. Montrer que les deux applications φ et ψ sont des bijections réciproques.

Partie II. Énumération et génération aléatoire

Question 4. À tout couple de mots de Dyck (w, w'), on associe un couple de chemins *décalés* (c, c') où c est le chemin de Dyck issu du point $(2, 0)$ associé au mot w et c' est le chemin de Dyck issu du point $(0, 0)$ et associé au mot $aaw'bb$.

1. Donner une caractérisation des couples de chemins décalés associés aux couples de mots de Dyck ordonnés.

2. Donner une bijection entre les couples de chemins de Dyck décalés de longueur commune $2n+2$ et un sous-ensemble des couples de chemins décalés de longueur respective $2n+4$ et $2n$.
3. En déduire que le nombre de couples de mots de Dyck ordonnés de longueur $2n$ est
$$C_{n+2}C_n - C_{n+1}^2$$
où $C_n = \frac{1}{n+1}\binom{2n}{n}$ est le nombre de chemins de Dyck de longueur $2n$. (En déduire le nombre de réalisations avec n sommets internes.)
4. En déduire un algorithme de génération aléatoire pour les couples de mots de Dyck ordonnés de longueur $2n$ basé sur le rejet. Quelle est la complexité de votre algorithme ?

Question 5. On note maintenant $c(i,j,k)$ le nombre de couples de mots (w,w') de $\{a,b\}^*$ tels que $|w| = |w'| = k$, $|w|_a = i$, $|w'|_a = j$ et tels que, pour toute factorisation $w = uv$, $w' = u'v'$ avec $|u| = |u'|$, on ait $|u'|_a \geqslant |u|_a$.

1. Écrire une relation de récurrence liant $c(i,j,k)$ aux $(c(i,j,k-1))_{i,j}$ pour $k \geqslant 1$.
2. En déduire un autre algorithme de génération aléatoire pour les couples de mots de Dyck ordonnés de longueur $2n$. Quelle est la complexité de votre algorithme ?