Master 2
MATHÉMATIQUES-INFORMATIQUE

Analyse d'Algorithmes et Arbres

Support de cours
pour le module T3

Brigitte CHAUVIN, Danièle GARDY

Automne 2004

Chapitre 1

Arbres, algorithmes et analyses

Lors de la conception d'un système informatique, ou après sa réalisation, se place une phase d'évaluation de ses performances. Pour cela, il y a plusieurs manières de faire, de la plus informatique à la plus mathématique :

- mesures sur un système existant ;
- simulations ;
- analyses théoriques.

Ces approches sont complémentaires ; aucune d'elles n'est suffisante à elle seule ; toutes sont utiles.

Les mesures apportent des informations précises sur le comportement d'un système donné, mais leur résultat peut dépendre de l'art du programmeur, de la machine (matériel ou logiciel ; il faut recommencer les mesures quand on change de machine), de la charge du système à un moment donné (toute observation modifie le système observé...), etc. De plus, pour mesurer les performances d'un système il faut l'avoir déjà réalisé... et si on veut prédire ? faire des choix entre plusieurs solutions ?

Les simulations peuvent elles aussi dépendre de la machine, de l'art de la personne qui a programmé la simulation, etc. Elles sont intéressantes pour obtenir des analyses ou des prédictions lorsqu'on ne peut pas faire de modèle mathématique, ou qu'un modèle précis est trop complexe pour être exploitable.

Les analyses théoriques fournissent des théorèmes ; une fois prouvée, la complexité "abstraite" est indépendante de l'art du programmeur et de la machine. Par contre, cette approche demande un effort intellectuel supplémentaire, avec des outils mathématiques qu'il faut maîtriser. Les résultats fournis doivent souvent être complétés par des mesures ; par exemple, si un algorithme nécessite n^2 opérations unitaires (lecture de clé, envoi de message...), encore faut-il savoir, pour une machine donnée, le temps pris par une opération unitaire.

Les travaux théoriques établissent d'abord des modèles, pour représenter des données, des algorithmes ou des systèmes, puis cherchent à démontrer des résultats mathématiques relatifs à ces modèles. Il faut donc, dans un premier temps, choisir les paramètres à estimer, qui mesureront un coût, typiquement le temps d'exécution ou la place mémoire pour exécuter un algorithme ou représenter une structure de données. Une fois ce paramètre de coût choisi, il y a deux types d'approche : soit on s'intéresse au cas le pire, soit au cas moyen, ou mieux encore à la distribution de probabilité. La première approche conduit à identifier des classes de problèmes toujours résolubles en un temps polynomial, par exemple, ou d'autres qui ne le sont pas ; c'est la théorie de la complexité abstraite. La deuxième approche part de l'idée que le comportement d'un algorithme est souvent caractérisé de façon plus pertinente par ses performances sur la plus grande partie des données d'entrée que sur des cas exceptionnels, et conduit à l'analyse en moyenne et en distribution ; c'est celle qui sera présentée ici.

Le comportement global d'un système est étudié dans ce qu'on appelle d'habitude l'"évaluation de performances". Par contraste, nous nous intéressons à des aspects précis d'un système : *l'analyse* fine du comportement d'un *algorithme,* agissant sur une *structure de données.*

1.1 Analyse d'algorithmes

L'analyse d'algorithmes s'applique à beaucoup de sujets ; l'unité de ce domaine vient plus du point de vue employé et des méthodes utilisées que du domaine de l'informatique étudié. Des exemples classiques sont les structures arborescentes ou les algorithmes de tri ou de hachage ; mais l'analyse d'algorithmes a aussi prouvé son intérêt dans des domaines tels que les systèmes de récriture, les bases de données, ou la gestion de réseaux, entre autres.

La démarche générale pour étudier un algorithme sera la suivante :
- établir un modèle des données que l'algorithme prend en entrées ;
- dégager le ou les paramètres pertinents, qui détermineront le *coût moyen* c_n de l'algorithme sur des données de taille n ;
- établir une formule pour ce coût moyen, par exemple à l'aide d'une équation de récurrence qu'on résoudra, soit directement, soit en employant des outils adaptés (séries génératrices) ;
- chercher un équivalent asymptotique de c_n, indiquant la façon dont c_n croît avec la taille n des données.

Nous avons parlé ici du coût moyen d'un algorithme. Plus précisément, le coût d'un algorithme sur une structure de données de taille n peut être considéré comme une variable aléatoire ; on voudra alors calculer, en plus de son coût moyen, sa variance, et éventuellement d'autres moments, voire étudier la distribution de cette variable aléatoire, ou plutôt la convergence asymptotique vers une v.a. de loi connue (par exemple Gaussienne). On peut aussi envisager d'étudier le comportement dynamique d'une structure de données, lorsqu'on ajoute ou supprime des données ; le résultat cherché est alors un processus aléatoire, caractérisant l'évolution du coût au cours du temps. De telles études font naturellement intervenir des outils mathématiques puissants, souvent de nature probabiliste même si leur base est souvent, comme pour l'étude en moyenne "simple", l'écriture de la série génératrice des coûts.

1.2 Arbres aléatoires

Il vient d'être dit que la représentation par processus aléatoires permet parfois une analyse fine des structures des algorithmes. Les structures arborescentes apparaissent fréquemment, ce qui conduit naturellement à une modélisation par des arbres aléatoires. De plus, les arbres aléatoires peuvent aussi être vus comme un cas particulier de graphes aléatoires, qui sont eux-mêmes des modèles des réseaux de télécommunication, par exemple.

Une bonne manipulation des arbres aléatoires nécessite de travailler sur des espaces de probabilités. Les processus aléatoires (on dit aussi stochastiques), qui appartiennent à la grande famille des processus de branchement, seront souvent utilisés. Ils peuvent décrire simplement l'évolution d'une population en comptant le nombre d'individus à la n-ième génération, ou bien décrire des évolutions plus riches, ce qui correspond à l'idée de marquer les nœuds ou les branches d'un arbre par toutes sortes de variables qui présentent un intérêt pour l'étude.

Mais d'autres processus stochastiques pourront aussi intervenir (ce sera le cas par exemple pour les arbres binaires de recherche) et dans tous les cas, des méthodes probabilistes classiques, couplées parfois à des méthodes analytiques, seront d'un grand secours pour comprendre l'asymptotique d'une variable de coût d'un algorithme. Des martingales apparaîtront, des marches aléatoires seront associées aux arbres et les différentes notions de convergence fourniront autant de notions de limite.

1.3 Analyse automatique et calcul formel

Il peut arriver que les équations sur les coûts soient relativement faciles à obtenir (par exemple une équation de récurrence), mais que leur résolution exacte demande des outils mathématiques assez complexes. L'utilisation d'un système de calcul formel permet dans certains cas d'obtenir une solution mathématique, et dans bien des cas de développer une intuition de la solution.

L'analyse d'algorithmes est donc partiellement automatisable, et automatisée ; le système *Luo*, initialement développé à l'Inria par P. Flajolet, B. Salvy et P. Zimmermann au début des années 90, permet l'analyse automatique de paramètres liés à des structures de données appartenant à une certaine classe (les *constructions admissibles*) : à partir de leurs spécifications, le système construit la série génératrice de coût associée et donne l'équivalent asymptotique du coût moyen. Une version ultérieure a été intégrée aux versions récentes du système de calcul formel *Maple,* sous le nom de *Combstruct.*

1.4 Thèmes étudiés

Ce cours d'introduction présentera les thèmes suivants :

– Les tris : tri à bulles, tri par sélection, tri par insertion, tri rapide ; modèle des permutations aléatoires.
– Le hachage : hachage externe (complexité d'une recherche avec ou sans succès, plus longue sous-liste) ; hachage linéaire (complexité d'une recherche avec ou sans succès).
– Une structure arborescente de base : les arbres binaires ; énumération, étude de paramètres de base (longueur de cheminement) ; extension aux familles simples

d'arbres.
- Processus de branchement ; arbre de Galton-Watson ; arbre biaisé ; processus surcritique et théorème de Kesten-Stigum ; marche aléatoire branchante.
- Arbres binaires de recherche : hauteur, coût ; lien avec le tri rapide. Méthode de contraction.

Pour chacun de ces sujets, nous rappellerons (ou présenterons) les structures de données informatiques de base, et montrerons comment dégager les paramètres de coût et construire un modèle mathématique permettant l'analyse de ces paramètres. Les outils mathématiques nécessaires seront présentés lorsque nous en aurons besoin : fonctions génératrices, analyse complexe et théorie des fonctions analytiques (formule de Cauchy, lemmes de transfert...), méthode de point fixe, convergence de lois de probabilité, martingales...

1.5 Bibliographie

1.5.1 Algorithmique

Quelques livres de base présentant les algorithmes et structures de données classiques, avec leurs performances (sans toujours les justifications) :

1. **T.H. Cormen, C.E. Leiserson, R.L. Rivest :** *Introduction à l'algorithmique.* Dunod, 1990.
   Gros livre, contient des tas de choses à un niveau assez élémentaire.
2. **C. Froidevaux, M-C. Gaudel, M. Soria :** *Types de données et Algorithmes.* McGraw Hill Paris, 1990.
3. **R. Sedgewick :** *Algorithms.* Addison-Wesley.
   Donne beaucoup d'algorithmes ; il existe des versions reprenant ce livre en codant les algorithmes dans divers langages de programmation, et des traductions françaises.
4. **S. Skiena :** *The Algorithm design manual.* Springer Verlag, 1997.
   Présente des techniques de conception d'algorithmes, puis donne des indications sur nombre de problèmes classiques, avec en particulier des références à des sites web. L'auteur lui-même a créé un site web sur lequel sont disponibles des implémentations d'algorithmes.

1.5.2 Analyse d'algorithmes

1. **P. Flajolet, R. Sedgewick :** *An introduction to the analysis of algorithms.* Addison-Wesley, 1996.
   Par des spécialistes incontestés du sujet, un livre très complet, pour une approche accessible aux non-spécialistes. Un deuxième volume est en préparation, qui sera sans doute plus spécialisé.
2. **G. Gonnet, R. Baeza-Yates :** *Handbook of Algorithms and Data Structures.* Deuxième édition, Addison-Wesley, 1991.
   Il s'agit d'un livre de référence, qui reprend tous les algorithmes ou structures de données proposés pour résoudre les problèmes de tri (interne ou externe), recherche (dans une liste ou un tableau, par hachage, par arbre, multi-dimensionnelle), files de

priorité et sélection d'un élément de rang donné, traitement de chaînes de caractères, opérations arithmétiques et multiplication de matrices.

3. **R.L. Graham, D.E. Knuth, O. Patashnik :** *Concrete Mathematics.* Addison-Wesley, 1989.
   Mathématiques discrètes ; les outils de base pour l'analyse d'algorithmes.
4. **M. Hofri :** *Analysis of Algorithms - Mathematical Methods, Computational Tools.* Oxford University Press, 1997.
5. **D.E. Knuth :** *The Art of Computer Programming : Vol. 1 : Fundamental algorithms ; Vol. 2 : Seminumerical algorithms, Vol. 3 : Searching and sorting.* Addison-Wesley, 1973.
   Le livre qui a posé les fondations de l'analyse d'algorithmes ; on y trouve beaucoup de résultats dans une présentation fouillée.
6. **W. Szpankowski :** *Average Case Analysis of Algorithms on Sequences.* Wiley and Sons, Déc. 2000,
   Ce livre traite des algorithmes sur les chaînes de caractères : algorithmes de compression, structures digitales (tries, arbres Patricia, arbres suffixes), informatique sur le génome, etc.

Mentionnons aussi le site web de *l'encyclopédie des suites entières,* de S. Plouffe et N.J. Sloane, à l'adresse http ://www.research.att.com/∼njas/sequences/index.html. Cette encyclopédie rassemble des suites combinatoires classiques et permet, à partir de la connaissance des premiers termes d'une suite, de chercher si elle peut correspondre à une suite connue.

1.5.3 Arbres aléatoires

Processus aléatoires

1. **Karlin et Taylor :** *A first course in stochastic processes.* Academic Press.
2. **N. Bouleau :** *Processus stochastiques et applications.* Hermann.
3. **Norris :** *Markov chains.* Cambridge University Press, 1998.

Arbres et processus de branchement

1. **Athreya et Ney :** *Branching processes.* Springer, 1972.
2. **H. Mahmoud :** *Evolution of random search trees.* J. Wiley and Sons, 1992.
   Arbres de recherche, arbres multidimensionnels, structures digitales.
3. **Y. Peres :** *Cours sur les arbres,* Ecole d'été de Saint-Flour, 1997. Lecture Notes 1717. Springer, 1999.

Des bibliographies plus complètes sur les arbres aléatoires sont indiquées à la fin des chapitres 5 et 6.

Chapitre 2

Hachage

2.1 Les principes de base

Le principe du hachage consiste à répartir des éléments dans une table T, en calculant, à partir de la clé de l'élément, l'adresse de la table où chercher cet élément. Les méthodes de hachage permettent de retrouver une clé, parmi un ensemble de n clés, en un nombre d'opérations constant en moyenne.

Soit m le nombre de places de la table de hachage ; les places de T sont numérotées de 0 à $m-1$. On a donc besoin tout d'abord d'une *fonction de hachage* associant à chaque clé possible une adresse dans la table (i.e. un entier entre 0 et $m-1$), de façon à éparpiller le plus possible les clés dans la table. Cependant, il est à peu près impossible d'éviter que deux clés distinctes donnent la même adresse dans la table (on parle alors de *collision*) ; il faut donc apprendre à gérer cette situation.

La fonction de hachage doit associer à chaque clé une adresse, i.e. un entier entre 0 et $m-1$. Elle doit être *déterministe* (renvoyer toujours la même adresse pour une clé donnée), *aléatoire* (disperser les clés "au hasard" dans la table), *uniforme* (il n'y a pas d'adresses privilégiées), et enfin *facile à calculer.*

2.1.1 Modèle

Dans toute la suite, nous noterons h la fonction de hachage ; la valeur $h(X)$ associée à une clé X est appelée *adresse primaire de* X ; la taille de la table où les clés seront gardées étant m, $h(X)$ est donc un entier de $[0, m-1]$. Le nombre de clés présentes dans la table sera noté n. Le nombre de cases de la table de hachage sera noté m. Un paramètre important est le *taux de remplissage ;* c'est le rapport $\alpha = n/m$ entre le nombre d'éléments à stocker et le nombre de places.

Pour l'analyse des performances du hachage, nous supposerons que les n clés sont insérées séquentiellement dans la table, et qu'elles sont toutes *distinctes.* Si x_j est la j-ième clé, notons $a_j = h(x_j)$; nous supposons que les adresses successives a_j sont indépendantes et équidistribuées, de même loi uniforme sur $\{0, ..., m-1\}$. En d'autres termes, chacune des m^n suites de hachage $a_1 a_2 ... a_n$ de $\{0, ..., m-1\}^n$ a la même probabilité $1/m^n$ d'apparaître.

2.1.2 Nombre de collisions

L'étude du nombre de collisions montre que les collisions sont inévitables, mais qu'il est très rare d'avoir beaucoup d'éléments ayant la même valeur de hachage. En effet, soit une fonction de hachage h répartissant uniformément les clés dans la table. La probabilité d'avoir au moins une collision sur n clés est

$$1 - \left(1 - \frac{1}{m}\right)\left(1 - \frac{2}{m}\right)\cdots\left(1 - \frac{n-1}{m}\right) = 1 - \prod_{j=1}^{n-1}\left(1 - \frac{j}{m}\right).$$

Exercice : On veut ranger 100 éléments dans une table, de manière à avoir une collision au plus 50% du temps. Quelle doit être la taille minimale de la table de hachage ?

La probabilité de trouver, parmi les n clés, k clés ayant une valeur de hachage spécifiée v, et donc la même adresse primaire dans une table de m adresses, suit une loi binômiale :

$$Proba(m, n, k) = \binom{n}{k}\left(\frac{1}{m}\right)^k\left(1 - \frac{1}{m}\right)^{n-k}.$$

Lorsque le nombre n de clés et la taille m de la table augmentent, tout en gardant un taux de remplissage α constant, la probabilité que k clés aient une même adresse de hachage v est approximée par une loi de Poisson, suivant l'approximation usuelle de la loi binômiale :

$$\mathrm{Proba}(m, n, k) \sim p(k; \alpha) = \frac{\alpha^k}{k!}\, e^{-\alpha}.$$

Exercice : Calculez, pour une table de $m = 100$ places remplie à moitié, puis entièrement, les probabilités exactes $Proba(m, n, k)$ pour $k = 0...10$. Quelles conséquences pratiques pouvez-vous en tirer ? Calculez également les approximations $p(k; \alpha)$. Quelle est la qualité de l'approximation ?

On peut classer les méthodes de hachage suivant la manière de résoudre les collisions : en les chaînant à l'extérieur de la table : **hachage externe**, ou à l'intérieur de la table : **hachage interne** (parmi les différentes méthodes de hachage interne, nous étudierons uniquement le hachage linéaire).

2.2 Hachage externe

Dans le hachage externe, les éléments en surnombre dans une case de la table sont rangés dans une zone de débordement, classiquement implémentée par une liste chaînée.

2.2.1 Algorithmes

L'algorithme de recherche d'une clé donnée X dans une table de hachage T est un simple parcours séquentiel de la liste chaînée associée à la case $\mathsf{T[h(X)]}$. L'insertion d'une clé X se fait de façon similaire : on commence par vérifier que la clé n'est pas déjà présente dans la table, ce qui nécessite un accès à la table et un parcours de la liste associée à la case d'adresse v= h(X) ; si on n'a pas trouvé X, on l'insère en fin de la liste associée à la case T[v]. Nous donnons ci-dessous un algorithme à cet effet ; il utilise les variables auxiliaires v pour l'adresse primaire de la clé X, prec et courant pour parcourir la liste, et nouveau pour créer une nouvelle cellule, contenant X, à ajouter à la liste. Remarquons que l'insertion du nouvel élément X se fait en fin de liste, après l'avoir parcourue toute entière pour vérifier que X n'est pas déjà présent.

```
procedure insertion ( E/S T : tableau [0..m-1] de case ; E X : element) ;
aux v : 0...m-1 ;
    prec, courant, nouveau : ↑ cellule ;
debut
      v :=h(X) ;
      creer(nouveau) ;
      nouveau↑.cle := X ;
      nouveau↑.suivant := nil ;
      si T[v] = nil alors T[v] := nouveau
      sinon
            si T[v].↑cle = X alors ERREUR
            sinon
                  prec := T[v] ;
                  courant := T[v].↑suivant ;
                  tant que courant ≠ nil faire
                          si courant↑.cle = X alors ERREUR
                          sinon prec := courant ; courant := courant↑.suivant
                          fin si
                  fin tant que
                  prec↑.suivant := nouveau
            fin si
      fin si
fin
```

Notons enfin qu'il est facile de supprimer un élément précédemment inséré, puisque cela se ramène à une recherche, puis à une suppression dans une liste chaînée.

2.2.2 Performances

Nombre moyen de comparaisons

Le coût d'une recherche, avec ou sans succès, se mesure en nombre de comparaisons entre clés. Le coût moyen C_n d'une *recherche sans succès,* i.e. d'une insertion, est égal au taux de remplissage :

$$C_n = \alpha = n/m.$$

En effet, soit λ_i la longueur de la ième liste ($0 \leq i < m$). Si on cherche la clé dans la i-ème liste, on fera exactement λ_i comparaisons. Le nombre moyen de comparaisons, pour toutes les clés possibles, i.e. pour toutes les adresses $v \in \{0, ..., m-1\}$ possibles, est donc de $(1/m) \sum_{i=1}^{m} \lambda_i = n/m$.

Le coût moyen C'_n d'une *recherche avec succès* est de

$$C'_n = \frac{n-1}{2m} + 1 = \frac{\alpha}{2} + 1 - \frac{1}{2m}.$$

En effet, soit $n(X_k)$ le nombre de comparaisons entre clés effectuées lors de l'insertion (i.e. de la recherche sans succès) de la k-ième clé X_k : le nombre de comparaisons lors de la recherche de X_k sera de $n(X_k) + 1$. Or l'insertion de X_k s'est faite en fin de liste, et nous venons de voir que le nombre moyen de comparaisons lors de l'insertion de la k-ième clé x_k est $E[n(X_k)] = C_{k-1} = (k-1)/m$. Donc, le nombre moyen de comparaisons lors de la recherche avec succès d'une clé quelconque sera

$$C'_n = \frac{1}{n}\sum_{k=1}^{n}(1 + C_{k-1}) = \frac{1}{n}\sum_{k=1}^{n}(1 + \frac{k-1}{m}) = 1 + \frac{n-1}{2m}.$$

Pour un taux de remplissage fixé, le hachage externe fait donc, en moyenne, un nombre constant de comparaisons pour retrouver une clé donnée.

Exercice : Pouvez-vous calculer la variance du coût d'une recherche sans succès ? avec succès ?

Longueur de la plus longue sous-liste

Nous avons vu que la longueur moyenne d'une liste dans une table de hachage de m cases contenant n clés est égale à $\alpha = n/m$. Cependant, la plus longue sous-liste a une longueur pouvant aller jusqu'à n (si toutes les clés se retrouvent à la même adresse). Mais c'est très peu probable : les longueurs des m listes sont des variables aléatoires équidistribuées suivant une loi de Poisson et corrélées (leur somme est égale à n), et on peut montrer que, si on prend la moyenne sur tous les tableaux possibles, la longueur de la plus longue sous-liste est en moyenne d'ordre $\log n / \log\log n$.

Exercice : Soit U la longueur de la plus longue sous-liste ; montrer que, lorsque n et m tendent vers l'infini à taux de remplissage α constant, $E[U] \sim \log n / log \log n$. Pour cela, on pourra d'abord établir que $E[U] = \sum_{k \geq 0} Proba(U > k)$. On introduira ensuite les fonctions génératrices exponentielles e^{mz}, énumérant toutes les répartitions possibles des n clés dans m listes, éventuellement vides, et $f(z)$, énumérant les répartitions des clés telles que les m listes ont chacune au plus k clés ; on montrera que $f(z) = e^{mz} - e_k(z)$, où $e_k(z) = \sum_{i \leq k} z^i/i!$ est la fonction exponentielle tronquée, et que $Proba(U > k) = [z^n]f(z)/[z^n]e^{mz}$, où $[z^m]f(z)$ est le coefficient de z^m dans la fonction $f(z)$. On terminera par une analyse asymptotique de $E[U] = (n!/m^n)\,[z^n]\{\sum_{k \geq 0}(e^{mz} - e_k(z)^m)\}$.

Une variante intéressante a été introduite par Azar, Broder, Karlin et Upfal, pour les cas où cette longueur maximale est encore trop importante : il s'agit simplement de calculer deux adresses de hachage, de regarder les longueurs des deux sous-listes correspondantes, et d'insérer la clé dans la sous-liste la plus courte (la case la moins remplie). Ceci assure que la longueur de la plus longue sous-liste restera d'ordre $\log\log n$, ce qui est largement suffisant en pratique. Par exemple, pour $n = 10^6$, on trouve $\log n/\log\log n = 5,26...$ (ce qui est déjà peu), et $\log\log n = 2,62...$

Notons que cette technique d'équilibrage de charge s'avère aussi très utile dans d'autres contextes, par exemple pour allouer dynamiquement des processeurs à des processus.

2.3 Hachage linéaire

Lorsque les collisions sont gérées dans la table, on parle de hachage interne. Un exemple classique est le *hachage linéaire,* obtenu en prenant comme suite d'adresses essayées en cas de collision les cases successives de la table : $h_{i+1}(X) = (h_i(X)+1)\ mod\ m$; $h_0(X) = h(X)$ est l'adresse primaire de X.

2.3.1 Algorithmes

Pour le hachage interne, la table de hachage est simplement représentée par un tableau de m cases, chaque case contenant une clé. Nous donnons tout d'abord l'algorithme de recherche d'une clé X dans une table T ; cet algorithme renvoie un indice place de $[0, m]$; si X est dans la table, alors place contient l'indice de la case où il se trouve ; si X n'est pas dans la table, alors place = m. On supposera qu'une valeur spéciale vide sert à reconnaître le cas où une case ne contient pas d'élément.[1] Dans la procédure donnée ci-dessous, la variable v contient les adresses successives dans la table ; la variable j compte le nombre de cases visitées, et sert à reconnaître le cas où la table est pleine (lorsqu'on a fait le tour de la table sans trouver la clé cherchée).

```
procedure recherche (E T : tableau [0..m-1] de case ; X : element ; S place : 0..m) ;
aux j,v : entier ;
debut
        v := h(X) ; j := 1 ;
        tant que ((j≤m) et (T[v] ≠ vide) et (T[v] ≠ X)) faire
                j :=j+1 ; v :=v+1 ;
                si (v=m) alors v :=1 fin si ;
        fin tant que ;
        si (T[v] = X) alors place := v sinon place := m fin si
fin ;
```

La procédure d'insertion est une variante de la procédure de recherche ; elle insère la clé X dans le tableau T, et génère une erreur si X appartenait déjà à T ou si le tableau est plein. Pour que ces deux procédures soient correctes, il faut que, dans une suite d'adresses secondaires $h_i(X)$, la première case vide rencontrée marque bien l'emplacement où insérer X, et donc qu'une case qui a été pleine ne puisse jamais redevenir vide.

2.3.2 Performances

La mesure de complexité adéquate est ici, non plus le nombre de comparaisons entre clés, mais le nombre d'examens de places de la table. On suppose ci-dessous que n et m

[1] Naturellement, il est possible d'ajouter un champ à chaque case, et d'y garder un booléen indiquant si la case est occupée, ou non.

sont “grands”, mais que leur rapport α reste fixé. On s’intéresse au cas où toutes les clés sont distinctes ; on peut alors montrer que le nombre moyen d’opérations pour rechercher une clé est asymptotiquement constant. Plus exactement, on a le résultat suivant :

1. *Le coût moyen asymptotique d’une recherche sans succès est de*

$$C_n = \frac{1}{2} + \frac{1}{2} Q_1(m,n) = \frac{1}{2}\left(1 + \frac{1}{1-\alpha}\right) + O\left(\frac{1}{n}\right),$$

   avec

$$Q_1(m,n) = \sum_{k=0}^{n} (k+1)\, \frac{n!}{m^k\,(n-k)!}.$$

2. *Pour une recherche avec succès, le coût moyen asymptotique est de*

$$C'_n = \frac{1}{2} + \frac{1}{2} Q_0(m,n-1) = \frac{1}{2}\left(1 + \frac{1}{(1-\alpha)^2}\right) + O\left(\frac{1}{n}\right),$$

   avec

$$Q_0(m,n) = \sum_{k=0}^{n} \frac{n!}{m^k\,(n-k)!}.$$

Par exemple, cela donne, lorsque la table est à moitié remplie ($\alpha = 1/2$), 1,5 accès pour une recherche avec succès et 2,5 accès pour une recherche sans succès. Lorsque la table est pleine à 90% ($\alpha = 9/10$), les coûts d’accès deviennent respectivement 5,5 et 50,5. La dégradation des performances, lorsque le taux de remplissage augmente, s’explique intuitivement par le fait que les listes de débordement de diverses cases fusionnent entre elles.

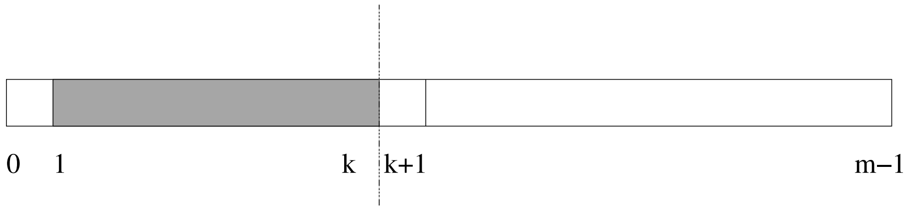
L’analyse des performances du hachage linéaire (recherche avec ou sans succès) est due à D. Knuth en 1962, et est considérée comme marquant le début de l’analyse d’algorithmes. Nous la présentons ci-dessous.

Définissons les quantités
- $f_{n,m}$: nombre de suites de hachage distribuant n clés dans m cases, en laissant la case T[0] vide,
- $g_{n,m,k}$: nombre de suites de hachage telles que les cases T[0] et T[k+1] soient vides, et les cases T[1], ... , T[k] contiennent chacune une clé,
- $p_{n,m,j}$: nombre de suites de hachage telles que l’insertion de la $(n+1)$-ième clé nécessite de tester $j+1$ cases.

 – Nous commençons par calculer $f_{n,m}$: remarquons que, par symétrie, chaque case a la même probabilité d’être vide. Il y a n clés, donc $m-n$ cases vides, et la case T[0], par exemple, est vide avec une probabilité $(m-n)/m$. Comme il y a m^n suites de hachage au total, nous obtenons

$$f_{n,m} = \frac{m-n}{m}\, m^n = (m-n)\, m^{n-1}.$$

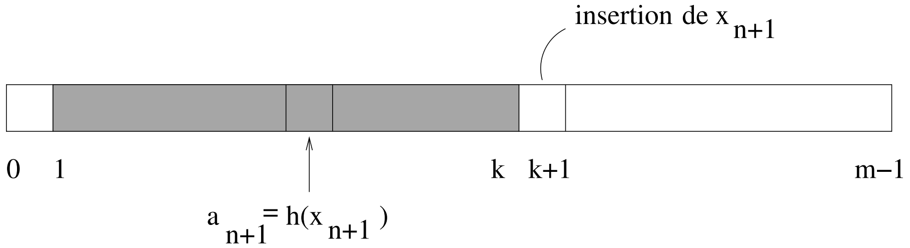
 – Nous calculons maintenant $g_{n,m,k}$:



Il y a $\binom{n}{k}$ manières de choisir les k clés qui sont dans les cases T[1...k]. En divisant la table de hachage en deux sous-tables, T[0...k], dont la première case est vide (et il ne peut y avoir qu'une seule case vide, puisque ici le nombre de clés et le nombre de cases diffèrent de 1), et T[k+1...m-1], dont $m-n+1$ cases sont vides, y compris la première case en T[k+1], nous voyons que

$$g_{n,m,k} = \binom{n}{k} f_{k,k+1}\, f_{n-k,m-k-1}.$$

– Nous pouvons maintenant calculer $p_{n,m,j}$. Soit $a_{n+1} = h(x_{n+1})$ l'adresse primaire de la $n+1$-ième clé ; si le nombre d'accès à une case de la table, pour insérer la clé x_{n+1}, vaut $j+1$, c'est que la case $\mathsf{T}[a_{n+1}]$ était occupée, ainsi que les $j-1$ cases suivantes $\mathsf{T}[a_{n+1}+1]$, ..., $\mathsf{T}[a_{n+1}+j-1]$, et que la case $\mathsf{T}[a_{n+1}+j]$ était libre. Soit $k \geq j$ la longueur du bloc contenant la case $\mathsf{T}[a_{n+1}]$; par symétrie on peut supposer que la case T[0] est libre, et que ce bloc commence en T[1] :



Pour un k donné,il y a $g_{n,m,k}$ telles configurations ; le nombre total de configurations s'obtient en sommant sur tous les k possibles :

$$p_{n,m,j} = \sum_{k=j}^{n} g_{n,m,k}.$$

– Passons maintenant au coût moyen d'une recherche sans succès :

$$C_n = \frac{1}{m^n} \sum_{j\geq 0} (j+1) p_{n,m,j}.$$

Après quelques calculs algébriques, nous trouvons que

$$C_n = \frac{1}{2}\left(1 + Q_0(m, n-1)\right),$$

où Q_0 est une fonction dûe à Ramanujan :

$$Q_0(m,n) = \sum_{i=0}^{n} \frac{n!}{m^i (n-i)!}.$$

L'étude asymptotique termine la preuve.

Exercice : Démontrer l'égalité $C_n = (1+Q_0(m, n-1))/2$. On pourra introduire les sommes

$$S^i_{n,m,t} = \frac{m-n-t}{m^n} \sum_k \binom{n}{k} (k+t)^{k-1+i} (m-k-t)^{n-k-1},$$

qu'on calculera en établissant une relation de récurrence entre $S^i_{n,m,t}$, $S^i_{n-1,m,t+1}$ et $S^{i-1}_{n,m,t}$, puis en calculant $S^0_{n,m,t}$ grâce à l'identité (qu'on pourra démontrer)

$$(\alpha+\beta)(n+\alpha+\beta) = \alpha\,\beta \sum_k \binom{n}{k} (k+\alpha)^{k-1} (n-k-\beta)^{n-k-1}.$$

Exercice : En utilisant la relation $C'_n = (1/n) \sum_{k=1}^{n} C_{k-1}$, calculez le coût moyen (exact puis asymptotique) d'une recherche avec succès.

Exercice : Avec les techniques utilisées ci-dessus, pouvez-vous calculer la variance du coût d'une recherche sans succès ? avec succès ?

Chapitre 3

Algorithmes de tri et permutations aléatoires

Nous étudions ici des algorithmes permettant de trier un ensemble de n éléments, chaque élément étant composé d'une clé et d'autres champs. Nous réduirons un élément à sa clé, que nous supposons appartenir à un ensemble totalement ordonné.

Nous nous intéressons dans ce cours aux algorithmes de tri procédant uniquement par comparaison entre clés et par transferts de clés, et donc valables pour tous les types de clés. On peut montrer qu'un tel algorithme ne peut pas faire moins de $\lg(n!) = \Theta(n \lg n)$ comparaisons, en moyenne, pour trier n clés. Cependant, si on possède des informations supplémentaires sur les clés à trier, on peut obtenir des méthodes de tri de complexité linéaire (tri par paquets, tri radix...), qui ont donc un intérêt pratique évident. De telles méthodes nécessitent cependant de la place mémoire auxiliaire, et ne sont pas valables pour tous les types de clés ; nous ne les présenterons pas ici.

Les algorithmes étudiés sont : le tri à bulles, le tri par sélection, le tri par insertion et le tri rapide. Pour chacun d'eux, nous donnerons une version de leur code, afin de bien mettre en évidence les instructions influant sur leur coût.

3.1 Le modèle des permutations

Les algorithmes présentés ci-dessous travaillent sur un tableau de n clés, en procédant par comparaisons. Nous supposons que les clés sont dans un tableau T[1..n].[1] Les tableaux sont à trier en ordre croissant : en sortie du tri, T[1] $\leq$ ... $\leq$ T[n].

Lors de l'analyse du temps d'exécution d'un algorithme de tri, on se rend facilement compte que les valeurs exactes des éléments n'interviennent pas, mais seul leur ordre respectif. En supposant que *les clés sont toutes distinctes* (hypothèse que nous ferons ici), on peut se ramener au cas où les clés sont les n premiers entiers $1, 2, \ldots n$. Une instance d'un tableau T est alors vue comme une *permutation* τ sur l'ensemble $\{1, \ldots n\}$, i.e. une bijection de $\{1, \ldots n\}$ sur lui-même, associant à l'entier i la valeur $\tau(i)$.

[1] Le cas échéant, ce tableau sera étendu pour inclure une sentinelle en T[0], ou en T[n+1], suivant les besoins de l'algorithme présenté.

On supposera que toutes les permutations sont équiprobables ; chacune apparaîtra donc avec une probabilité $1/n!$. On verra que l'analyse des algorithmes ci-dessous conduit à identifier certains paramètres des permutations (nombre d'inversions, nombre de cycles, nombre de maxima provisoires de gauche à droite), qui déterminent le temps d'exécution, i.e. le coût, des algorithmes.

3.2 Tri à bulles

```
procedure tri-bulles (E/S T : tableau [1..n] de element) ;
aux i,j : entier ;
debut
    pour i de 1 a n-1 faire
        pour j de n a i+1 faire
            si T[j] < T[j-1] alors echanger(T[j],T[j-1]) finsi
        fin pour
    fin pour
fin ;
```

Il est possible d'analyser cet algorithme en affectant un coût (temps d'exécution) à chaque opération de base : affectation à un entier, à un élément du tableau T, incrémentation ou décrémentation d'entier, comparaison de deux clés, ou de deux entiers ; ces temps dépendent du système informatique sur lequel s'exécute l'algorithme, et peuvent être mesurés. Une évaluation du nombre de fois où chaque opération est effectuée nous permettra alors de connaître précisément le temps pris par l'algorithme sur un jeu de données.

En pratique, on cherchera à simplifier l'analyse tout en gardant ses caractéristiques essentielles. On ne considérera donc pas toutes les opérations effectuées par l'algorithme, tout d'abord parce que le nombre de répétitions de certaines opérations est facilement accessible, et on se ramènera à ne considérer que quelques opérations dont le coût est considéré comme primordial (comparer deux clés peut prendre notablement plus de temps que comparer deux entiers, ou tester un booléen). Ainsi, pour le tri à bulles, le nombre d'opérations sur les indices i et j est lié au nombre d'itérations des boucles interne et externe ; les paramètres déterminant le coût seront le nombre N_c de *comparaisons entre clés* et le nombre N_e d'*échanges de clés*. Soient C_e et C_c les coûts respectifs d'un échange et d'une comparaison de clés ; le coût global de l'algorithme peut s'écrire $C_e\, N_e + C_c\, N_c$.

Le nombre de comparaisons entre clés faites par le tri à bulles est constant, quel que soit le tableau : au i-ème passage dans la boucle externe, on fait $n - i$ comparaisons, soit en tout $N_c = \sum_{i=1}^{n-1}(n-i) = n(n-1)/2$ comparaisons.

Sur un tableau quelconque, le nombre d'échanges de clés est égal au nombre de paires (i, j) telles que $1 \leq i < j \leq n$ et que T[i] > T[j], c'est-à-dire au *nombre d'inversions* de la permutation associée au tableau T[1...n] que l'on trie. Le temps pris par l'algorithme va donc dépendre du tableau fourni en entrée. Au pire, le nombre d'inversions de T vaut $n(n-1)/2$: tableau trié en ordre inverse ; au mieux il sera nul : tableau déjà trié.

Il est pertinent de s'intéresser au temps pris par l'algorithme sur un tableau choisi "au hasard", i.e. au *temps moyen* d'exécution, qui sera donc lié au nombre moyen d'inversions d'une permutation aléatoire. Cette dernière valeur vaut $n(n-1)/4$; cela peut se voir simplement en raisonnant à la fois sur un tableau T et sur son tableau miroir $\tilde{T}$, défini par : $\tilde{T}[i] = T[n-i+1]$: chaque inversion possible apparaît, soit dans le tableau d'origine, soit dans le tableau miroir, et il y a $n(n-1)/2$ inversions possibles...

Ce raisonnement nous donne le nombre moyen d'inversions, mais pas plus. Une analyse plus mathématique du nombre d'inversions nous permet de retrouver cette moyenne, et d'obtenir la variance. Notons X le nombre total d'inversions, et X_j le nombre de ces inversions qui proviennent du placement du j-ème élément dans la permutation portant sur les $j-1$ premières valeurs : $X = X_1 + X_2 + ... + X_n$. Naturellement, $X_1 = 0$; pour $j > 1$, X_j prend les valeurs $0, 1, ..., j-1$ avec une probabilité uniforme $1/j$: le rang où on insère j parmi la permutation sur $\{1, 2, ..., j-1\}$ suffit à déterminer le nombre d'inversions (i, j) avec $i < j$ et $\tau(i) > \tau(j)$.

Le placement du j-ème terme se fait indépendamment des choix faits pour les $j-1$ termes précédents, et le nombre total d'inversions X est donc la somme de n variables indépendantes Comme $E[X_j] = (j-1)/2$ et $\sigma^2(X_j) = (j^2-1)/12$, on obtient

$$\begin{aligned} E[X] &= \textstyle\sum_j E[X_j] &&= \frac{n(n-1)}{4}; \\ \sigma^2(X) &= \sigma^2(X_j) &&= \frac{n(n-1)(2n+5)}{72}. \end{aligned}$$

Les variables aléatoires X_j ne sont pas équidistribuées (les lois uniformes ne portent pas sur le même domaine), et il n'est donc pas possible d'appliquer la version classique du théorème central limite à X. Il est cependant possible de montrer que sa distribution limite est normale.

Le tri à bulles a donc une complexité quadratique : il fait de l'ordre de n^2 opérations (comparaisons et échanges de clés), en moyenne et au pire.

<u>Exercice :</u> Etudier le nombre d'inversions en utilisant la fonction génératrice $f_n(u)$ définie par :

$$f_n(u) = \sum_k \frac{\alpha_{n,k}}{n!} u^k,$$

où $\alpha_{n,k}$ est le nombre de permutations sur n éléments ayant k inversions. Vous trouverez une expression de $f_n(u)$ sous forme de produit fini de polynômes élémentaires, puis retrouverez la moyenne et la variance du nombre d'inversions.

3.3 Tri par insertion

Le principe du tri par insertion est le suivant :

Pour i de 2 à n, insérer le i-ème élément T[i] dans T[1...i-1] (qui est déjà trié !).

Il y a plusieurs manières de rechercher la place de T[i] ; nous étudierons ici l'algorithme simple dans lequel la recherche de la place de T[i] se fait séquentiellement à partir de la droite, et en déplaçant au fur et à mesure les éléments T[j] qui sont plus grands que l'élément à placer T[i]. Dans ce cas, il est souhaitable d'insérer une *sentinelle* en T[0], pour arrêter la recherche de la place de T[i], lorsqu'il est plus petit que tous ceux qui l'ont précédé. Ceci conduit à l'algorithme suivant :

```
procedure tri-insertion (E/S T : tableau [0..n] de element);
aux i,j : entier;
    X : element;
debut
        pour i de 2 a n faire
                j := i-1; X := T[i];
                tant que (T[j] > X) faire T[j+1] := T[j]; j := j-1 fin tant que
                T[j+1] := X
        fin pour
fin;
```

Le temps d'exécution de cet algorithme est là encore fonction d'un paramètre de la permutation aléatoire correspondant à l'ordre initial des clés (en supposant ces clés toutes distinctes) : le nombre d'inversions de cette permutation, $Inv(T)$.

<u>Exercice :</u> Analyse de la complexité du tri par insertion.
1. Pour un tableau T donné, exprimez le nombre de comparaisons entre clés faites par le tri par insertion, $N_c(T)$, et le nombre de transferts, $N_t(T)$, en fonction du nombre d'inversions de T.
2. Quell est la performance du tri par insertion dans le cas le pire ? en moyenne ?
3. Supposons que le tri par insertion soit appelé sur la concaténation de p listes L_j, chacune de taille moyenne m ($mp = n$) ; supposons de plus que les éléments de la liste L_j soient tous inférieurs aux éléments de la liste de rang L_{j+1} ($1 \leq j < p$), et que la concaténation respecte l'ordre des listes. Quelle est la complexité au pire du tri par insertion dans ce cas ?
4. En vous inspirant de ce résultat, pouvez-vous déterminer des situations dans lesquelles le tri par insertion aura des performances intéressantes ?

3.4 Tri par sélection

Le principe du tri par sélection est très simple :

> *On cherche le plus petit élément et on l'amène à sa place en* $T[1]$ *; on cherche le deuxième plus petit élément et on l'amène en* $T[2]$ *; etc...*

La recherche du plus petit élément de $T[1 \ldots n]$, puis de $T[i \ldots n]$, se fera séquentiellement. Nous donnons ci-dessous une version itérative du tri par sélection, dans laquelle l'indice i de la boucle externe indique le rang de l'élément que l'on recherche, l'indice j varie entre 1 et n, et l'indice k pointe sur l'élément minimal trouvé jusqu'ici : $T[k] = min\{T[i], T[i+1], \ldots, T[j]\}$.

```
procedure tri-selection (E/S T : tableau [1 ... n] de element);
```

```
aux i,j,k : entier;
debut
    pour i de 1 a n-1 faire
        j := i;
        pour k de i+1 a n faire
            si T[k] < T[j] alors j := k finsi
        fin pour
        echanger (T[i], T[j]);
    fin pour
fin
```

Le tri par sélection est progressif (la i-ème itération de la boucle fournit le i-ème élément), mais non stable. C'est un tri sur place, de complexité quadratique en moyenne et au pire. En effet, il faut $n-i$ comparaisons entre clés pour trouver le minimum de {T[i] ,..., T[n]}, et on répète ceci pour i variant de 1 à n ; au total il faut donc $n(n-1)/2$ comparaisons, quel que soit le tableau initial. Quant au nombre d'échanges de clés, il est égal à $n-1$ (il y a un échange à chacune des $n-1$ itérations de la boucle externe).

Intéressons-nous maintenant au nombre d'affectations j :=k : c'est le nombre de *minima provisoires de gauche à droite* (MPGD) de la permutation associée au tableau T[2]. Ce nombre peut être mis en relation avec le nombre de cycles d'une permutation, à l'aide de l'écriture canonique d'une permutation : Soit une permutation τ ayant k cycles ; on ordonne chaque cycle en commençant par le plus petit élément, puis on ordonne les cycles entre eux, par numéro de premier élément décroissant. On obtient alors une permutation $\tau^{'}$ ayant k minima provisoires de gauche à droite. Par exemple, soit la permutation

$$\tau = (3\ 7\ 2\ 5\ 4\ 6\ 1).$$

On a $\tau(1) = 3$, $\tau(2) = 7$, etc. Les cycles sont $(1\ 3\ 2\ 7)$, $(4\ 5)$ et (6), et la permutation associée, qui a pour MPGD 6, 4 et 1, est

$$\tau^{'} = (6 \mid 4\ 5 \mid 1\ 3\ 2\ 7).$$

Il est aisé de se rendre compte que la transformation $\tau \to \tau^{'}$ est bijective ; des résultats sur le nombre de cycles d'une permutation se traduiront donc en résultats analogues sur le nombre de MPGD. Or le nombre de permutations de $\{1, \ldots, n\}$ ayant k cycles est bien connu en combinatoire ; il s'agit d'un *nombre de Stirling de première espèce,* noté $s_{n,k}$. De tels nombres satisfont une équation de récurrence : Lorsqu'on ajoute un nouvel élément x, soit il crée un cycle de longueur 1, et il y avait donc $k-1$ cycles sur les $n-1$ éléments, soit il va dans un cycle préexistant, de longueur p, et il y a alors p façons de l'insérer ; au total il y a donc $n-1$ façons d'insérer x dans un des k cycles. On obtient donc

$$s_{n,k} = s_{n-1,k-1} + (n-1)\ s_{n-1,k}, \tag{3.1}$$

avec les conditions aux limites $s_{n,k} = 0$ pour $n = 0$ ou $k = 0$ (sauf $s_{0,0} = 1$), $s_{n,n} = 1$ et $s_{n,k} = 0$ pour $k > n$.

[2]En probabilités, de telles variables aléatoires sont appelées des “records”.

Le nombre moyen de MPGD d'une permutation sera donc égal au nombre moyen de cycles d'une permutation,

$$X_n := \frac{1}{n!} \sum_{\tau\ perm.} Nombre - cycles(\tau) = \frac{1}{n!} \sum_{k=1}^{n} k\ s_{n,k}.$$

En fait, il y a identité de distribution entre le nombre de cycles et le nombre de MPGD.

L'équation de récurrence (3.1) sur les nombre de Stirling conduit, après quelques calculs simples, à une équation sur X_n : $X_n = X_{n-1} + 1/n$, avec $X_1 = 1$, d'où l'on déduit que

$$X_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = H_n \sim \log n.$$

On sait aussi que les nombres de Stirling de première espèce suivent asymptotiquement une loi normale.

3.5 Tri rapide

Le tri rapide dépend de la structuration de l'ensemble des clés par rapport à une clé spéciale, utilisée comme *pivot*. Le principe est le suivant :

> *Mettre à sa place définitive la clé X choisie comme pivot, de telle sorte que les clés inférieures ou égales à X se retrouvent à sa gauche, et que celles strictement supérieures à X se retrouvent à sa droite ; puis trier récursivement les deux sous-listes ainsi obtenues.*

Dans la procédure ci-dessous, le tableau T a une case supplémentaire T[n+1] contenant une sentinelle plus grande que toutes les clés à trier.

```
procedure tri-rapide (E/S T : tableau [1..n+1] de element; E droite, gauche : entier);
aux k : entier;
debut
        si (gauche < droite) alors
           placer (T, gauche, droite, k);
           tri-rapide (T, gauche, k-1);
           tri-rapide (T, k+1, droite)
fin;
```

La procédure de placement doit donc trouver la place k du pivot, bien sûr sans trier le tableau ! Nous choisissons comme pivot T[gauche]. Lors de la recherche de la place du pivot, il faut prévoir les cas où ce pivot est, soit plus petit que tous les éléments déjà rencontrés, soit plus grand. Pour éviter les accès à d'autres cases que celles concernées par la procédure de partition, voire les accès à des cases non existantes, il faut donc prévoir des sentinelles, à gauche et à droite du tableau. En fait, le pivot, choisi en T[gauche], joue le rôle de sentinelle gauche. Quant à la sentinelle droite, elle est en général présente lors des appels récursifs : c'est un ancien pivot ! Le seul cas posant réellement problème est celui où $j = n$, i.e. où on travaille sur la partie droite du tableau. Pour éviter de vérifier que l'indice bas reste inférieur ou égal à j, nous utilisons une sentinelle en T[n+1], plus grande que tous les éléments susceptibles d'être présents dans le tableau.

```
procedure placer (E/S T : tableau [1..n+1] de element ; E gauche, droite : entier ;
                  S k : entier) ;
aux bas, haut : entier ;
debut
    bas := gauche +1 ; haut := droite ;
    tant que (bas ≤ haut) faire
        tant que (T[bas] ≤ T[gauche]) faire bas := bas+1 fin tant que ;
        tant que (T[haut] > T[gauche]) faire haut := haut-1 fin tant que ;
        si (bas < haut) alors
          echanger (T[bas], T[haut]) ;
          bas := bas+1 ; haut := haut-1
        fin si
    fin tant que
    echanger (T[gauche], T[haut]) ;
    k := haut
fin ;
```

Etude de la complexité moyenne :
Soient $c(T)$ et $p(T)$ les coûts respectifs de l'algorithme de tri et de la procédure de partition sur un tableau T de n clés, mesurés en nombre de comparaisons entre clés. Soit c_n le coût moyen de l'algorithme de tri sur n clés : $c_n = \sum_T c(T)\, proba(T)$; soit aussi $p_n = \sum_T p(T)\, proba(T)$ le coût moyen de l'appel de la partition sur $T[1...n]$, avant les appels récursifs du tri rapide. Les valeurs initiales sont $c_0 = c_1 = p_0 = p_1 = 0$, et $c_2 = p_2 = 3$ (voir les deux possibilités sur un tableau de 2 éléments).

Soit un tableau T, et soit k la place finale du pivot. Notons $T^{'}$ et $T^{''}$ les sous-tableaux $T[1...k-1]$ et $T[k+1...n]$ après placement du pivot :

$$c(T) = p(T) + c(T^{'}) + c(T^{''}).$$

Le tableau de départ T est choisi avec une probabilité uniforme parmi les $n!$ tableaux possibles (les clés sont supposées toutes distinctes). Définissons la variable aléatoire $U(T)$, à valeurs dans $\{1, 2, ..., n\}$, égale à l'indice k donnant la place finale du pivot. Dans le modèle des permutations aléatoires, U est une v.a. équidistribuée. Le pivot va donc à la place k avec une probabilité $1/n$, et

$$\begin{aligned} c_n &= \sum_{|T|=n} Proba(T)p(T) + \sum_T Proba(T)(c(T^{'}) + c(T^{''})) \\ &= p_n + \sum_{k=1}^{n} \sum_{T:\, le\, pivot\, va\, en\, T[k]} Proba(T)(c(T^{'}) + c(T^{''})). \end{aligned}$$

Les tableaux $T^{'}$ et $T^{''}$ suivent eux aussi une loi uniforme, respectivement parmi les permutations sur $k-1$ éléments, et parmi celles sur $n-k$ éléments, et on obtient donc une relation de récurrence sur les c_p, valable a priori pour $n \geq 3$:

$$c_n = p_n + \frac{1}{n}\sum_{k=1}^{n}(c_{k-1} + c_{n-k}) = p_n + \frac{2}{n}\sum_{k=0}^{n-1} c_k.$$

Il est aisé de vérifier que cette dernière relation est aussi valide pour $n = 2$. On obtient donc $n(c_n - p_n) = 2\sum_{k=1}^{n-1} c_k$, qui conduit par soustraction à l'égalité suivante, valide pour $n \geq 3$:

$$n(c_n - p_n) - (n-1)(c_{n-1} - p_{n-1}) = 2c_{n-1}.$$

Nous pouvons procéder à une analyse sans connaître la loi exacte de p_n, à partir de l'encadrement $n \leq p_n \leq n+1$. En prenant p_n toujours égal à $n+1$, nous obtiendrons une borne supérieure c_n^+, vérifiant :

$$nc_n^+ = (n+1)c_{n-1}^+ + 2n.$$

Ceci se résoud aisément :

$$\frac{c_n^+}{n+1} = \frac{c_{n-1}^+}{n} + \frac{2}{n+1} \qquad\qquad (n \geq 3),$$

d'où

$$\frac{c_n^+}{n+1} = 2H_{n+1} - \frac{8}{3}$$

et finalement

$$c_n^+ = 2(n+1)H_n - \frac{8n+2}{3} = 2n\log n - 1.5122...n + o(n).$$

Le même calcul, conduit avec la borne inférieure c_n^- obtenue pour $p_n = n$, donne $c_n^- = 2(n+1)H_{n+1} + O(1)$, donc le même terme principal, qui est aussi celui de c_n.

Le tri rapide a donc un comportement moyen, sur n clés, en $n\log n$. Dans le cas le pire, le tri rapide fait de l'ordre de n^2 opérations ; ce cas est atteint, par exemple, sur des listes triées ou triées en ordre inverse. On peut montrer que la variance du nombre de comparaisons fait par le tri rapide est $\sigma^2 \sim n\sqrt{7 - 2\pi^2/3}$; la distribution limite existe, mais n'est pas une des distributions classiques.

Améliorations : Il existe des variantes permettant d'améliorer encore le tri rapide "de base" : choix du pivot comme médiane de trois éléments (ceci assure qu'aucune des deux sous-listes sur lesquelles on appelle récursivement le tri n'est vide), appel à une autre méthode de tri pour des listes de petite taille (on utilise en général le tri par insertion ; on a vu que dans ce cas le nombre d'inversions est d'ordre au plus linéaire, et le tri par insertion a donc une complexité linéaire).

Remarque : le tri rapide est le tri standard du système Unix.

Le tri rapide est lié aux arbres binaires de recherche, dont nous parlerons dans un prochain chapitre : l'ensemble des clés (à trier ou à stocker) est structuré de la même manière, avec une partition par rapport à une clé donnée (le pivot ou la racine de l'arbre), et il y a donc analogie des paramètres.

Chapitre 4

Arbres binaires et familles simples d'arbres

4.1 Arbres de Catalan

Les *arbres binaires,* aussi appelés *arbres de Catalan,* satisfont une définition récursive ($\emptyset$ désigne l'arbre vide et $\bullet$ la racine d'un arbre non vide) :

$$\mathcal{B} = \emptyset + (\bullet, \mathcal{B}, \mathcal{B}). \tag{4.1}$$

En dehors de leur utilité propre, les arbres binaires servent aussi à représenter les arbres planaires d'arité quelconque, par la bijection "fils aîné - frère droit".

On parlera indifféremment d'arbre à n nœuds, ou n sommets, ou de taille n ; lorsqu'aucune précision n'est donnée, il s'agit du nombre total de sommets de l'arbre (nœuds internes et feuilles). Nous considérerons toujours des arbres de taille finie.

Tout arbre à n nœuds peut être mis en bijection avec un arbre complet (i.e. où tout nœud interne a deux fils), en ajoutant $n+1$ feuilles ; l'arbre complet obtenu a n nœuds internes et $2n+1$ sommets au total.

4.1.1 Dénombrement

La première question qui se pose est de dénombrer le nombre b_n d'arbres à n sommets. On obtient facilement $b_0 = 1$, $b_1 = 1$, $b_2 = 2$, $b_3 = 5$, $b_4 = 14$, etc. En décomposant l'ensemble des arbres de taille $n \geq 1$ suivant la taille k du sous-arbre gauche, qui varie de 0 à $n-1$, on obtient une relation de récurrence sur la suite $(b_n)_{n\geq 0}$:

$$b_n = \sum_{k=0}^{n-1} b_k \, b_{n-k-1}. \tag{4.2}$$

La technique classique pour résoudre cette relation est de passer par la fonction

$$b(z) := \sum_{n\geq 0} b_n \, z^n,$$

souvent appelée la *série génératrice* de la suite (b_n). Nous travaillerons sur ces objets, d'une part en tant que séries formelles, d'autre part comme séries entières, définissant donc une fonction d'une variable complexe, à laquelle nous pourrons appliquer la théorie des fonctions holonomes. Ainsi, en considérant $b(z)$ comme une série formelle, prendre son carré et regrouper les termes de même degré en z nous conduit à

$$\begin{aligned} b^2(z) &= b_0^2 + (b_0b_1 + b_1b_0)z + (b_0b_2 + b_1^2 + b_2b_0)z^2 \\ &\quad +(b_0b_3 + b_1b_2 + b_2b_1 + b_3b_0)z^3 + \dots \end{aligned}$$

et l'équation (4.2) permet de reconnaître que

$$b^2(z) = \sum_{n\geq 0} b_{n+1}z^n.$$

Des manipulations algébriques élémentaires conduisent à l'équation

$$1 + zb^2(z) = b(z) \tag{4.3}$$

qui se résoud aisément (le choix entre les deux racines se fait en tenant compte de $b(0) = b_0 = 1$) :

$$b(z) = \frac{1}{2z}(1 - \sqrt{1-4z}).$$

Il est aussi possible d'obtenir directement l'équation (4.3) à partir de la définition (4.1). Pour obtenir b_n, le coefficient de z^n dans $b(z)$, il suffit alors d'utiliser la formule de Taylor (la notation $[z^n]f(z)$ désigne le coefficient de z^n dans la fonction $f(z)$, considérée en tant que série génératrice). Pour le cas qui nous intéresse, nous obtenons en appliquant la formule de Taylor (voir ci-dessous)

$$b_n = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n+1}\binom{2n}{n}.$$

Les nombres ci-dessus sont les *nombres de Catalan,* classiquement notés C_n. Leur comportement asymptotique pour $n \to +\infty$ s'obtient à l'aide de la formule de Stirling : $n! \sim \sqrt{2\pi n}\,(n/e)^n$, et on trouve

$$C_n \sim \frac{4^n}{n\sqrt{\pi n}}.$$

4.1.2 Asymptotique de coefficients et formule de Taylor

Nous avons

$$[z^n](1+z)^\alpha = \frac{1}{n!}\alpha(\alpha-1)\dots(\alpha-n+1) \qquad (\alpha \in \mathcal{R}).$$

Lorsque α est entier positif, on retrouve la formule du binôme : $[z^n](1+z)^\alpha = \binom{\alpha}{n}$. Notons aussi deux formules fréquemment rencontrées :

$$\begin{aligned} [z^n]\sqrt{1-z} &= -\frac{(2n-2)!}{2^{2n-1}n!(n-1)!} = -\frac{C_{n-1}}{2^{2n-1}} \sim \frac{-1}{2n\sqrt{\pi n}}\,; \\ [z^n]\frac{1}{\sqrt{1-z}} &= \frac{1}{4^n}\binom{2n}{n} = \frac{(n+1)C_n}{4^n} \sim \frac{1}{\sqrt{\pi n}}. \end{aligned}$$

4.2 Longueur de cheminement d'un arbre binaire

La *longueur de cheminement* $lc(\mathcal{B})$ d'un arbre binaire $\mathcal{B}$ est définie comme la somme des profondeurs des différents sommets de $\mathcal{B}$. Nous conviendrons que la profondeur de la racine d'un arbre est égale à 0.

La longueur de cheminement intervient dans l'analyse d'algorithmes faisant intervenir des parcours d'arbres ; ainsi, la profondeur moyenne d'un nœud dans un arbre $\mathcal{B}$ à n sommets est $lc(\mathcal{B})/n$.

On décompose souvent la longueur de cheminement en deux parties : la longueur de cheminement *interne,* $lci(\mathcal{B})$, obtenue en sommant les profondeurs des sommets internes de $\mathcal{B}$, et la longueur de cheminement *externe,* $lce(\mathcal{B})$, qui est la somme des profondeurs des feuilles : $lc(\mathcal{B}) = lci(\mathcal{B}) + lce(\mathcal{B})$. Lorsque $\mathcal{B}$ est un arbre binaire *complet*[1] de n sommets internes et $n+1$ feuilles, les longueurs de cheminement interne et externe sont reliées très simplement :

$$lce(\mathcal{B}) = lci(\mathcal{B}) + 2n. \tag{4.4}$$

Cette relation se montre facilement, par récurrence sur le nombre n de sommets internes. Le cas de base est obtenu pour $n = 1$: il y a un seul arbre complet ayant un nœud interne et deux feuilles ; ses longueurs de cheminement interne et externe sont respectivement 0 et 2 et la relation (4.4) est bien vérifiée ; le passage d'un arbre complet à n sommets internes à un arbre complet à $n+1$ sommets internes se fait en remplaçant une feuille par un nœud interne, et les deux feuilles associées, et on vérifie aisément qu'une telle transformation conserve la relation (4.4).

Revenons à la longueur de cheminement d'un arbre binaire $\mathcal{B}$ à n sommets. Il est facile d'en donner un encadrement :

$$\sum_{1 \le k \le n} [\log_2 k] \le lc(\mathcal{B}) \le \frac{n(n-1)}{2}.$$

La borne inférieure correspond à un arbre le plus compact possible, ayant ses feuilles sur les deux derniers niveaux seulement, et la borne supérieure à un arbre filiforme. Asymptotiquement, la longueur de cheminement d'un arbre binaire est donc d'ordre au moins $n \log n$, et au plus n^2. En fait, l'ordre exact est $n\sqrt{n}$. Pour le voir, intéressons-nous à la somme des longueurs de cheminement, cumulées sur tous les arbres de taille n :

$$l_n := \sum_{taille(\mathcal{B})=n} lc(\mathcal{B}).$$

Cette équation, combinée à l'équation de récurrence reliant la longueur de cheminement d'un arbre $\mathcal{B} = (\bullet, \mathcal{G}, \mathcal{D})$ à celles de ses sous-arbres gauche $\mathcal{G}$ et droit $\mathcal{D}$:

$$lc(\mathcal{B}) = taille(\mathcal{G}) + taille(\mathcal{D}) + lc(\mathcal{G}) + lc(\mathcal{D}), \tag{4.5}$$

[1] Un arbre complet est un arbre dans lequel tout sommet interne a deux fils ; il n'y a donc pas de nœud simple.

donne après quelques calculs

$$l_n = 2\sum_{k=0}^{n-1} (k\, b_k + l_k)\, b_{n-k-1}.$$

Pour résoudre cette récurrence, comme plus haut, on utilise la série génératrice associée : $l(z) := \sum_{n\geq 0} l_n z^n$; on trouve qu'elle satisfait l'équation

$$l(z) = 2z^2\, b(z)\, b^{'}(z) + 2z\, l(z)\, b(z),$$

et donc que

$$l(z) = \frac{2z^2\, b(z)\, b^{'}(z)}{1-2zb(z)} = \frac{1}{1-4z} - \frac{1-z}{z\sqrt{1-4z}} + \frac{1}{z}.$$

Comme précédemment, on obtient l_n en appliquant la formule de Taylor, cette fois avec $\alpha = -1/2$:

$$l_n = 4^n + \binom{2n}{n} - \binom{2n+2}{n+1} = 4^n - \frac{3n+1}{n+1}\binom{2n}{n} = 4^n - (3n+1)C_n.$$

Pour trouver la longueur de cheminement moyenne d'un arbre de taille n, il suffit de diviser par le nombre C_n de tels arbres ; on trouve que la valeur exacte est $(4^n/C_n) - (3n+1)$, d'ordre asymptotique $n\sqrt{\pi n}$ (plus précisément, $n\sqrt{\pi\, n} - 3n + (9/8)\sqrt{\pi n} - 1 + o(n^{-1/2})$). La profondeur moyenne d'un sommet de $\mathcal{B}$ est $lc(\mathcal{B})/n$; dans un arbre pris au hasard, elle est asymptotiquement égale à $\sqrt{\pi n}$.

Si nous voulons maintenant avoir une idée de la variance de la longueur de cheminement, nous pourrons utiliser la *fonction génératrice bivariée* $L(u,z) := \sum_{\mathcal{B}} u^{lc(\mathcal{B})} z^{|\mathcal{B}|}$: la probabilité qu'un arbre ait une longueur de cheminement égale à k, sachant que sa taille est n, vaut $[u^k z^n]L(u,z)/[z^n]L(1,z)$. On montre facilement que la fonction $L(u,z)$ vérifie l'équation fonctionnelle

$$L(u,z) = 1 + z\, L(u, uz)^2. \tag{4.6}$$

En prenant $u = 1$, nous retrouvons l'équation définissant la série génératrice des nombres de Catalan, i.e. $L(1,z) = b(z)$, comme on pouvait s'y attendre. La dérivée partielle $\partial L/\partial u$, prise en $u = 1$, redonne la fonction $l(z) = \sum_{\mathcal{B}} lc(\mathcal{B}) z^{|\mathcal{B}|}$. Quant à la variance, elle s'obtiendra à partir du coefficient de z^n dans la dérivée $\partial^2 L/\partial u^2$, prise toujours en $u = 1$. Les deux fonctions $\partial L/\partial u|_{u=1}$ et $\partial^2 L/\partial u^2|_{u=1}$ peuvent s'obtenir en dérivant l'équation (4.6) par rapport à u ; nous laissons les calculs en exercice.

4.3 Valuations additives sur les arbres de Catalan

L'approche que nous avons utilisée pour étudier la longueur de cheminement s'étend à d'autres paramètres de ces arbres, à condition qu'ils satisfassent une relation de récurrence analogue à la relation (4.5). Une *valuation additive* est une fonction à valeurs réelles v, définie sur l'ensemble des arbres binaires, et telle que, si $\mathcal{B} = (\bullet, \mathcal{G}, \mathcal{D})$,

$$v(\mathcal{B}) = r(\mathcal{B}) + v(\mathcal{G}) + v(\mathcal{D}). \tag{4.7}$$

La valuation $r(\mathcal{B})$ est le coût associé à la racine, et est en général très simple. Par exemple, la valuation *taille,* associant à un arbre son nombre de nœuds, est obtenue pour $r(\mathcal{B}) = 1$; la valuation *longueur de cheminement* correspond à $r(\mathcal{B}) = |\mathcal{B}| - 1$.

Soit $V(z)$ la fonction génératrice de la valuation cumulée $v_n := \sum_{|\mathcal{B}|=n} v(\mathcal{B})$; cette fonction s'écrit aussi $V(z) = \sum_{\mathcal{B}} v(\mathcal{B})$, et s'exprime simplement à partir de la fonction génératrice $R(z)$ de la valuation de base $r(\mathcal{B})$. En effet,

$$\begin{aligned} V(z) &= \sum_{\mathcal{B}=(\bullet,\mathcal{G},\mathcal{D})} (r(\mathcal{B}) + v(\mathcal{G}) + v(\mathcal{D}))\, z^{|\mathcal{B}|} \\ &= \sum_{\mathcal{B}} r(\mathcal{B}) z^{|\mathcal{B}|} + 2 \sum_{\mathcal{G},\mathcal{D}} v(\mathcal{G}) z^{1+|\mathcal{G}|+|\mathcal{D}|} \\ &= R(z) + 2zV(z)b(z), \end{aligned}$$

où $b(z) = \sum_{\mathcal{B}} z^{|\mathcal{B}|}$ est, comme précédemment, la série énumérant les arbres binaires. On peut alors facilement obtenir

$$V(z) = \frac{R(z)}{1 - 2zb(z)} = \frac{R(z)}{\sqrt{1-4z}}.$$

La valeur moyenne de v sur les arbres de taille donnée n peut maintenant être obtenue ; c'est simplement $[z^n]V(z)/C_n$.

Comme exemple de valuation additive, outre la taille d'un arbre et sa longueur de cheminement, étudions le nombre de nœuds doubles d'un arbre binaire. Ce nombre est obtenu en prenant comme coût à la racine dans l'équation (4.7) $r(\mathcal{B}) = 1$ si les sous-arbres gauche et droit sont non vides, et 0 sinon :

$$R(z) = \sum_{\mathcal{G},\mathcal{D}\neq\emptyset} z^{1+|\mathcal{G}|+|\mathcal{D}|} = z(b(z) - 1)^2.$$

On en tire, après quelques calculs simples, que le nombre moyen de nœuds doubles dans un arbre binaire de taille n est asymptotiquement égal à $n/4$.

Exercices :
1. Peut-on obtenir la variance du nombre de nœuds doubles ?
2. Etudier le nombre de feuilles (resp. de nœuds simples) dans un arbre binaire.

4.4 Familles simples d'arbres

Tous les résultats que nous avons obtenu sur les arbres binaires sont valides pour des classes d'arbres plus générales, les *familles simples d'arbres* (pour reprendre une terminologie dûe à A. Meir et J.W. Moon).

Une famille simple d'arbres $\mathcal{F}$ est définie récursivement à partir d'un ensemble $\mathcal{S}$ de symboles, chaque symbole ayant une arité associée. Un arbre de $\mathcal{F}$ s'obtient en choisissant d'abord une racine dans $\mathcal{S}$, puis un nombre de sous-arbres de $\mathcal{F}$ égal à l'arité de la racine. L'ensemble $\mathcal{S}$ doit avoir au moins un symbole d'arité 0, pour assurer que les arbres construits soient finis. D'autre part, *la famille $\mathcal{F}$ ne contient pas l'arbre vide.*

Comme exemple de famille simple, considérons les expressions arithmétiques construites sur l'ensemble $\mathcal{S} = \{x, \surd, +, *\}$, où le symbole x est d'arité 0, et correspond donc aux feuilles, où le symbole $\surd$ est d'arité 1, et où les symboles $+$ et $*$ sont binaires. Les expressions de ce type peuvent s'écrire sous forme de grammaire :

$$\mathcal{E} = x \,|\, \sqrt{\mathcal{E}} \,|\; \mathcal{E} + \mathcal{E} \,|\, \mathcal{E} * \mathcal{E}\,,$$

ce qui nous permet d'obtenir la fonction génératrice de dénombrement :

$$F(z) := \sum_{\mathcal{E}} z^{|\mathcal{E}|} = z + zF(z) + 2zF^2(z),$$

d'où nous tirons $F(z) = (1 - z - \sqrt{1 - 2z - 7z^2})/(4z)$. Le nombre d'expressions de taille n est $f_n = [z^n]F(z)$. Il n'en existe pas de forme close analogue aux nombres de Catalan. On peut cependant donner f_n comme somme pondérée de binomiaux, à partir du développement de $\sqrt{1 - 2z - 7z^2} = \sqrt{1 - (2z + 7z^2)}$; on trouve

$$f_n = \frac{1}{2^{n+2}} \sum_{\frac{n+1}{2} \le m \le n+1} C_{m-1} \binom{m}{n+1-m} 7^{n+1-m}.$$

Cette formule ne donne pas immédiatement une idée du comportement asymptotique de f_n lorsque n tend vers l'infini. Pour cela, nous partirons plutôt de l'expression de la fonction $F(z)$ elle-même, en utilisant par exemple un *lemme de transfert* dû à P. Flajolet et A. Odlyzko. L'idée fondamentale est de considérer F comme une fonction analytique en z, et de relier ses singularités $\rho = (2\sqrt{2} - 1)/7$ et $\rho_1 = -(2\sqrt{2} + 1)/7$, ou plutôt sa singularité de plus petit module (ici égale à ρ), au comportement asymptotique de ses coefficients. Sans rentrer dans les détails, tout se passe comme si, au voisinage de la singularité principale ρ, on pouvait remplacer $F(z) = (1 - z - \sqrt{(1 - z/\rho)(1 - z/\rho_1)})/(4z)$ par $F_1(z) = (1 - \rho - \sqrt{(1 - z/\rho)(1 - \rho/\rho_1)})/(4\rho)$. On trouve que

$$F(z) = -\sqrt{1 + \frac{1}{2\sqrt{2}}}\,\sqrt{1 - \frac{z}{\rho}} + O(1 - z/\rho).$$

Le coefficient de z^n dans F_1 vient alors du terme $\sqrt{1 - z/\rho}$, ce qui donne asymptotiquement

$$f_n \sim \frac{\sqrt{1 + \frac{1}{2\sqrt{2}}}}{2n\sqrt{\pi n}}\, \rho^{-n}.$$

4.4.1 Dénombrement exact et formule de Lagrange

Le dénombrement que nous avons effectué sur l'exemple précédent peut être généralisé. Soit ϕ_k le nombre de symboles de $\mathcal{S}$ d'arité k, et soient $\Phi(u) := \sum_k \phi_k u^k$ et $F(z) := \sum_n f_n z^n$ les séries génératrices de dénombrement respectives de $\mathcal{S}$ et $\mathcal{F}$; nous avons $\Phi(0) = \phi_0 \neq 0$. La fonction $F(z)$ vérifie l'équation

$$F(z) = \phi_0 z + \phi_1 z F(z) + ... + \phi_k z F(z)^k + ...,$$

qui peut s'écrire

$$F(z) = z\Phi(F(z)). \tag{4.8}$$

En d'autres termes, F est solution d'une équation implicite, dont nous pourrons tirer certaines informations. Nous pourrons parfois résoudre explicitement l'équation (4.8). Dans certains cas, ceci permet d'obtenir une expression pour le coefficient f_n. Dans d'autres cas, on utilisera plutôt la formule de Lagrange pour calculer f_n. Enfin, l'équation implicite permet aussi d'obtenir des résultats sur le comportement asymptotique des f_n.

La *formule de Lagrange* relie les coefficients de $F(z)$ à ceux de la fonction $\Phi(u)$ apparaissant dans l'équation implicite :

$$[z^n]F(z) = \frac{1}{n}[u^{n-1}]\Phi^n(u).$$

Considérons par exemple la famille $\mathcal{P}$ des *arbres planaires généraux,* dans lesquels un nœud peut être d'arité quelconque. L'ensemble $\mathcal{S}$ a exactement un symbole d'arité k, et sa fonction génératrice est donc $\Phi(t) = \sum_{k\geq 0} t^k = (1-t)^{-1}$. La fonction $P(z)$ énumérant les arbres de $\mathcal{P}$ satisfait l'équation $P(z) = z/(1-P(z))$, qui peut bien entendu se résoudre directement. Si nous utilisons la formule de Lagrange, nous obtenons

$$[z^n]P(z) = \frac{1}{n}[u^{n-1}]\frac{1}{(1-u)^{n-1}} = \frac{1}{n}\binom{2n-2}{n-1} = C_{n-1}.$$

Comme autre exemple, prenons les *arbres ternaires*, dans lesquels les nœuds internes sont d'arité 3. Nous avons $\Phi(t) = 1 + t^3$, ce qui donne une équation de degré 3 sur F : $F(z) = z + zF(z)^3$. La formule de Lagrange donne, pour $n = 3p+1$ (seules valeurs pour lesquelles le coefficient est non nul)

$$[z^n]F(z) = \frac{1}{3p+1}[u^{3p}](1+u^3)^{3p+1} = \frac{1}{3p+1}\binom{3p+1}{p}.$$

Revenons enfin à l'exemple des expressions arithmétiques traité plus haut ; ici $\Phi(u) = 1 + u + 2u^2$, ce qui donne une autre expression pour le coefficient de z^n :

$$f_n = \frac{1}{n}\sum_{\frac{n-1}{2}\leq p\leq n-1}\binom{n}{p}\binom{p}{n-1-p}2^{n-1-p}.$$

Exercices :
1. Vérifier que les deux expressions obtenues pour le nombre f_n d'expressions arithmétiques construites sur x, $\sqrt{}$, $+$ et $*$ sont égales.
2. Calculer le nombre d'arbres p-aires, i.e. dont les nœuds ont pour seules arités possibles 0 (feuilles) ou p.

4.4.2 Dénombrement asymptotique

Lorsqu'il n'est pas possible d'obtenir une formule close pour f_n, l'équation (4.8) nous permet cependant d'obtenir son équivalent asymptotique. Posons $G(y,z) := y - z\Phi(y)$; la fonction $F(z)$ est définie comme la solution y de l'équation $G(y,z) = 0$. Le théorème des fonctions implicites nous assure de son existence, et le comportement asymptotique de f_n est relié à la singularité ρ de $F(z)$. Cette singularité est obtenue lorsqu'il n'est plus possible de résoudre l'équation $G(y,z)=0$ par rapport à y, i.e. lorsque la dérivée $\partial G/\partial y$ s'annule. Notons $\tau = F(\rho)$; ρ et τ sont solution du système

$$\begin{aligned} \tau &= \rho\Phi(\tau); \\ 1 &= \rho\Phi'(\tau). \end{aligned}$$

Nous résolvons d'abord l'équation $\tau\Phi'(\tau) = \Phi(\tau)$, qui nous permet de déterminer τ (supposé unique) ; puis nous obtenons la valeur de ρ, égale à $\tau/\Phi(\tau)$. Nous allons maintenant écrire l'équation (4.8) sous la forme $z = y/\Phi(y)$, i.e. considérer z comme une fonction de y, que nous développons autour de sa singularité τ :

$$z = \rho + (y-\tau)\left(\frac{y}{\Phi(y)}\right)'_{y=\tau} + \frac{1}{2}(y-\tau)^2\left(\frac{y}{\Phi(y)}\right)''_{y=\tau} + O((y-\tau)^3).$$

Par définition de τ, le coefficient de $(y-\tau)$ vaut $1/\Phi(\tau) - \tau\Phi'(\tau)/\Phi(\tau)^2$, et est donc nul, et celui de $(y-\tau)^2$ est égal à $-\rho\Phi''(\tau)/\Phi(\tau)$, ce qui donne

$$z - \rho = -\frac{\rho\,\Phi''(\tau)}{2\,\Phi(\tau)}(y-\tau)^2 + O\left((y-\tau)^3\right).$$

Nous inversons alors cette égalité, pour obtenir un développement de y en fonction de z au voisinage de ρ :

$$(y(z)-\tau)^2 = -\frac{2\,\Phi(\tau)}{\rho\,\Phi''(\tau)}(z-\rho) + O((z-\rho)^{3/2}),$$

ce qui donne, en choisissant la racine adéquate,

$$y(z) = \tau - \sqrt{\frac{2\,\Phi(\tau)}{\Phi''(\tau)}}\,\sqrt{1-z/\rho} + O((z-\rho)^{3/2}). \tag{4.9}$$

Nous avons donc

$$[z^n]y(z) = -\sqrt{\frac{2\,\Phi(\tau)}{\Phi''(\tau)}}\,[z^n]\,\sqrt{1-z/\rho} + [z^n]O((z-\rho)^{3/2}).$$

En général, on ne peut pas intervertir le coefficient de z^n avec le terme $O(...)$. Cependant, un résultat de P. Flajolet et A. Odlyzko, auquel nous avons déjà fait allusion plus haut (lemme de transfert), permet ici de justifier cette interversion, et on obtient

$$[z^n]y(z) = \sqrt{\frac{2\,\Phi(\tau)}{\Phi''(\tau)}}\,\rho^{-n}\frac{1}{2n\sqrt{\pi n}} + O([z^n]\,(z-\rho)^{3/2}),$$

soit finalement

$$f_n = \sqrt{\frac{\Phi(\tau)}{2\,\pi\,\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\,(1+O(1/n)).$$

Exercice : Que donne la méthode que nous venons de présenter pour les différentes familles simples d'arbres rencontrées ?

4.5 Familles simples d'arbres et valuations additives

Les valuations additives que nous avons rencontrées sur les arbres de Catalan peuvent être étendues aux familles simples d'arbres :

$$v(\mathcal{B}) = r(\mathcal{B}) + \sum_{\mathcal{A}\ sous-arbre\ de\ \mathcal{B}} v(\mathcal{A}). \tag{4.10}$$

Nous avons déjà rencontré la fonction de dénombrement $F(z) = \sum_{\mathcal{B}} z^{|\mathcal{B}|}$; introduisons la fonction génératrice de la valuation cumulée $V(z) := \sum_{\mathcal{B}} v(\mathcal{B}) z^{|\mathcal{B}|}$. Le coût moyen de v sur un arbre de taille fixée n sera $[z^n]V(z)/[z^n]F(z)$.

4.5.1 Comportement asymptotique

L'équation de récurrence (4.10) sur la valuation $v(\mathcal{B})$ se traduit en une équation sur sa fonction génératrice :

$$\begin{aligned}
V(z) &= \sum_{\mathcal{B}} r(\mathcal{B}) z^{|\mathcal{B}|} + \sum_{\mathcal{B}} \sum_{\mathcal{A}\ sous-arbre\ de\ \mathcal{B}} v(\mathcal{A}) z^{|\mathcal{B}|} \\
&= R(z) + \sum_{k\geq 0} \sum_{\mathcal{B}=(\bullet,\mathcal{A}_1,\ldots,\mathcal{A}_k)} (v(\mathcal{A}_1) + \ldots + v(\mathcal{A}_k)) z^{1+|\mathcal{A}_1|+\ldots+|\mathcal{A}_k|} \\
&= R(z) + z \sum_{k\geq 0} k \sum_{\mathcal{B}=(\bullet,\mathcal{A}_1,\ldots,\mathcal{A}_k)} v(\mathcal{A}_1) z^{|\mathcal{A}_1|+\ldots+|\mathcal{A}_k|} \\
&= R(z) + z \sum_{k\geq 0} k\, \phi_k V(z) F(z)^{k-1} \\
&= R(z) + z\, V(z)\, \Phi^{'}(F(z)).
\end{aligned}$$

Cette dernière équation nous permet d'obtenir

$$V(z) = \frac{R(z)}{1 - z\Phi'(F(z))}.$$

Au voisinage de $z = \rho$, il suffit de reporter le développement de $F(z)$ obtenu ci-dessus (cf. l'équation (4.9)) pour obtenir un développement de Taylor de $V(z)$, qui dépend naturellement du comportement de $R(z)$ en ρ. On peut alors obtenir l'équivalent asymptotique de $[z^n]V(z)$ pour $n \to +\infty$, et donc du coût moyen de la valuation v sur un arbre de taille n.

Exercices :
1. Etudier le comportement asymptotique de la longueur de cheminement moyenne.
2. Asymptotique du nombre de nœuds d'arité 1 dans la famille des arbres unaires-binaires.
3. Peut-on se servir de la fonction $W(u, z) := \sum_{\mathcal{B}} u^{v(\mathcal{B})} z^{|\mathcal{B}|}$ pour étudier la variance de la valuation v ?

4.5.2 Un exemple : analyse du coût de la différentiation

Nous montrons ici comment les idées précédentes permettent d'obtenir le coût moyen d'un algorithme de différentiation symbolique. Nous travaillerons sur des arbres représentant des expressions arithmétiques, construites à l'aide de symboles de variables, de constantes, et d'opérateurs. L'ensemble des symboles forme donc un ensemble $\mathcal{S}$, de fonction de dénombrement $\Phi(u)$. Nous en avons rencontré un exemple dans la section 3.4 ; nous le reprendrons et calculerons le coût moyen d'une dérivation sur des expressions construites à partir des symboles x, $\surd$, $+$ et $*$.

L'algorithme de différentiation prend en entrée une expression f, représentée par un arbre, et calcule l'expression dérivée Δf, elle aussi représentée par un arbre. Le coût de cet algorithme, en mémoire comme en temps, est lié à la taille de l'arbre dérivé : la valuation retenue est $\delta(f) = |\Delta f|$. Cette valuation satisfait une équation de récurrence du style de l'équation (4.10) :

$$\begin{aligned}
\delta(x) &= 1; \\
\delta(\sqrt{f}) &= 4 + |f| + \delta(f); \\
\delta(f+g) &= 1 + \delta(f) + \delta(g); \\
\delta(f*g) &= 3 + |f| + |g| + \delta(f) + \delta(g).
\end{aligned}$$

Soit $D(z) := \sum_f \delta(f) z^{|f|}$; les équations ci-dessus se traduisent sur $D(z)$ en

$$D(z) = \frac{z\,(1+2F(z))\,(1+2F(z)+zF'(z))}{1-z(1+4F(z))},$$

ce qui, compte tenu des relations $F^2(z) = ((1-z)F(z)-z)/2z$ et $F'(z) = F(z)/z(1-z-4zF(z))$, nous permet d'obtenir la valeur exacte de $D(z)$:

$$D(z) = \frac{-1+z+14z^2+14z^3+(1-6z^2)\sqrt{1-2z-7z^2}}{4z(1-2z-7z^2)}.$$

Au voisinage de $z = \rho$, nous obtenons

$$D(z) = \frac{1+\sqrt{2}}{8(1-z/\rho)} + O((1-z/\rho)^{-1/2}),$$

ce qui donne, en appliquant à nouveau un lemme de transfert : $[z^n]D(z) \sim (1+\sqrt{2})/8.\rho^{-n}$. Comme le nombre d'expressions de taille n vaut asymptotiquement $\rho^{-n}/(2n\sqrt{\pi n})$, la taille moyenne d'un arbre obtenu par dérivation d'une expression de taille n vaut donc asymptotiquement $an^{3/2}$, avec $a = (1+\sqrt{2})/(4\sqrt{\pi}\sqrt{1+1/2\sqrt{2}}) \sim 0,29268...$

Exercices :
1. Reprendre l'analyse avec l'ensemble de symboles $\mathcal{S} = \{x, +, -, *, \surd\}$, clos pour la dérivation. Attention : vous devez recalculer la fonction génératrice de dénombrement $F(z)$.
2. Etendre les résultats obtenus à un ensemble quelconque $\mathcal{S}$ de symboles.

Chapitre 5

Arbres aléatoires et processus de branchement

5.1 Motivations. Premiers arbres

Qu'elles viennent des mathématiques ou d'autres sciences que l'informatique, les raisons et motivations pour étudier les arbres sont nombreuses, citons parmi elles :

- le fait que les arbres constituent une simplification par rapport à une structure de réseau (verres de spin en physique) ou par rapport à un graphe (notion d'arbre couvrant), tout en restant une structure riche en dépendances.

- une motivation historiquement importante qui fut l'étude des modèles de ramification ou de reproduction, par Bienaymé, Galton et Watson au $XIX^{\text{ème}}$ siècle. Ces questions se posent maintenant naturellement en biologie moléculaire ou en physique.

- le *codage* d'ensembles compliqués de $\mathbb{R}^{\mathrm{d}}$, par un arbre, simplifie souvent la compréhension de la géométrie de l'ensemble. La théorie des fractales aléatoires ressort de ce domaine.

A transformer en exercice

Lemme 1 *Pour un arbre de Galton-Watson critique de fonction génératrice f, appelons $I(s)$ la fonction génératrice du cardinal de l'arbre. Alors $I(s)$ est l'unique solution de l'équation*

$$\frac{I(s)}{f \circ I(s)} = s \tag{5.1}$$

Démonstration :

$$I(s) = \mathbb{E}(s^{|\omega|})$$

et

$$|\omega| = 1 + \sum_{j=1}^{N} |T_j(\omega)|,$$

donc

$$s^{|\omega|} = s \prod_{j=1}^{N} s^{|T_j(\omega)|},$$

et en appliquant la propriété de branchement (5.3),

$$I(s) = s\mathbb{E}(I(s)^N) = sf \circ I(s)$$

d'où l'équation. De plus, pour un processus de branchement critique, la fonction $\frac{u}{f(u)}$ est croissante (calculer la dérivée et remarquer que $\sum_{k\geq 0} p_k(1-k)u^k$ est un barycentre) et vaut 0 en 0 et 1 en 1, d'où l'unicité. □

Appliquons ceci à la loi géométrique pour laquelle $f(s) = \frac{1}{2-s}$. L'équation 5.1 est du second degré en $I(s)$ et fournit

$$I(s) = 1 - \sqrt{1-s}.$$

Puisque $\sigma(\omega) = 2|J(\omega)|$, on a bien

$$\mathbb{E}_+(s^\sigma) = 1 - \sqrt{1-s^2}\ .$$

5.2 Espaces d'arbres aléatoires

5.2.1 Modèle d'arbre pour les processus de branchement

Le modèle le plus simple que l'on cherche à construire correspond à la situation où l'on se donne une loi de probabilité $(p_k)_{k\in\mathbb{N}}$ sur $\mathbb{N}$, et l'on souhaite étudier le processus Z_n défini par $Z_0 = 1$ (un seul ancêtre) et l'ancêtre a k descendants avec probabilité p_k. Les descendants se reproduisent avec la même loi, indépendamment les uns des autres et de leur ascendant.

Nous allons pour cela définir un espace probabilisé d'arbres $(\Omega, \mathcal{F}, \mathbb{P})$, où $\mathbb{P}$ sera calibré en fonction du phénomène aléatoire que l'on souhaite étudier. Partons de U l'espace des suites finies $u = j_1...j_n$ d'entiers strictement positifs. La suite notée $\emptyset$ (parfois aussi notée $*$) appartient à U. Ainsi

$$U = \{\emptyset\} \cup \bigcup_{n\geq 1} \mathbb{N}^{*n}$$

U est l'ensemble d'"étiquettes" qui nommeront canoniquement les noeuds de l'arbre. La longueur d'une suite est notée $|u|$; la suite obtenue par concaténation de u et v se note uv. $|\emptyset| = 0$.

Définition 1 *Un <u>arbre</u> ω est un sous-ensemble de U qui vérifie les trois conditions suivantes*

(i) $\emptyset \in \omega$ (un arbre n'est jamais vide, il contient au moins la racine)

(ii) $uv \in \omega \Rightarrow u \in \omega$ (on peut remonter le long d'une branche vers la racine)

(iii) $u \in \omega \Rightarrow \exists N_u(\omega) \in \mathbb{N}$ tel que $uj \in \omega, j \geq 1 \Leftrightarrow 1 \leq j \leq N_u(\omega)$ (on peut descendre vers les feuilles)

On voit que la donnée de la suite des $N_u, u \in U$ (nombre de descendants de chaque nœud u), assortie des conditions précédentes, suffit à définir un arbre.

Autrement dit, un arbre "probabiliste" est un arbre planaire, fini ou non, dont les noeuds sont d'arité variable et sont étiquetés par des suites finies d'entiers indiquant la place du noeud dans l'arbre. Dans la suite de ce chapitre, on appellera "arbre" un élément défini comme ci-dessus.

On notera plutôt N au lieu de $N_\emptyset$. N est l'arité de la racine. L'ensemble Ω des arbres est muni de la tribu $\mathcal{F}$ engendrée par les $\Omega_u, u \in U$ où $\Omega_u = \{\omega \in \Omega, u \in \omega\}$. Ainsi les N_u sont des variables aléatoires définies sur Ω_u et N_u désigne le nombre de descendants du noeud u. Pour tout entier n la n-ième génération est

$$z_n(\omega) = \{u \in \omega, |u| = n\}$$

et

$$Z_n(\omega) = Card\ z_n(\omega)$$

(on a $Z_1 = N$). Cette variable aléatoire va beaucoup nous intéresser dans ce chapitre.
Attention : notez bien qu'ici n désigne le niveau courant, le numéro de génération considéré et pas du tout le nombre total de noeuds (comme dans les chapitres précédents), qui peut d'ailleurs être infini.

L'espace des arbres Ω est alors muni d'une filtration $\mathcal{F}_n$ engendrée par les $z_m, 0 \leq m \leq n$ et de translations

$$\begin{array}{rccll} T_u : & \Omega & \rightarrow & \Omega & \\ & \omega & \mapsto & T_u(\omega) & = \{v, v \in U, uv \in \omega\} \end{array}$$

de sorte que T_u désigne le sous-arbre dont la racine est u. Par exemple, $T_1, T_2, \ldots, T_N$ sont les sous-arbres fils de la racine, ils constituent une forêt. On voit facilement que pour tous n et k dans $\mathbb{N}$,

$$Z_{n+k} = \sum_{u \in z_n} Z_k \circ T_u \tag{5.2}$$

qui est une relation typique et fondamentale dans l'étude des processus de branchement.

Remarque méthodologique : Le caractère récursif de la structure d'arbre est à la base de la plupart des méthodes de démonstration ; il peut se traduire de deux manières que l'on classe en "méthodes forward" et "méthodes backward". Dans la "méthode forward" on décrit la situation à l'étape $n+1$ en fonction de la situation à l'étape n, puis en prenant l'espérance conditionnelle sachant le passé jusqu'à l'instant n, on fait apparaître des propriétés de martingales ou plus généralement des invariants. Dans la "méthode backward", on décrit la situation à l'étape $n+1$ comme reproduisant la situation à l'étape n pour les enfants de l'ancêtre. Cette méthode s'appelle aussi "divide-and-conquer". La relation de récurrence ainsi obtenue se traduit par une équation différentielle sur les fonctions génératrices correspondantes. Ces deux classes de méthodes sont complémentaires. Par exemple, sur le processus Z_n ci-dessus, un raisonnement forward utilisera la relation

$$Z_{n+1} = \sum_{u \in z_n} Z_1 \circ T_u = \sum_{u \in z_n} N_u$$

alors qu'un raisonnement backward utilisera la relation

$$Z_{n+1} = \sum_{i=1}^{N} Z_n \circ T_i$$

Un arbre peut être fini ou infini. La hauteur d'un arbre ω est

$$h(\omega) = sup\{|u|, u \in \omega\}.$$

Dans la suite, définir un processus de branchement reviendra à probabiliser l'espace $(\Omega, \mathcal{F})$. Si l'on veut par exemple mettre des “clés” à chaque noeud de l'arbre, il faudra que chaque noeud de l'arbre soit muni d'une “marque” disons X_u qui peut être aussi compliquée et riche que nécessaire (X_u pourra être une variable aléatoire), c'est la notion d'“arbre marqué” que l'on verra au paragraphe 5.3.1.

5.2.2 Arbre de Galton-Watson

Proposition-Définition 1 *Soit $p = (p_k)_{k\geq 0}$ une loi de probabilité sur $\mathbb{N}$. Alors il existe une unique probabilité $\mathbb{P}_p$ (notée $\mathbb{P}$ quand il n'y a pas d'ambiguité) sur $(\Omega, \mathcal{F})$ telle que*

(i) N a pour loi p.

(ii) conditionnellement sachant $\{N = j\}$, les variables $T_1, T_2, ..., T_j$ sont indépendantes et de même loi $\mathbb{P}_p$.

Par conséquent, pour tout $n \geq 0$, conditionnellement par rapport à $\mathcal{F}_n$, les variables aléatoires $\{T_u, u \in z_n\}$ sont i.i.d. de loi $\mathbb{P}_p$. Autrement dit encore, pour toutes fonctions mesurables positives $f_u, u \in U$

$$\mathbb{E}_p^{\mathcal{F}_n}(\prod_{u\in z_n} f_u \circ T_u) = \prod_{u\in z_n} \mathbb{E}_p(f_u) \tag{5.3}$$

La suite $(Z_n)_{n\geq 0}$ s'appelle un processus de Galton-Watson de loi p et d'état initial $Z_0 = 1$. L'égalité (5.3) s'appelle propriété de branchement.

<u>Démonstration</u>

Dans [15], pages 202, 203. C'est une propriété d'existence de processus de Markov. □

Dans la suite, on utilisera beaucoup la fonction génératrice de la loi p :

$$f(s) = \mathbb{E}_p(s^N) = \sum_{k\geq 0} p_k s^k = \mathbb{E}_p(s^{Z_1}), \quad s \in [0, 1]$$

On pose

$$m = f'(1) = \sum_{k\geq 0} kp_k = \mathbb{E}(Z_1).$$

Dans la suite on supposera toujours que la moyenne m est finie (attention, cela ne veut pas dire du tout que N est borné) et on déduit immédiatement de la relation 5.2 et de la propriété de branchement 5.3 que

$$\mathbb{E}_p(s^{Z_n}) = f^{(n)}(s)$$

où $f^{(n)}(s) := f \circ f \circ \cdots \circ f(s)$, n fois.

La première question qui se pose est celle de l'extinction du processus. Si

$$q = \mathbb{P}_p(\exists n, Z_n = 0)$$

est la probabilité d'extinction et si le temps d'extinction T est le premier entier n tel que $Z_n = 0$, on a

Proposition 1 *Extinction du processus :*
(i) $q = \lim_{n \to +\infty} f^{(n)}(0)$
(ii) $q = 1 \Leftrightarrow m \leq 1$ *(quand* $p_1 \neq 1$*)*
q *est la plus petite racine dans* $[0,1]$ *de l'équation* $f(s) = s$

Le processus de branchement est dit sous-critique (resp critique, surcritique) lorsque $m \leq 1$ (resp $m = 1$, $m > 1$). Remarquons que si $p_0 = 0$, alors le processus est bien entendu surcritique (on supposera dans tout ce chapitre que l'on n'est pas dans le cas trivial où $p_1 = 1$).
La taille de la population est précisée grâce à la proposition suivante

Proposition 2 *Martingale de Galton-Watson.*

$$W_n := \frac{Z_n}{m^n}$$

est une $\mathcal{F}_n$*-martingale d'espérance égale à 1, qui converge p.s. vers* W *quand* $n \to +\infty$*. En particulier* $\mathbb{E}(Z_n) = m^n$*.*

Démonstration : immédiate avec

$$Z_{n+1} = \sum_{u \in z_n} Z_1 \circ T_u \qquad \square$$

Le lemme suivant qui établit une loi 0-1 pour un processus de Galton-Watson sera souvent utile. Il est appliqué tout de suite pour relier l'extinction de l'arbre à W.

Lemme 2 *On dit qu'une propriété est "héritée" quand, lorsqu'un arbre possède cette propriété, les arbres* $T_1, T_2, ..., T_N$ *la possèdent aussi, et si tout arbre fini la possède. Toute propriété héritée est de probabilité q ou 1. Par conséquent, conditionnellement à la non-extinction, toute propriété héritée est de probabilité 0 ou 1.*

Démonstration : Soit E l'ensemble des arbres possédant une certaine propriété héritée. Par la propriété de branchement appliquée à $1_E(\omega) \leq \prod_{i=1}^{N} 1_E(T_i(\omega))$, on obtient $\mathbb{P}(E) \leq f(\mathbb{P}(E))$ et comme $\mathbb{P}(E) \geq q$, c'est que $\mathbb{P}(E) \in \{q, 1\}$. $\square$

Comme $\{W = 0\}$ est clairement une propriété héritée, on a la

Proposition 3 $\mathbb{P}(W = 0) = q$ *ou* 1
Soit $W = 0$ *p.s. soit* $\mathbb{P}(W > 0) = 1 - q$ *et alors* $\{W > 0\} = \{Z_n \to \infty\} = E^c$*.*

Par conséquent, pour un processus de branchement sous-critique ou critique, $q = 1$ et $W = 0$ p.s. et pour un processus de branchement surcritique, on ne sait pas bien (encore) on peut avoir $W = 0$ p.s. ou bien $\mathbb{P}(W = 0) = q$ et $W > 0$ sur l'ensemble de non-extinction.

La relation suivante qui se déduit immédiatement de la propriété de branchement, est fondamentale dans l'étude de la frontière d'un arbre. Appelons $W_u := W \circ T_u = \lim_{n\to\infty} \frac{Z_n \circ T_u}{m^n}$ de sorte que pour tout entier n, p.s.

$$W = \frac{1}{m^n} \sum_{u \in z_n} W_u \tag{5.4}$$

Exercice : Ecrire cette relation pour $n = 1$ et en déduire une équation fonctionnelle satisfaite par la transformée de Laplace de W.

La question suivante qui se pose est : dans le cas où W_n converge vers une limite strictement positive, c'est-à-dire dans le cas surcritique, y a-t-il convergence dans L^1 ? Cette question est résolue dans le thérème de Kesten-Stigum du paragraphe 5.2.4. Ce théorème peut être démontré de plusieurs manières. Une démonstration récente est basée sur la description d'un arbre remarquable, l'"arbre biaisé" (cf. Lyons, Pemantle et Peres [13]).

5.2.3 L'arbre biaisé

Par le théorème de Kolmogorov, toute famille compatible de probabilités Q_n définies sur $(\Omega, \mathcal{F}_n)$ induit une unique probabilité sur $(\Omega, \mathcal{F})$ qui prolonge les Q_n. Ceci permet de définir une probabilité $\widehat{\mathbb{P}}$ sur $(\Omega, \mathcal{F})$ telle que

$$\widehat{\mathbb{P}} = \frac{Z_n}{m^n} \,.\, \mathbb{P} \text{ sur } \mathcal{F}_n, \forall n \in \mathbb{N}$$

autrement dit,

$$\forall n \in \mathbb{N}, \forall A \in \mathcal{F}_n, \quad \widehat{\mathbb{P}}(A) = \int_A \frac{Z_n}{m^n}\, d\mathbb{P}$$

On appellera <u>arbre biaisé</u> l'arbre sous $\widehat{\mathbb{P}}$.

<u>Propriétés de $\widehat{\mathbb{P}}$</u>

$$\begin{aligned} \widehat{\mathbb{P}}(T > n) &= \widehat{\mathbb{P}}(Z_n \neq 0) = \int 1_{Z_n \neq 0} \frac{Z_n}{m^n}\, d\mathbb{P} \\ &= \int 1_{Z_n \neq 0} \frac{Z_n}{m^n}\, d\mathbb{P} + \int 1_{Z_n = 0} \frac{Z_n}{m^n}\, d\mathbb{P} = \int \frac{Z_n}{m^n}\, d\mathbb{P} = \mathbb{E}(\frac{Z_n}{m^n}) = 1 \end{aligned}$$

de sorte que pour tout n, $\{T > n\}$ est de $\widehat{\mathbb{P}}$-probabilité égale à 1 et donc $\widehat{\mathbb{P}}$ est portée par $\{T = \infty\}$. Ainsi, si $m \leq 1$, $\mathbb{P}$ et $\widehat{\mathbb{P}}$ sont étrangères.

Dans le cas $m > 1$, un petit raisonnement d'intégration, non détaillé ici (voir Peres [16]) conduit à la dichotomie (*) suivante :

1. soit on est dans le cas

$$W = 0 \ \mathbb{P}p.s. \Leftrightarrow \mathbb{P}(W = 0) = 1 \Leftrightarrow \mathbb{P} \perp \widehat{\mathbb{P}} \Leftrightarrow W = \infty \ \widehat{\mathbb{P}}p.s.$$

2. soit on est dans le cas

$$\mathbb{E}(W) = 1 \Leftrightarrow \widehat{\mathbb{P}} << \mathbb{P} \Leftrightarrow W < \infty \ \widehat{\mathbb{P}}p.s.$$

Revenons aux cas sous-critique ou critique pour lesquels on peut expliquer la loi biaisée de la façon suivante.

Proposition 4 *Dans le cas $m \leq 1$ et pour tout $A \in \bigcup \mathcal{F}_n$,*

$$\widehat{\mathbb{P}}(A) = \lim_{n\to\infty} \mathbb{P}(A/Z_n \neq 0) \tag{5.5}$$

Ainsi l'arbre biaisé est obtenu en conditionnant un arbre de Galton-Watson sous-critique ou critique à survivre indéfiniment.

Démonstration

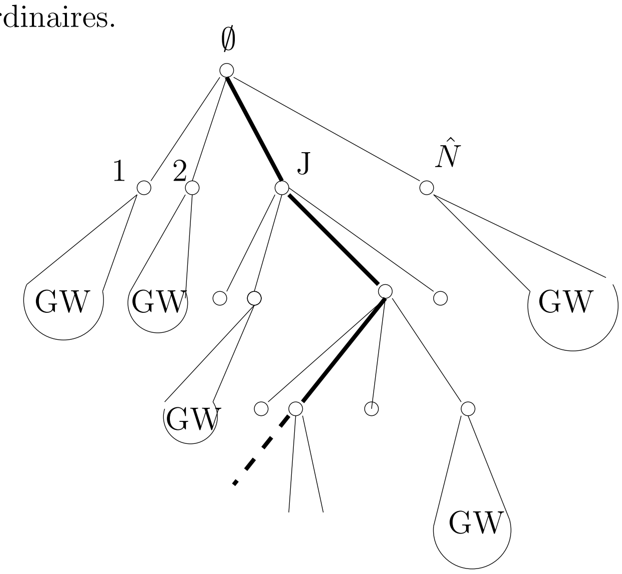
Partir de $\mathbb{P}(A/Z_{n+p} \neq 0)$ pour $A \in \mathcal{F}_p$ et utiliser la propriété de branchement. □

Proposition 5 *Description de l'arbre sous $\widehat{\mathbb{P}}$: $\forall (A_j)_{j\in\mathbb{N}} \in \mathcal{F}, \forall k \geq 0$,*

$$\widehat{\mathbb{P}}(N = k, T_j \in A_j, j = 1, \ldots, k) = \frac{kp_k}{m} \frac{1}{k} \sum_{j=1}^{k} \mathbb{P}(A_1) \ldots \widehat{\mathbb{P}}(A_j) \ldots \mathbb{P}(A_k) \tag{5.6}$$

Démonstration : utiliser la définition de $\widehat{\mathbb{P}}$. □

Autrement dit, pour l'arbre biaisé, la loi de reproduction $\hat{N}$ de l'ancêtre est biaisée et $\mathbb{P}(\hat{N} = k) = \frac{kp_k}{m}$. Puis un descendant, disons J, est choisi au hasard uniformément parmi la première génération et l'arbre T_J sera biaisé alors que les T_j pour $j \neq J$ sont des arbres de Galton-Watson ordinaires.



Il y a donc une branche infinie (une épine dorsale) et des cheveux Galton-Watson qui en partent.

Exercice : Dessiner et décrire un arbre biaisé pour $p_0 = p_2 = 1/2$.

Remarque : interprétation en termes de processus de Galton-Watson avec immigration.

Un processus de G-W avec immigration est caractérisé par une loi de reproduction et une loi d'immigration. A chaque génération il y a immigration de disons Y_n particules où les Y_n sont iid de loi la loi d'immigration. Pendant ce temps, les particules se reproduisent indépendamment les unes des autres avec la loi de reproduction.

On voit ainsi que la description précédente peut aussi se traduire par : $Z_n - 1$ sous $\widehat{\mathbb{P}}$ est un processus de Galton-Watson avec immigration $Y_n = \hat{N}_n - 1$ où $\hat{N}$ a pour loi $\frac{kp_k}{m}$.

5.2.4 Processus surcritique. Le théorème de Kesten-Stigum

Théorème 1 *Théorème de Kesten-Stigum (1966) :*

Pour un processus de Galton-Watson surcritique (où $Z_0 = 1$ et $1 < m < \infty$), les assertions suivantes sont équivalentes

(i) $\mathbb{P}(W = 0) = q$

(ii) $\mathbb{E}(W) = 1$

(iii) $W_n := \frac{Z_n}{m^n} \xrightarrow{L^1} W$

(iv) (W_n) *est uniformément intégrable*

(v) $\mathbb{E}(\sup_n W_n) < +\infty$

(vi) $\mathbb{E}(Nlog^+N) < +\infty$*, c'est-à dire* $\sum_{k=2}^{\infty} p_k(klogk) < \infty$

Démonstration :

Les équivalences entre (ii) (iii) et (iv) sont des résultats généraux. (ii) $\Rightarrow$ (i) se déduit facilement du lemme 5.4 sur les propiétés héritées : puisque $W = 0$ est une propriété héritée, $\mathbb{P}(W = 0) = q$ ou 1. (v) $\Rightarrow$ (iv) est trivial. Montrons (ii) $\Rightarrow$ (v). Cela repose sur le lemme suivant dont la démonstration se trouve dans [1] page 27.

Lemme 3 *Supposons que* $\mathbb{P}(W = 0) = q$*. Posons* $M = sup_n W_n$ *Alors il existe* A *et* $B > 0$ *tels que* $\forall x \geq 1, \forall n \in \mathbb{N}$

$$\mathbb{P}(W \geq Ax) \geq B\mathbb{P}(M \geq x)$$

Avec ce lemme, on a

$$\begin{aligned} \mathbb{E}(M) &= \int_0^{\infty} \mathbb{P}(M \geq x)dx \leq 1 + \frac{1}{B}\int_1^{\infty} \mathbb{P}(W \geq Ax)dx \\ &\leq 1 + \frac{1}{B}\mathbb{E}(\frac{W}{A}) = 1 + \frac{\mathbb{E}(W)}{AB} < \infty \end{aligned}$$

et donc (v) est vérifié.

On va voir que la suite de la démonstration du théorème de Kesten-Stigum se déduit de l'étude de l'arbre biaisé. On a immédiatement que :

$$\text{(vi) } \mathbb{E}(Nlog^+N) < +\infty \Leftrightarrow \mathbb{E}(log\hat{N}) < \infty \tag{5.7}$$

où $\hat{N}$ est la v.a. à valeurs dans $\mathbb{N}$ de loi $\mathbb{P}(\hat{N} = k) = \frac{kp_k}{m}$ que l'on a définie dans l'arbre biaisé. Le lemme suivant est un exercice d'intégration et il est nécessaire pour montrer le théorème de Seneta qui suit.

Lemme 4 *Soient* $X, X_1, X_2, \ldots$ *iid, positives. Alors p.s.*

$$\limsup_{n\to\infty} \frac{1}{n}X_n = 0 \text{ si } E(X) < \infty$$

$$\limsup_{n\to\infty} \frac{1}{n}X_n = \infty \text{ si } E(X) = \infty$$

Admettons le théorème suivant sur les processus avec immigration, dû à Seneta (1970) (démonstration dans Peres [16], qui utilise le lemme ci-dessus).

Théorème 2 *Théorème de Seneta (1970) :*
Soit Z_n les tailles des générations dans un processus de GW avec immigration Y_n. On suppose que m, moyenne de la loi de reproduction est > 1. Si $\mathbb{E}(log^+Y) < \infty$, alors la limite de Z_n/m^n existe et est finie p.s. Si $\mathbb{E}(log^+Y) = \infty$, alors $\limsup Z_n/c^n = \infty$ pour toute constante $c > 0$.

Il reste à montrer (vi) $\Rightarrow$ (ii) et (i) $\Rightarrow$ (vi).

Supposons (vi). Alors $\mathbb{E}(log\hat{N}) < \infty$. Or sous $\widehat{\mathbb{P}}$, $Z_n = 1 + (Z_n - 1)$ où $Z_n - 1$ est un processus de Galton-Watson avec immigration $\hat{N} - 1$. Par le théorème 2 d'immigration de Seneta, on en déduit $W < \infty$ $\widehat{\mathbb{P}}p.s.$, et grâce à la dichotomie (*), donc $\mathbb{E}(W) = 1$, c'est-à-dire (ii).

Inversement si l'on n'a pas (vi), alors par le même théorème, $W = \infty$ $\widehat{\mathbb{P}}p.s.$, et donc $\mathbb{P}(W = 0) = 1 \neq q$ puisque $m > 1$ donc (i) n'est pas vérifié. Ce qui termine la démonstration du théorème de Kesten-Stigum. □

5.2.5 Processus sous-critique ou critique

Le processus s'éteint presque sûrement. La question est de savoir à quelle vitesse et quel est son comportement avant l'extinction. Nous donnons ci-dessous les résultats principaux. Les preuves reposent sur l'utilisation de la fonction génératrice f et elles ont aussi été revisitées avec l'arbre biaisé.

Cas sous-critique

Théorème 3 *Théorème de Yaglom (1947, puis améliorations en 1967) :*
Pour un processus de Galton-Watson sous-critique (où $0 < m < 1$), il y a une loi limite du processus conditionné par la non extinction :

$$\mathbb{P}(Z_n = j/Z_n > 0) \stackrel{n\to\infty}{\longrightarrow} Q(j)$$

où Q est une loi de probabilité sur $\mathbb{N}$ non dégénérée.

On montre aussi que le taux de décroissance de $P(Z_n > 0)$ (qui est majoré par $E(Z_n) = m^n$) est bien de l'ordre de m^n, sous condition de moment.

Théorème 4 *(1967)*
La suite de terme général $P(Z_n > 0)/m^n$ est décroissante. Dans le cas sous-critique $m < 1$, les propriétés suivantes sont équivalentes
(i) $\lim\limits_{n\to\infty} \dfrac{\mathbb{P}(Z_n > 0)}{m^n} > 0$
(ii) $\sup\limits_n \mathbb{E}(Z_n/Z_n > 0) < \infty$
(iii) $\mathbb{E}(Nlog^+N) < \infty$.

Cas critique

Théorème 5 *:*
On suppose $m = 1$ et $\sigma^2 := Var(N) \leq \infty$. Alors

$$\lim_{n\to\infty} n\mathbb{P}(Z_n > 0) = \frac{2}{\sigma^2}$$

et si $\sigma^2 < \infty$, on a la loi limite de Yaglom : la loi conditionnelle de $\frac{Z_n}{n}$ converge vers une loi exponentielle de moyenne $\frac{\sigma^2}{2}$. Si $\sigma = \infty$, alors cette loi conditionnelle converge vers $+\infty$.

5.3 Processus spatiaux

5.3.1 Arbres marqués

Pour des modèles plus complexes que celui de Galton-Watson, on a besoin de "marquer" les branches de l'arbre par des variables aléatoires.

Définition 2 *Soit Γ un espace mesurable. Un* *<u>arbre marqué</u>* *est un couple*

$$\bar{\omega} = (\omega, (\gamma_u)_{u\in\omega})$$

où ω est un arbre et pour tout $u \in U$, $\gamma_u \in \Gamma$. On notera $\bar{\Omega}$ l'espace des arbres marqués par Γ et comme dans l'espace des arbres Ω, on appellera $\bar{\Omega}_u$ l'ensemble des arbres marqués contenant un nœud u. Les γ_u sont alors des variables aléatoires définies sur $\bar{\Omega}_u$ à valeurs dans Γ. L'espace des arbres marqués est muni de la tribu $\bar{\mathcal{F}}$ engendrée par ces applications. On note sans indice les variables aléatoires attachées à l'ancêtre.

On a l'analogue de la définition-proposition 1 :

Proposition-Définition 2 *Soit Q une loi de probabilité sur $\mathbb{N} \times \Gamma$, où Γ est un espace mesurable. Alors il existe une unique probabilité $\mathbb{P}_Q$ (notée $\mathbb{P}$ quand il n'y a pas d'ambiguité) sur $(\bar{\Omega}, \bar{\mathcal{F}})$ telle que*
(i) (N, γ) a pour loi Q.
(ii) (propriété de branchement) conditionnellement sachant $\{N = j\}$ et γ, les variables $T_1, T_2, ..., T_j$ sont indépendantes et de même loi $\mathbb{P}$.

Par conséquent, pour tout $n \geq 0$, conditionnellement par rapport à $\bar{\mathcal{F}}_n$ (tribu engendrée par les variables $N_u, |u| < n$ et $\gamma_u, |u| \leq n$), les variables aléatoires $\{T_u, u \in z_n\}$ sont i.i.d. de loi $\mathbb{P}$. Autrement dit encore, pour toutes fonctions mesurables positives $f_u, u \in U$

$$\mathbb{E}^{\bar{\mathcal{F}}_n}(\prod_{u\in z_n} f_u \circ T_u) = \prod_{u\in z_n} \mathbb{E}(f_u) \ . \tag{5.8}$$

Les marques peuvent être des durées de vie à valeurs dans $\mathbb{R}$; dans ce cas en appelant Z_t le nombre d'individus en vie à l'instant t, on obtient le processus des âges de Bellmann-Harris (voir chapitre IV de [A-N]).

Les marques peuvent être des trajectoires ; prenons pour Γ l'espace des fonctions continues sur un intervalle de la forme $[0, \sigma]$ à valeurs dans $\mathbb{R}$ ou $(\mathbb{R}^{\mathrm{d}})$ et prenons comme loi de γ celle d'un mouvement brownien standard arrêté à un temps exponentiel. Alors la mesure des emplacements des particules à l'instant t est un processus de branchement brownien. Nous allons prendre dans la suite l'analogue discret du branchement brownien : la marche aléatoire branchante (branching random walk).

5.3.2 Marche aléatoire branchante

Le phénomène se décrit de la façon suivante : un ancêtre se trouve en 0 (ou en x) à l'instant 0. Ses enfants forment la première génération et leurs déplacements sont donnés par un processus ponctuel $\mathcal{Z}$ sur $\mathbb{R}$ qui est la somme des mesures de Dirac en chaque déplacement (δ désigne la mesure de Dirac).

$$\mathcal{Z} = \delta_{\gamma_1} + \cdots + \delta_{\gamma_N}$$

N est le nombre aléatoire d'enfants de l'ancêtre et les γ_i sont leurs déplacements. Il n'y a pas d'hypothèse d'indépendance. Ensuite, chaque particule u donne naissance, indépendamment des autres particules et du passé, à des enfants selon un processus ponctuel copie de $\mathcal{Z}$. Ce modèle est discret mais un modèle continu se décrit tout aussi bien avec des durées de vie aléatoires (cf. Uchiyama [17]), nous y reviendrons plus loin. La loi du processus $\mathcal{Z}$ est donnée par sa mesure d'intensité, disons μ (μ est une mesure sur $\mathbb{R}$) par : pour toute fonction f mesurable positive sur $\mathbb{R}$,

$$\mathbb{E}(\mathcal{Z}(f)) = \mathbb{E}(\sum_{j=1}^{N} f(\gamma_j)) = \int_{\mathbb{R}} f(x)\mu(dx)\,.$$

L'espace considéré $\bar{\Omega}$ est celui des arbres marqués par les déplacements γ_u à valeurs dans $\mathbb{R}$ des particules ($\Gamma = \mathbb{R}$). On définit la position X_u de la particule $u = i_1 i_2 \dots i_n$ par

$$X_u = X_\emptyset + \gamma_{i_1} + \gamma_{i_1 i_2} + \cdots + \gamma_{i_1 \dots i_n}\,.$$

On effectue ainsi une marche aléatoire sur chaque branche allant de l'ancêtre au nœud u. Il s'agit ainsi d'un modèle <u>additif</u> le long de chaque branche de l'arbre. En passant à l'exponentielle (c'est-à-dire en considérant comme variables les e^{X_u}) on obtiendrait un modèle <u>multiplicatif</u> le long des branches, comme dans les cascades ou les fractales notamment.

Pour décrire le processus, on appelle

$$\mathcal{Z}_n := \sum_{|u|=n} \delta_{X_u}$$

le processus ponctuel des positions des particules de la n-ième génération. On a $\mathcal{Z}_0 = \delta_x$ et $\mathcal{Z}_1 = \sum_{j=1}^{N} \delta_{x+\gamma_j}$.

La propriété de branchement qui suit s'est enrichie de la composante spatiale.

Proposition-Définition 3 *Pour toute mesure μ de comptage sur $\mathbb{R}$ (qu'on supposera dans la suite de logLaplace finie partout) et pour tout $x \in \mathbb{R}$, il existe une unique probabilité $\mathbb{P}_x$ sur $(\bar{\Omega}, \bar{\mathcal{F}})$ telle que* *(i)* $X_\emptyset = x$

$\mathcal{Z}_1$ est un processus ponctuel de mesure d'intensité μ, c'est-à-dire que pour toute fonction f mesurable positive

$$\mathbb{E}_x(\mathcal{Z}_1(f)) = \mathbb{E}_x(\sum_{j=1}^{N} f(X_j)) = \int_{\mathbb{R}} f(x+y)\mu(dy)$$

(ii) pour tout $n > 0$, conditionnellement à $\bar{\mathcal{F}}_n$ (tribu engendrée par les variables $N_u, |u| < n$ et $\gamma_u, |u| \leq n$), les variables aléatoires $\{T_u, u \in z_n\}$ sont i.i.d. de loi $\mathbb{P}_{X_u}$. Autrement dit encore, pour toutes fonctions mesurables positives $f_u, u \in U$

$$\mathbb{E}_x^{\bar{\mathcal{F}}_n}(\prod_{u \in z_n} f_u \circ T_u) = \prod_{u \in z_n} \mathbb{E}_{X_u}(f_u) \tag{5.9}$$

Dans la suite on notera $\mathbb{E}$ pour $\mathbb{E}_0$. On supposera se trouver dans le cas surcritique où $m := \mathbb{E}(\mathcal{Z}(1)) = \mathbb{E}(N) = \mu(\mathbb{R}) > 1$.

Remarquons que l'intensité du processus $\mathcal{Z}_n$ ci-dessus est alors μ^{*n}. L'analogue de la martingale Z_n/m^n du processus de Galton-Watson est pour tout θ réel

$$W_n(\theta) = \sum_{u \in z_n} e^{-\theta X_u - n\Lambda(\theta)} = \int_{\mathbb{R}} e^{-\theta x - n\Lambda(\theta)} \mathcal{Z}_n(dx)$$

où $\Lambda(\theta) = \log m(\theta)$ et

$$m(\theta) = \mathbb{E}(\sum_{j=1}^{N} e^{-\theta X_j}) = \int_{\mathbb{R}} e^{-\theta x} \mu(dx) = \mathbb{E}(\mathcal{Z}_1(e^{-\theta \cdot})),$$

est la transformée de Laplace du processus $\mathcal{Z}$ (la notation $\mathcal{Z}_1(\theta) := \mathcal{Z}_1(e^{-\theta \cdot})$ sera utilisée par la suite).

Remarquons qu'en faisant $\theta = 0$, on retrouve le processus de Galton-Watson sous-jacent. Le processus des âges est également un cas particulier de ce modèle en supposant la reproduction et le mouvement indépendants et en supposant que la loi du déplacement est concentrée sur $\mathbb{R}_+$ (le déplacement devient un temps).

La propriété de martingale passe sans difficulté au cas spatial :

Théorème 6 *(Kingman, 1975)*

Pour tout $\theta \in \mathbb{R}$, $W_n(\theta)$ est une $\mathcal{F}_n$-martingale positive. Elle converge presque sûrement vers $W(\theta)$ avec a priori $\mathbb{E}(W(\theta)) \leq 1$.

<u>Démonstration</u> : on applique la propriété de branchement à

$$\begin{aligned}
\mathbb{E}_x^{\mathcal{F}_n} W_{n+1}(\theta) &= \tfrac{1}{m(\theta)^{n+1}} \sum_{u \in z_n} \mathbb{E}_x^{\mathcal{F}_n} \sum_{j=1}^{N \circ T_u} e^{-\theta X_j \circ T_u} \\
&= \tfrac{1}{m(\theta)^{n+1}} \sum_{u \in z_n} \mathbb{E}_{X_u}(\sum_{j=1}^{N} e^{-\theta X_j}) \\
&= \sum_{u \in z_n} \frac{e^{-\theta X_u}}{m(\theta)^n} = W_n(\theta)
\end{aligned}$$

$\square$

L'analogue du théorème de Kesten-Stigum existe mais un phénomène nouveau apparaît qui sépare l'espace en deux zones.

Théorème 7 *(Biggins 1977)*

Quand $n \to +\infty$, $W_n(\theta) \xrightarrow{L^1} W(\theta)$ *et* $\mathbb{E}(W(\theta)) = 1$
si et seulement si

$$\log m(\theta) - \theta \frac{m'(\theta)}{m(\theta)} > 0 \tag{5.10}$$

et

$$\mathbb{E}(\mathcal{Z}_1(\theta) \log \mathcal{Z}_1(\theta)) < \infty.$$

Dans le cas contraire, $W_n(\theta) \longrightarrow 0$ *presque sûrement.*

On voit que (5.10) définit deux zones de $\mathbb{R}$ limitées par deux valeurs critiques θ_c et θ'_c. Dans la zone $\theta_c < \theta < \theta'_c$ on a convergence L^1 sous "xlog x". Dans la zone $\theta < \theta_c$ ou $\theta > \theta'_c$ on a $W_n(\theta) \longrightarrow 0$.

Démonstration : plusieurs démonstrations existent, dont une utilisant l'extension au cas spatial de l'arbre biaisé (cf. Lyons [12]).

5.3.3 Un exemple de marche aléatoire branchante : la bisection

Le modèle de la bisection est appelé aussi "Kolmogorov's rock". Dans ce modèle, un objet est de masse 1 à l'instant 0. A l'instant 1, il se coupe en deux objets de masse U et $1 - U$ où U est une variable aléatoire de loi uniforme sur l'intervalle $]0, 1[$. Les deux morceaux évoluent ensuite indépendamment l'un de l'autre. A l'instant n, il y a 2^n morceaux, et chacun se coupe à l'instant $n+1$ en deux morceaux de taille respectivement U fois sa taille et $(1-U)$ fois sa taille. Ainsi la masse de chaque morceau est le produit de variables aléatoires de loi uniforme sur $]0, 1[$. En prenant les logarithmes de ces variables, on obtient un modèle additif, qui est une marche aléatoire branchante.

Avec les notations du paragraphe 5.3.2, soit U une variable aléatoire de loi uniforme sur $]0, 1[$. Soit le processus ponctuel $\mathcal{Z}$ défini par

$$\mathcal{Z} = \delta_{-\log U} + \delta_{-\log(1-U)} .$$

Il définit une marche aléatoire branchante dans laquelle l'ancêtre a exactement deux enfants ($N \equiv 2$) qui se déplacent en $\gamma_1 = -\log U$ et $\gamma_2 = -\log(1 - U)$.

Exercice : Calculer pour un paramètre θ réel, la transformée de Laplace $m(\theta)$ et son logarithme $\Lambda(\theta)$. Ecrire la martingale additive que l'on appellera $M_n^{BIS}(\theta)$. Traduire le théorème de Biggins, en particulier déterminer la zone de convergence L^1 de la martingale additive. En faisant le changement de variable $z = (1 + \theta)/2$, traduire cette zone sur un intervalle en z : $]z_-, z_+[$. Les bornes z_- et z_+ sont solution d'une équation que l'on écrira. Ecrire la martingale additive de la bisection en fonction du paramètre z.

Exercice : Une marche aléatoire branchante cousine de la bisection.

Soit U une variable aléatoire de loi uniforme sur $]0,1[$. Soit la marche aléatoire branchante définie par le processus ponctuel

$$\mathcal{Z} = 2\delta_{-\log U}\,.$$

Dans cette marche, l'ancêtre a exactement 2 descendants, comme dans la bisection, mais ils sont tous les deux situés au même endroit, en $\gamma_1 = \gamma_2 = -\log U$. Ecrire la transformée de Laplace $m(\theta)$ et son logarithme $\Lambda(\theta)$. Ecrire la martingale additive que l'on appellera $M_n^{GEN}(\theta)$. Que remarquez-vous ?

5.4 Lignes d'arrêt

Définition 3 *On dit qu'un sous-ensemble L de U a la propriété de ligne si et seulement si tout élément de L n'a pas d'ancêtre dans L, autrement dit si et seulement si pour tous éléments u et v de la ligne L on n'a ni $u < v$ ni $v < u$. S'il s'agit d'un sous-ensemble $L(\omega)$ d'un arbre ω ceci est encore équivalent à T_u et T_v disjoints pour tous éléments u et v de la ligne.*

Une ligne $L(\omega)$ définit l'ensemble $D(\omega)$ des descendants de la ligne

$$D(\omega) = \{u \in \omega, \exists v \in L(\omega), v \leq u\}$$

Si l'on considère la tribu engendrée par les N_u (et éventuellement les γ_u) pour tous les $u \notin D(\omega)$, on souhaite écrire une propriété de branchement plus générale. Pour cela on introduit la notion de ligne d'arrêt.

Pour un processus spatial ou plus généralement pour les arbres marqués, on définit $\mathcal{A}_u(s)$, la filtration attachée à chaque branche. Alors une ligne d'arrêt sera une collection de variables $\tau_u, u \in U$, définies chacune sur Ω_u et à valeurs dans $[0, \sigma_u]$ (où σ_u est la durée de vie de u) ou bien $\{0,1\}$ s'il s'agit d'un processus discret et vérifiant les deux conditions

(i) τ_u est un temps d'arrêt pour $\mathcal{A}_u(s)$,

(ii) l'ensemble $L_\tau(\omega) := \{u \in \omega, 0 \leq \tau_u < \sigma_u\}$ a la propriété de ligne.

Par exemple pour la marche aléatoire branchante, pour tout réel positif a, posons $\tau_u = 0$ si $X_u \geq a$ et pour tous les $v < u$, $X_v < a$, $\tau_u = 1$ sinon. Alors $(\tau_u, u \in U)$ est une ligne d'arrêt et $L_a := L_\tau = \{u, X_u \geq a, \forall v < u,\ X_v < a\}$ est la ligne des premières particules à atteindre la droite d'équation $x = a$. On peut bien entendu imaginer toutes sortes de lignes d'arrêt pertinentes suivant le modèle considéré. On vérifie facilement que la n-ième génération est une ligne d'arrêt ainsi que l'ensemble des particules en vie à l'instant t dans un processus continu.

Proposition 6 *Soit $\tau = \{\tau_u, u \in U\}$ une ligne d'arrêt et soit $\mathcal{F}_\tau$ la tribu associée. Alors conditionnellement à $\mathcal{F}_\tau$ les sous-arbres $T_u, u \in L_\tau$ sont indépendants et de même loi $\mathbb{P}_{X_u(\tau_u)}$.*

Cette propriété de branchement (analogue de la propriété de Markov forte sur les arbres) avec lignes d'arrêt est utilisée notamment dans [6] pour un processus de branchement brownien, dans [4] pour une marche aléatoire branchante ou dans [5] en théorie du contrôle.

Bibliographie

[1] S. Asmussen, H. Hering. Branching processes. Birkhauser. 1983

[2] K.B. Athreya, P. Ney. Branching processes. Springer. 1972

[3] J.D.Biggins. Martingale convergence in the branching random walk. Adv in Appl. Probab. 10, 62-84, 1978

[4] J.D.Biggins, A.E. Kyprianou. Seneta-Heyde norming in the branching random walk. Ann. Probab. **25**, no. 1, 337–360, 1997.

[5] C. Ceci, A. Gerardi. Filtering of a branching process given its split times. J. Appl. Prob. **55**, 27-50, 1997.

[6] B. Chauvin. Product martingales ans stopping lines for branching brownian motion. Ann. Probab. **19**, 3, 1195-1205. 1991.

[7] L. Comtet. Advanced Combinatorics. Reidel. Dordrecht. 1974

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest. *Introduction à l'algorithmique*. Dunod. 1990.

[9] P. Flajolet, R. Sedgewick. An introduction to analysis of algorithms. Addison-Wesley.

[10] D. Knuth. The art of computer programming. Vol. 1. Fundamental Algorithms. Addison-Wesley. Second edition 1973

[11] D. Knuth. The art of computer programming. Vol. 3. Sorting and searching. Addison-Wesley. 1973

[12] R. Lyons. A simple path to Biggins' martingale convergence for branching random walk. Classical and Modern Branching Processes. Ed. Athreya and Jagers. IMA Volumes in Mathematics and its Applications. Vol. 84. Springer. 1997.

[13] R. Lyons, R. Pemantle, Y. Peres. Conceptual proofs of $L \log L$ criteria for mean behavior of branching processes. *Ann. Probab.* **23**, 1125-1138, 1995.

[14] H.M. Mahmoud. Evolution of Random Search Trees. Wiley. 1992

[15] J. Neveu. Arbres et processus de Galton-Watson. *Ann. IHP* **22**, 2, 199-207, 1986

[16] Y.Peres. Probability on trees : an introductory climb. Cours Ecole d'été de Saint-Flour 1997. Lecture Notes in Mathematics 1717. Springer 1999.

[17] K. Uchiyama. Spatial growth of a branching process of particles living in $\mathbb{R}^d$. *Ann. Proba.*, **10**, 4, 896-918, 1982

Chapitre 6

Arbres binaires de recherche

Une référence sur le sujet est le livre de Mahmoud [9]

L'espace probabilisé utilisé est analogue à celui du chapitre précédent. Dans ce chapitre, les outils sont mis en place section 1. Dans les sections 2 et 3 on étudie l'arbre binaire de recherche par des raisonnements de type forward et dans la section 4 par des raisonnements de type backward.

6.1 Définitions

Au lieu de travailler sur un espace d'arbres planaires étiquetés par les suites finies d'entiers comme dans le chapitre 5, nous allons travailler sur un espace d'arbres binaires, étiquetés par

$$\mathbb{U} = \{\emptyset\} \cup \bigcup_{n\geq 1} \{0,1\}^n$$

qui est l'ensemble des mots sur l'alphabet à deux lettres $\{0,1\}$. Comme d'habitude, $\emptyset$ désigne le mot vide, et les mêmes notations que dans le chapitre 5 sont utilisées pour les mots de $\mathbb{U}$. Le fils gauche du mot u est $u0$ et le fils droit est $u1$.

Par définition, un arbre binaire complet T est un sous-ensemble fini de $\mathbb{U}$ tel que

$$\begin{cases} \emptyset \in T \\ \text{si } uv \in T \text{ alors } u \in T\,, \\ u1 \in T \Leftrightarrow u0 \in T\,. \end{cases}$$

Les éléments de T sont les noeuds, la racine est $\emptyset$, le nombre de lettres d'un nœud u est noté $|u|$, c'est la profondeur de u dans l'arbre ($|\emptyset| = 0$). On note **BinTree** l'espace des arbres binaires complets.

Notons qu'un arbre binaire complet $T \in$ **BinTree** peut être décrit par l'ensemble ∂T de ses feuilles, qui sont par définition les nœuds qui sont dans T mais qui n'ont pas de descendant. On appelle aussi les feuilles nœuds externes. Les autres nœuds de $T\backslash\partial T$ sont appelés les nœuds internes.

Introduisons maintenant les arbres binaires de recherche étiquetés (abre), qui sont très utilisés pour stocker des données ordonnées.

Soit A un ensemble totalement ordonné d'éléments appelés des clés, et pour tout $n \geq 1$, soit $(x_1, ..., x_n)$ tirés dans A sans remise. L'arbre binaire de recherche étiqueté construit à partir de ces données est l'arbre binaire complet dans lequel chaque nœud interne est muni d'une clé de la façon suivante : la première clé x_1 est mise à la racine. Puis la deuxième clé x_2 est assignée au fils gauche si elle est plus petite que x_1, elle est assignée au fils droit si elle est plus grande que x_1. On continue récursivement de cette façon. Après n insertions, on a un arbre binaire de recherche étiqueté dans lequel n nœuds ont une étiquette, ils sont considérés comme les nœuds internes. On ajoute $n+1$ feuilles (non étiquetées) pour avoir un arbre binaire complet étiqueté à n nœuds internes.

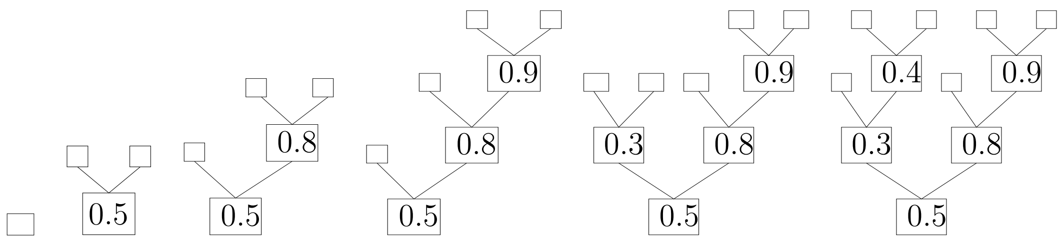


FIG. 6.1 – ABR étiqueté construit à partir de la suite de données 0.5, 0.8, 0.9, 0.3, 0.4 (les carrés vides sont les feuilles).

Une autre construction récursive équivalente est la suivante :
- x_1 est l'étiquette de la racine
- le sous-arbre gauche est l'abr associé à $\{x_2, \dots x_n\} \cap]-\infty, x_1[$
- le sous-arbre droit est l'abr associé à $\{x_2, \dots x_n\} \cap]x_1, +\infty[$

Remarque : différentes suites de données peuvent donner le même arbre. Par exemple, 0.5, 0.3, 0.8, 0.4, 0.9 donne le même arbre que dans la figure 1.

Introduisons maintenant le modèle probabiliste. On suppose que les données $(x_i)_{i\geq 1}$ successivement insérées sont des variables aléatoires indépendantes et équidistribuées ayant même loi continue F sur l'intervalle $[0,1]$. Sous ce modèle, appelons $L_n^{(F)}$ l'arbre binaire de recherche étiqueté ainsi obtenu ; c'est une variable aléatoire à valeurs dans l'ensemble des arbres binaires complets dans lesquels chaque nœud interne a une étiquette appartenant à $[0,1]$. La suite $(L_n^{(F)}, n \geq 0)$ est une chaine de Markov.

En fait, on est surtout intéressé par l'arbre sous-jacent $\mathcal{T}_n^{(F)}$ de $L_n^{(F)}$, c'est-à-dire l'arbre qui a la même structure arborescente que $L_n^{(F)}$, mais sans les étiquettes. On pose

$$\big(\mathcal{T}_n^{(F)}, n \geq 0\big) := \big(UNDER(L_n^{(F)}), n \geq 0\big);$$

par construction $\mathcal{T}_n^{(F)}$ est un arbre binaire complet.

Ce modèle probabiliste s'appelle "modèle des permutations uniformes" pour la raison suivante : pour tout $n \geq 1$, la suite $x_1, .., x_n$ induit presque sûrement une permutation σ_n (le réordonnement) telle que $x_{\sigma_n(1)} < x_{\sigma_n(2)} < \cdots < x_{\sigma_n(n)}$. Comme les x_i sont échangeables, la loi de σ_n est la loi uniforme sur l'ensemble $\mathcal{S}_n$ des permutations de $\{1, .., n\}$. Tout ceci ne dépend pas de la loi choisie pour F, donc sans perte de généralité nous pouvons supposer que F est la loi uniforme sur $[0,1]$ et on peut écrire L_n au lieu de $L_n^{(F)}$ et $\mathcal{T}_n$ au lieu de $\mathcal{T}_n^{(F)}$.

Attention : dans ce modèle des permutations uniformes (random permutation model), la loi uniforme est sur $\mathcal{S}_n$, le groupe des permutations, pas sur les arbres ; ce qui signifie que les abr ne sont pas équiprobables, à l'inverse des arbres de Catalan.

Evolution dynamique de l'arbre

Voyons comment se fait l'insertion de la $(n+1)$-ème donnée x_{n+1} dans l'arbre à n nœuds internes $\mathcal{T}_n$. Comme le réordonnement σ_n dépend de l'ordre mais pas des valeurs des $x_1, .., x_n$, la probabilité sachant $\mathcal{T}_n$ pour que la $(n+1)$-ème donnée x_{n+1} soit à une certaine place est

$$\begin{aligned} P\big(x_{n+1} \in (x_{\sigma_n(j)}, x_{\sigma_n(j+1)}) \,|\, \sigma_n\big) &= P\big(x_{n+1} \in (x_{\sigma_n(j)}, x_{\sigma_n(j+1)})\big) \\ &= P(\sigma_{n+1}(j+1) = n+1) = (n+1)^{-1} \end{aligned}$$

pour tout $j \in \{0, 1, .., n\}$, où $x_{\sigma_n(0)} := 0$ et $x_{\sigma_n(n+1)} := 1$.

On peut aussi exprimer cette relation avec les rangs R_k des données x_k : les variables aléatoires $R_k = \sum_{j=1}^{k} \mathbb{1}_{x_j \leq x_k}, k \geq 1$ sont indépendantes et R_k est de loi uniforme sur $\{1, \ldots, k\}$, de sorte que $P(R_{n+1} = j+1 \,|\, R_1, .., R_n) = (n+1)^{-1}$.

En résumé en termes d'évolution dynamique de l'abr, **l'insertion de la $n+1$-ème donnée est uniforme parmi les $n+1$ feuilles**. Dans ce modèle probabiliste, la suite $(\mathcal{T}_n)_{n\geq 0}$ est une chaine de Markov à valeurs dans **BinTree** définie par $\mathcal{T}_0 = \{\emptyset\}$ et

$$\begin{aligned} \mathcal{T}_{n+1} &= \mathcal{T}_n \cup \{D_n0, D_n1\}\,, \\ P(D_n = u \,|\, \mathcal{T}_n) &= (n+1)^{-1}, \quad u \in \partial\mathcal{T}_n\,; \end{aligned} \tag{6.1}$$

la feuille D_n de $\mathcal{T}_n$ est le nœud aléatoire sur lequel la $n+1$-ème donnée est insérée, au niveau d_n.

Notons que l'évolution de l'arbre étiqueté L_n (qui dépend fortement des valeurs $x_1, \ldots, x_n$ des clés déjà insérées) est différente de l'évolution de l'arbre $\mathcal{T}_n$ (qui n'en dépend pas). C'est le même phénomène que dans les chaines de Markov en milieu aléatoire.

Variables de coût : profondeur, hauteur

Remarquons que lorsqu'on insère une $n+1$-ème donnée, on la compare à la racine, puis à la clé racine du sous-arbre gauche (ou droit), etc... on ne la compare pas à toutes les clés mais seulement le long d'une branche, jusqu'à l'insérer sur une feuille, au niveau d_n. Ce nombre est ainsi le nombre de comparaisons effectuées, c'est une variable intéressante à étudier. Par ailleurs, cet entier est toujours inférieur ou égal à h_n (cas le pire), où h_n est la hauteur de l'arbre $\mathcal{T}_n$:

$$h_n := \max_{u \in \mathcal{T}_n} |\, u \,|$$

et est une autre variable aléatoire intéressante à étudier. On a toujours, en considérant les deux cas extrêmes

$$\lfloor log_2 n \rfloor \leq h_n \leq n-1$$

et on verra plus loin que l'ordre de grandeur de la hauteur est (heureusement) $\log n$.

6.2 Connection abr-marche aléatoire branchante

Nous allons d'abord décrire cette connection, puis l'utiliser d'une part dans l'étude de la hauteur des abr, d'autre part pour démontrer que certaines martingales associées à l'abr et à la bisection sont égales.

Les résultats sur la hauteur sont essentiellement dus à Devroye [5] et ils peuvent pour certains être compris, retrouvés ou améliorés à partir des résultats de Biggins [1] sur les marches aléatoires branchantes. Les résultats sur les martingales sont dans [3] et [4].

6.2.1 La connection

Rappelons que $\mathcal{T}_n$ est l'abr de taille n. Pour tout nœud u de $\mathcal{T}_n$, appelons $V_u^{(n)} =\mid \mathcal{T}_u \mid$ la taille (c'est-à-dire le nombre de nœuds) du sous-arbre issu de u. Pour la racine (notée $\emptyset$), $V_\emptyset^{(n)} = n$ et si $V_0^{(n)} =\mid \mathcal{T}_0 \mid$, alors $V_1^{(n)} =\mid \mathcal{T}_1 \mid= n-1-V_0^{(n)}$. On sait que $V_0^{(n)}$ est uniforme sur $\{0, 1, \ldots, n-1\}$ à cause du modèle probabiliste choisi.

Maintenant attachons à tous les nœuds u des variables aléatoires indépendantes U_u, de loi uniforme sur $[0, 1]$. Alors un calcul immédiat conduit au

Lemme 5 *: $V_0^{(n)}$ a même loi que $\lfloor nU_\emptyset \rfloor$; en corollaire, $n-1-V_0^{(n)} = V_1^{(n)}$ a même loi que $\lfloor n(1-U_\emptyset)\rfloor$.*

De la même façon, on obtient que pour n'importe quel nœud u de l'arbre, le sous-arbre gauche est de taille $V_{u0}^{(n)}$ qui a même loi que $\lfloor V_u^{(n)} U_u \rfloor$ et le sous-arbre droit est de taille $V_{u1}^{(n)}$ qui a même loi que $\lfloor V_u^{(n)}(1-U_u)\rfloor$. Finalement, pour un nœud à distance k de la racine, c'est-à-dire à profondeur k, u s'écrit $u = u_1 u_2 \ldots u_k$ et $V_u^{(n)}$ a même loi qu'un produit du type $\lfloor \ldots \lfloor \lfloor n\tilde{U}_\emptyset \rfloor \tilde{U}_{u_1} \rfloor \ldots \tilde{U}_{u_1 \ldots u_k} \rfloor$ (où l'on écrit $\tilde{U}_v$ au lieu de U_v pour un nœud v qui est une branche gauche et $1-U_v$ pour un nœud v qui est une branche droite).

La proposition suivante admet une version en loi (Devroye) et une version presque sûre, sa démonstration est admise (le principe de la démonstration du résultat p.s. est d'obtenir le résultat sur la version continue de l'abr, puis de le projeter sur l'abr par le principe de plongement).

Proposition 7 *Pour tout nœud u de l'abr $\mathcal{T}_n$, soit $V_u^{(n)}$ la taille du sous-arbre issu de u, alors p.s. quand $n \to +\infty$*

$$\frac{V_{u0}^{(n)}}{V_u^{(n)}} \longrightarrow U_{u0} \qquad et \qquad \frac{V_{u1}^{(n)}}{V_u^{(n)}} \longrightarrow U_{u1}$$

où U_{u0} et U_{u1} sont des variables aléatoires de loi uniforme sur l'intervalle $[0, 1]$ et $U_{u0} + U_{u1} = 1$. De plus, pour tous v et w de $\mathcal{T}_n$, v et w non frères, U_v et U_w sont indépendants.

Par conséquent p.s. quand $n \to +\infty$

$$\frac{V_u^{(n)}}{n} \longrightarrow \prod_{v \leq u} U_v$$

où les $U_v, v \leq u$ sont i.i.d. de loi uniforme sur $[0, 1]$ et $\forall v, U_{v0} + U_{v1} = 1$.

C'est cette proposition qui permet de relier des variables définies sur l'abr (les $V_u^{(n)}$) à des variables uniformes i.i.d. vérifiant la propriété $U_{u0} + U_{u1} = 1$, ce qui correspond à la bisection étudiée au paragraphe 5.3.3.

6.2.2 Hauteur des abr

La loi de $V_u^{(n)}$ est également digne d'intérêt car la loi de la hauteur h_n de l'arbre est reliée aux $V_u^{(n)}$ par la dualité (évidente) suivante :

Proposition 8

$$\{h_n \geq k\} \;=\; \{\max_{u,|u|=k} V_u^{(n)} \geq 1\}.$$

Attention : le calcul de la loi de h_n ne sera pas simple, car les $V_u^{(n)}$ sont très dépendants les uns des autres. Néanmoins il est clair grâce à la proposition précédente que la loi de la hauteur h_n est asymptotiquement (quand $n \to +\infty$) donnée par $\max_{u,|u|=k} \prod_{v \leq u} U_v$. Or dans la bisection (cf. paragraphe 5.3.3), la position X_u de la particule u est donnée par

$$X_u = -\log(\prod_{v\leq u} U_v)$$

$$e^{-X_u} = \prod_{v\leq u} U_v\,,$$

donc le maximum ci-dessus est donné par le minimum pour $|\ u\ | = k$ des positions X_u dans la bisection. Or le comportement du minimum des positions dans une marche aléatoire branchante a été élucidé par Biggins et l'on obtient

Théorème 8 *Quand $n \to +\infty$,*

$$\frac{h_n}{\log n} \longrightarrow c \quad p.s. \qquad \frac{h'_n}{\log n} \longrightarrow c' \quad p.s$$

où c et c' sont solutions de l'équation

$$x \log \frac{2e}{x} = 1.$$

où de façon duale, on a introduit h'_n le premier niveau d'apparition d'un noeud externe (c'est aussi le coût minimal d'insertion) ; h'_n s'appelle niveau de saturation de l'abr.

La conjecture dite de Robson disant que **la variance de la hauteur d'un arbre binaire de recherche est d'ordre de grandeur 1**, a longtemps tenu mais est maintenant résolue, et par plusieurs méthodes (Devroye, Reed [6], Drmota [7]). Des conjectures sont encore non résolues, sur la concentration de la hauteur autour de sa moyenne ou de sa médiane ou sur l'ordre de grandeur du nombre de feuilles au dernier niveau.

6.3 Profil des abr

6.3.1 Notations

Outre la hauteur et le niveau de saturation, si l'on veut connaître plus finement la répartition des noeuds par niveau dans l'arbre binaire de recherche, on introduit naturellement

$$U_k(n) := \text{le nombre de noeuds externes au niveau } k \text{ dans l'arbre } \mathcal{T}_n$$

et la collection des $U_k(n), k \geq 0$ s'appelle le <u>profil</u> de l'arbre.

On pourrait travailler de façon analogue avec les noeuds internes et introduire

$$V_k(n) := \text{le nombre de noeuds internes au niveau } k \text{ dans l'arbre } \mathcal{T}_n.$$

ainsi que

$$\begin{aligned} Z_k(n) &:= \text{le nombre total de noeuds au niveau } k \text{ dans l'arbre } \mathcal{T}_n \\ Z_k(n) &= U_k(n) + V_k(n). \end{aligned}$$

Comme il y a $n+1$ noeuds externes dans l'arbre $\mathcal{T}_n$ de taille n, la mesure empirique de répartition des noeuds externes est

$$\nu_n := \sum_{k=1}^{+\infty} \frac{U_k(n)}{n+1} \, \delta_{\{k\}}.$$

Comme la $(n+1)$-ième insertion se fait uniformément sur les $n+1$ noeuds externes de l'arbre $\mathcal{T}_n$, la loi du niveau d'insertion, ou profondeur d_n est donnée par

$$\mathbb{P}(d_n = k \mid \mathcal{T}_n) = \frac{U_k(n)}{n+1} = \nu_n(k), \tag{6.2}$$

et en prenant l'espérance

$$\mathbb{P}(d_n = k) = \mathbb{E}(\frac{U_k(n)}{n+1}) = \mathbb{E}(\nu_n(k))$$

autrement dit la loi de d_n est ce qu'on appelle la mesure d'intensité (déterministe) du processus ponctuel ν_n, par conséquent **les résultats en moyenne sur la mesure ν_n donneront des résultats en loi sur d_n.**

6.3.2 Résultats en moyenne sur le profil

Théorème 9 *(Lynch, 1965)*

$$\mathbb{E}U_k(n) = \frac{2^k}{n!} \begin{bmatrix} n \\ k \end{bmatrix}$$

où $\begin{bmatrix} n \\ k \end{bmatrix}$ sont les nombres de Stirling de première espèce, c'est-à-dire que $\begin{bmatrix} n \\ k \end{bmatrix}$ est le coefficient de x^k dans le développement de $x(x+1)\dots(x+n-1)$. Ces nombres vérifient la relation de récurrence

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + n \begin{bmatrix} n \\ k \end{bmatrix}.$$

Corollaire 1 *La loi de d_n est donnée par*

$$\mathbb{P}(d_n = k) = \frac{2^k}{(n+1)!} \begin{bmatrix} n \\ k \end{bmatrix}.$$

Preuve du théorème 9

On utilise fortement une sorte de propriété de branchement, ici le conditionnement par la taille du sous-arbre gauche $\mathcal{T}_0$, qui dit que : sachant la taille k du sous-arbre gauche $\mathcal{T}_0$, alors le couple $(\mathcal{T}_0, \mathcal{T}_1)$ a même loi que deux abr $\mathcal{T}_k$ et $\mathcal{T}_{n-1-k}$ de taille respectivement k et $n-1-k$, et indépendants. Posant

$$a_{n,k} := \mathbb{E}U_k(n)\ \frac{n!}{2^k},$$

un petit calcul permet de voir que les $a_{n,k}$ satisfont la relation de récurrence des nombres de Stirling et ont les mêmes valeurs initiales. □

6.3.3 Résultats p.s. sur le profil

On voit facilement que pour tout niveau k de l'arbre $\mathcal{T}_n$,

$$Z_{k+1}(n) = 2V_k(n)$$

et donc

$$Z_{k+1}(n) - Z_k(n) = V_k(n) - U_k(n)$$

Comme au plus bas niveau de l'arbre, il n'y a que des nœuds externes

$$U_{1+h_n}(n) = Z_{1+h_n}(n) \text{ et } V_{1+h_n}(n) = 0$$

et par une récurrence descendante sur k :

$$Z_k(n) = \sum_{j=k}^{1+h_n} \frac{U_j(n)}{2^{j-k}}$$

qui est une quantité du type “polynôme de niveau”.

Définition 4 *On appelle* **polynôme de niveau** *d'un arbre binaire de recherche $\mathcal{T}_n$, le polynôme défini pour tout paramètre $z \in \mathbb{C}$ par*

$$W_n(z) := \sum_{k=0}^{+\infty} U_k(n)\ z^k$$

Il s'agit bien d'un polynôme, car pour $k > 1 + h_n, U_k(n) = 0$. Bien entendu, c'est une variable aléatoire, puisque les $U_k(n)$ sont aléatoires.

6.3.4 Résultats en moyenne sur le polynôme de niveau

On obtient très facilement la moyenne du polynôme de niveau, car

$$\mathbb{E}W_n(z) = \sum_{k=0}^{+\infty} \mathbb{E}U_k(n)\ z^k$$

et d'après le théorème 9,

$$\begin{aligned}
\mathbb{E}W_n(z) &= \frac{1}{n!}\sum_{k=0}^{+\infty} 2^k z^k \begin{bmatrix} n \\ k \end{bmatrix} \\
&= \frac{1}{n!}2z(2z+1)\dots(2z+n-1) \\
&= \prod_{j=0}^{n-1}\frac{j+2z}{j+1} .
\end{aligned}$$

La loi de la profondeur est alors immédiate, car si $d_n(z)$ est sa série génératrice

$$d_n(z) = \sum_{k\geq 0} \mathbb{P}(d_n = k)z^k = \mathbb{E}(z^{d_n}) \ ,$$

comme $\mathbb{P}(d_n = k) = \frac{1}{n+1}\mathbb{E}U_k(n)$ (cf. equation (6.2)), on obtient

$$d_n(z) = \frac{1}{n+1}\mathbb{E}W_n(z) = \frac{1}{(n+1)!}\prod_{j=0}^{n-1}(j+2z)$$

qui est une expression assez explicite de la série génératrice de la profondeur d'insertion.

<u>Exercice</u> En déduire que

$$\mathbb{E}(d_n) = 2(H_{n+1} - 1)$$

et calculer $Var(d_n)$.

6.3.5 Résultats p.s. sur le polynôme de niveau

La relation évidente

$$U_k(n+1) - U_k(n) = -1\!\!1_{\{d_n = k\}} + 2\ 1\!\!1_{\{d_n = k-1\}}$$

permet d'avoir une relation de récurrence sur le "polynôme de niveau" :

$$\begin{aligned}
\mathbb{E}(W_{n+1}(z) \mid \mathcal{T}_n) &= \sum_{k\geq 0} z^k \Big[U_k(n) + 2\mathbb{P}(d_n = k-1 \mid \mathcal{T}_n) - \mathbb{P}(d_n = k \mid \mathcal{T}_n) \Big] \\
&= \frac{n+2z}{n+1}\ W_n(z) \ ,
\end{aligned}$$

ce qui signifie que le polynôme de niveau convenablement renormalisé est une martingale.

Théorème 10 *pour tout nombre complexe $z \in \mathbb{C}$,*

$$M_n(z) := \frac{W_n(z)}{\mathbb{E}(W_n(z))} = \frac{W_n(z)}{\prod_{j=0}^{n-1}\frac{j+2z}{j+1}}$$

est une $\mathcal{F}_n$-martingale. En notant

$$C_n(z) := \prod_{j=0}^{n-1} \frac{j+2z}{j+1} \,,$$

elle s'écrit aussi

$$M_n(z) := \frac{1}{C_n(z)} \sum_{u\in\partial\mathcal{T}_n} z^{|u|}$$

Elle converge p.s. pour tout z réel positif. Elle converge dans L^1 sur l'intervalle réel $]c'/2; c/2[$ (les constantes c et c' étant celles du théorème 8). Elle converge p.s. vers 0 en dehors de cet intervalle, y compris aux points critiques $c'/2$ et $c/2$. Elle converge dans L^2 sur la boule $B(1, \frac{1}{\sqrt{2}})$ de $\mathbb{C}$.

Idées de démonstration

La propriété de martingale vient du calcul ci-dessus. La convergence presque sûre dans $\mathbb{R}_+$ est celle de toute martingale positive. La convergence dans L^1 est plus difficile à voir. Elle est obtenue en bornant $M_n(z)$ dans L^p pour $p > 1$, de façon assez analogue à la méthode utilisée pour un processus de branchement spatial. Les détails sont dans [8]. La convergence L^2 pour z dans $\mathbb{C}$ s'obtient en calculant la covariance de la martingale et avec un peu d'analyse complexe (cf [2]). La convergence L^1 dans $\mathbb{C}$ et la convergence aux points critiques s'obtiennent par plongement en temps continu et se trouvent dans [3]. □

Globalement, cette étude tire parti de l'égalité

$$W_n(z) = \mathbb{E}(W_n(z))M_n(z)$$

qui permet d'étudier le polynôme de niveau en séparant la partie déterministe (son espérance qui est bien connue, cf. section précédente) et la partie aléatoire qui, étant une martingale, possède des propriétés agréables. C'est ainsi que des résultats de type théorème central limite et grandes déviations sur la mesure ν_n peuvent être obtenus, ainsi que l'ordre de grandeur de la largeur d'un arbre binaire de recherche. Appelons en effet $\overline{Z}_n$ la largeur de $\mathcal{T}_n$; c'est le maximum sur tous les niveaux du nombre de noeuds à chaque niveau :

$$\overline{Z}_n := \max_{k\geq 0} \ Z_k(n)$$

Théorème 11 *(largeur d'un arbre binaire de recherche)*

$$\frac{\overline{Z}_n}{n/\sqrt{\pi \log n}} = 1 + \mathcal{O}(\frac{1}{\sqrt{\log n}}) \quad p.s.$$

lorsque $n \to +\infty$.

6.3.6 Martingale de l'abr et martingale de la bisection

la martingale de l'abr du théorème 10 et la martingale de la bisection (cf. paragraphe 5.3.3) se ressemblent :

$$M_n^{abr}(z) = \frac{1}{C_n(z)} \sum_{u \in \partial \mathcal{T}_n} z^{|u|}$$

et

$$M_n^{BIS}(z) = \sum_{|u|=n} e^{(1-2z)X_u} z^{|u|}$$

(c'est la réponse à l'exercice posé en section 5.3.3...) et elles ont même domaine de convergence dans L^1. En fait on a le résultat suivant, qui exprime qu'elles se ressemblent beaucoup à n fixé, aux problèmes de mesurabilité près, et que leur limite quand $n \to \infty$ sont p.s. égales. La démonstration se trouve dans [4].

Théorème 12 *Pour tout $z \in]c'/2; c/2[$, si $\mathcal{F}_n^{BIS}$ est la filtration engendrée par les variables aléatoires uniformes qui définissent la bisection,*

$$\mathbb{E}(M_\infty^{abr}(z)/\mathcal{F}_n^{BIS}) = M_n^{BIS}(z)$$

et par conséquent

$$M_\infty^{abr}(z) = M_\infty^{BIS}(z)$$

6.4 Coût de construction d'un abr ; coût de recherche ; longueurs de cheminement.

6.4.1 Coût de construction ; longueur de cheminement interne

<u>Question</u> : combien de clés regarde-t-on avant de placer une nouvelle clé à la bonne place ? Dans un arbre dégénéré (correspondant au cas le pire), on examine $i-1$ clés avant d'insérer la i-ième clé donc on examine $N(N-1)/2$ nœuds pour construire un arbre de N noeuds. Dans le cas le meilleur, le nombre milieu est placé à la racine pour chaque sous-arbre, chaque sous-arbre est de taille $N/2$ et donc le coût de construction de l'arbre vérifie la relation (du type diviser pour régner) $T_N = 2T_{N/2} + N$ ce qui donne $T_N \sim N \log N$ comme nombre d'étapes pour construire l'arbre.

Etant donné un arbre binaire de recherche (construit ou partiellement construit), on aura soit à insérer une nouvelle clé, soit à chercher une clé, d'où un coût de construction et un coût de recherche, mesurés en nombre de comparaisons. Dans les deux cas, cela se fera par un unique chemin dans l'arbre, parcouru par un pointeur partant de la racine et qui chemine en éliminant à chaque comparaison avec une clé racine, le sous-arbre droit ou le sous-arbre gauche. On voit ainsi que le coût de construction d'un abr de taille n, compté en nombre de comparaisons, n'est autre que la longueur de cheminement interne I_n.

$$I_n = \sum_{u \in \mathcal{T}_n} |\, u \,| \quad .$$

Pour étudier la loi de I_n, on introduit assez naturellement la fonction génératrice bivariée

$$F(x,y) = \sum_{n,k\geq 0} \mathbb{P}(I_n = k) x^n y^k \ .$$

Comme d'habitude, on fait opérer une sorte de propriété de branchement pour obtenir une relation de récurrence sur I_n. En effet, appelons, en omettant l'indice n, T_1 et T_2 les sous-arbres respectivement gauche et droit de $\mathcal{T}_n$. Si T_1 est de taille p, alors T_2 est de taille $n-1-p$, $p = 0, \ldots, (n-1)$, et

$$I_n = I_p(T_1) + I_{n-1-p}(T_2) + (n-1) \tag{6.3}$$

et donc par l'indépendance conditionnelle des sous-arbres gauche et droit

$$\begin{aligned}
\mathbb{P}(I_n = k) &= \sum_{p=0}^{n-1} \mathbb{P}(\mid T_1 \mid = p)\mathbb{P}(I_n = k \ / \mid T_1 \mid = p) \\
&= \sum_{p=0}^{n-1} \frac{1}{n} \ \mathbb{P}(I_p(T_1) + I_{n-1-p}(T_2) + (n-1) = k \ / \mid T_1 \mid = p) \\
&= \sum_{p=0}^{n-1} \frac{1}{n} \sum_{k_1+k_2=k-(n-1)} \mathbb{P}(I_p = k_1)\mathbb{P}(I_{n-1-p} = k_2)
\end{aligned}$$

ce qui donne l'équation différentielle suivante

$$\frac{\partial F(x,y)}{\partial x} = F^2(xy, y) \ . \tag{6.4}$$

Cette équation informe en partie sur la loi de I_n, par exemple, en la reliant aux fonctions génératrices des moments de I_n. Faisons-le pour l'espérance (pour la variance, les calculs sont plus lourds). Appelons B_1 la fonction génératrice de la moyenne de I_n.

$$B_1(x) := \sum_{n\geq 0} \mathbb{E}(I_n) \ x^n$$

Alors

$$B_1(x) = \frac{\partial F(x,y)}{\partial y} \ |_{y=1}$$

et en dérivant par rapport à x

$$\begin{aligned}
B_1'(x) &= \frac{\partial}{\partial y} F^2(xy, y) \ |_{y=1} \\
&= 2xF(x,1)F^2(x,1) + 2F(x,1)B_1(x)
\end{aligned}$$

Or,

$$F(x,1) = \sum_{n,k\geq 0} \mathbb{P}(I_n = k) x^n = \frac{1}{1-x}$$

ce qui montre que B_1 est solution de l'équation différentielle

$$B_1'(x) = \frac{2x}{(1-x)^3} + \frac{2B_1(x)}{1-x} \tag{6.5}$$

équation qui se résout sans problème, en rajoutant les conditions initiales $I_0 = 0$ et $I_1 = 0$:

$$B_1(x) = \frac{-2}{(1-x)^2}\Big[x + \log(1-x)\Big] .$$

Pour obtenir $\mathbb{E}(I_n)$, il suffit alors d'extraire le coefficient de x^n dans $B_1(x)$. On peut par exemple écrire $B_1(x)$ comme le produit de deux séries

$$\begin{aligned} B_1(x) &= \frac{2}{(1-x)^2}\Big[\frac{x^2}{2} + \frac{x^3}{3} + \cdots + \frac{x^n}{n} + \ldots\Big] \\ &= 2x^2\Big[\sum_{j\geq 0}(j+1)x^j\Big]\Big[\sum_{k\geq 0}\frac{1}{k+2}x^k\Big] \end{aligned}$$

et le coefficient de x^n est donc donné par

$$\mathbb{E}(I_{n+2}) = 2\sum_{k=0}^{n}\frac{n-k+1}{k+2}$$

qui s'écrit encore

$$\mathbb{E}(I_n) = 2(n+1)H_n - 4n = 2(n+1)(H_{n+1}-1) - 2n$$

où H_n désigne la somme des inverses des n premiers entiers.

<u>Remarque</u>

On peut trouver directement le coût de construction moyen d'un abr de taille n, c'est-à-dire la longueur de cheminement interne moyenne $\mathbb{E}(I_n)$, en voyant que celle-ci vérifie l'équation de récurrence déduite de l'équation (6.3)

$$\mathbb{E}(I_n) = n - 1 + \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}(I_{k-1}) + \mathbb{E}(I_{n-k})$$

que l'on reconnaît, c'est l'équation de récurrence de l'algorithme de tri rapide. Elle se résout en

$$\mathbb{E}(I_n) = 2(n+1)(H_{n+1}-1) - 2n$$

où H_n désigne la somme des inverses des n premiers entiers. Cela montre que le coût moyen de construction d'un abr est d'ordre de grandeur $n \log n$ (comme dans le cas le meilleur !).

6.4.2 Coût de recherche dans un abr ; longueur de cheminement externe

Si E_n est la longueur de cheminement externe, c'est-à-dire la somme des longueurs de u pour tous les u noeuds externes de l'arbre $\mathcal{T}_n$ et si I_n est la longueur de cheminement interne, alors ces deux variables sont reliées par

Lemme 6

$$E_n = I_n + 2n$$

Démonstration : par récurrence sur n, en détaillant le mécanisme de passage de $\mathcal{T}_n$ à $\mathcal{T}_{n+1}$. On peut aussi remarquer que

$$\begin{aligned} E_{n+1} &= E_n - d_n + 2(1+d_n) = E_n + d_n + 2 \\ I_{n+1} &= I_n + d_n \end{aligned}$$

□

On dit qu'une recherche est réussie quand la clé objet de la recherche est trouvée. On dit que la recherche est vaine si elle se termine sur un noeud externe. On appelle d_n la profondeur (le niveau) d'insertion de la n+1-ième donnée, c'est aussi le coût d'une recherche vaine. Il peut arriver selon les configurations d'ordinateurs que ce coût de recherche soit considérable et non négligeable par rapport au coût de construction de l'abr.

Résultats en moyenne sur la profondeur d'insertion

Le coût moyen d'une recherche vaine est ainsi égal à la profondeur moyenne d'insertion

$$\mathbb{E}(d_n) = \frac{1}{n+1}\mathbb{E}(E_n)$$

et d'après le lemme 6

$$\mathbb{E}(d_n) = \frac{1}{n+1}(\mathbb{E}(I_n) + 2n)$$

soit encore, par les évaluations ci-dessus,

$$\mathbb{E}(d_n) = 2(H_{n+1} - 1)$$

ce qui montre que le coût moyen de recherche est d'ordre de grandeur $2\log n$. On retrouve bien entendu le résultat obtenu par la méthode de séries génératrices de la section 3.4.

Résultats p.s. sur la longueur de cheminement externe

Rappelons que

$$E_n = \sum_{u \in \partial\mathcal{T}_n} \mid u \mid$$

soit encore

$$E_n = \sum_{k \geq 0} k\ U_k(n)$$

de sorte que

$$E_n = W_n'(1)$$

où W_n est le polynôme de niveau.

Comme la dérivée d'une martingale est encore une martingale, il suffit de calculer $M_n'(z)$ pour $z \in \mathbb{R}$, pour obtenir que

$$M_n'(1) = \frac{W_n'(1)}{n+1} - \frac{\mathbb{E}W_n'(1)}{n+1} = \frac{W_n'(1)}{n+1} - 2(H_{n+1} - 1)$$

est une martingale, soit le

Théorème 13 *(Regnier, 1989)*
$\frac{1}{n+1}(E_n - \mathbb{E}(E_n))$ *est une* $\mathcal{F}_n$*-martingale.*

On a, bien entendu, des résultats analogues pour la longueur de cheminement interne I_n.

Loi limite du coût de Quicksort, méthode de contraction

On appelle "coût de Quicksort" le coût de recherche dans un abr ou bien la longueur de cheminement externe, c'est-à-dire la variable E_n que nous avons déjà regardée. Outre les résultats en moyenne et de type martingale vus dans les paragraphes précédents, la question se pose de la **loi limite** de ce coût renormalisé :

$$Y_n := \frac{1}{n}(E_n - \mathbb{E}(E_n)) \tag{6.6}$$

Dans ce paragraphe, par abus de notation, on écrira parfois Y_n au lieu de $\mathcal{L}(Y_n)$. Remarquons que c'est aussi $M'_n(1)$ où M_n est la martingale de l'abr.

Le résultat du théorème suivant repose sur la **méthode**, dite **de contraction** :

Etape 1 - On montre que Y_n vérifie une certaine équation en loi et lorsque $n \to +\infty$, la loi limite vérifie une equation du type (6.8), c'est-à-dire une equation de point fixe sur les lois de probabilités F : $S(F) = F$ où S est une transformation qui dépend du modèle.

Etape 2 - On munit l'espace M des mesures de probabilités d'une distance qui rende cet espace métrique et complet, c'est-à-dire de Banach. Par exemple, ce pourra être la distance de Wasserstein d_2 (qui est la distance de Mallow d_p pour $p = 2$).

Etape 3 - On étudie la transformation

$$S : F \longmapsto S(F) \tag{6.7}$$

comme un opérateur de (M, d_2) dans (M, d_2), avec pour objectif de démontrer que S est une **contraction**.

Etape 4 - Grâce aux étapes 2 et 3, on peut appliquer le théorème du point fixe (ou théorème de Banach) à la transformation S de (M, d_2) dans (M, d_2). On en déduit que la transformation S a un unique point fixe, solution de l'équation (6.8), et que la suite $(S^n(F))$ converge vers ce point fixe (toujours pour la distance d_2), à une vitesse exponentielle.

Remarquons bien qu'à ce stade, on aura obtenu la loi limite comme unique solution d'une équation de point fixe, mais que rien ne dit que la loi de Y_n ressemble à $(S^n(F))$.

Voyons ce que cette méthode permet d'obtenir pour Quicksort (cf. [11], [12]).

Théorème 14 *(Roesler, 1992) La loi du coût renormalisé de Quicksort Y_n, converge vers la loi d'une variable aléatoire Y, unique solution de l'équation (en loi)*

$$Y = \tau Y + (1 - \tau)\bar{Y} + C(\tau) \tag{6.8}$$

où

Y et $\bar{Y}$ sont indépendantes et de même loi
τ suit une loi uniforme sur $[0, 1]$
(C, τ) et les Y sont indépendantes
$C(x) = 1 + 2(x \log x + (1 - x) \log(1 - x))$

Démonstration

Etape 1 : Cherchons d'abord l'équation vérifiée par $\mathcal{L}(Y_n)$: rappelons que le coût de recherche E_n vérifie la relation de récurrence (équation en loi)

$$E_n = E_{Z_n-1} + \bar{E}_{n-Z_n} + n - 1$$

où Z_n suit une loi uniforme sur les entiers $\{1, 2, \ldots, n\}$. Il est alors facile d'en déduire l'équation en loi vérifiée par la variable renormalisée Y_n. Posons pour $i = 1, 2, \ldots, n$

$$C_n(i) = \frac{1}{n}\big[n - 1 + \mathbb{E}(E_{i-1}) + \mathbb{E}(E_{n-i}) - \mathbb{E}(E_n)\big]$$

de sorte que

$$C_n(Z_n) = \frac{1}{n}\big[n - 1 + \mathbb{E}(E_{Z_n-1}) + \mathbb{E}(E_{n-Z_n}) - \mathbb{E}(E_n)\big]$$

Alors

$$\begin{aligned}
Y_n &= \frac{E_n - \mathbb{E}(E_n)}{n} \\
&= \frac{1}{n}\big[E_{Z_n-1} + \bar{E}_{n-Z_n} + n - 1 - \mathbb{E}(E_n)\big] \\
&= \frac{1}{n}\big[(Z_n - 1)Y_{Z_n-1} + \mathbb{E}(E_{Z_n-1}) + (n - Z_n)\bar{Y}_{n-Z_n} + \mathbb{E}(E_{n-Z_n}) + n - 1 - \mathbb{E}(E_n)\big] \\
&= \frac{Z_n - 1}{n} Y_{Z_n-1} + \frac{n - Z_n}{n}\bar{Y}_{n-Z_n} + C_n(Z_n)
\end{aligned}$$

qui est l'équation cherchée.Voyons ensuite que

Lemme 7

$$\frac{Z_n}{n} \longrightarrow \; \textit{Unif}\,[0,1] \qquad \textit{en loi}$$

et

Lemme 8

$$\forall x \in]0,1[,\; C_n(nx) \longrightarrow C(x)\;, n \to +\infty$$

où

$$C(x) := 1 + 2(x\log x + (1-x)\log(1-x))$$

Le Lemme 7 est facile à démontrer, le lemme 8 revient à un calcul explicite puisque les $C_n(i)$ font intervenir les moyennes connues

$$\mathbb{E}(E_i) = 2(H_{i+1} - 1)(i + 1).$$

Ces deux lemmes permettent de passer à la limite dans l'équation sur Y_n (remarquons que les propriétés de martingale notamment de Y_n nous assuraient déjà cette convergence en loi) et on obtient ainsi que Y_n converge en loi vers Y solution de

$$Y = \tau Y + (1-\tau)\bar{Y} + C(\tau)$$

qui peut s'écrire $F = S(F)$ avec S qui est la transformation

$$F \longmapsto S(F) = \mathcal{L}(\tau X + (1-\tau)\bar{X} + C(\tau))$$

Etape 2 : On définit sur l'espace M des mesures de probabilités, pour toute $F, G \in M$,

$$d_2(F,G) = \inf\{\|X - Y\|_2, X \text{ de loi } F, Y \text{ de loi } G\}$$

où

$$\|X - Y\|_2 = \mathbb{E}(\mid X - Y \mid^2)^{1/2}$$

est la norme L^2. On peut voir que d_2 est une distance (distance de Wasserstein) qui rend (M, d_2) métrique et **complet**. On peut exprimer aussi cette distance sous forme intégrale, en montrant (cf. [10], th. 8.1) que l'inf ci-dessus est atteint pour une variable aléatoire $X = F^{-1}(U)$ et pour une variable aléatoire $Y = G^{-1}(U)$ où U est **la même** variable aléatoire de loi uniforme sur $[0,1]$, de sorte que

$$\begin{aligned} d_2(F,G) &= \|F^{-1}(U) - G^{-1}(U)\|_2 \\ &= \Big(\int_0^1 \mid F^{-1}(u) - G^{-1}(u) \mid^2 \; du\Big)^{1/2}. \end{aligned}$$

La convergence pour la distance d_2 revient à la convergence faible **et** la convergence des moments d'ordre 2.

Etape 3 :

Lemme 9 *Lemme de contraction*
La transformation

$$F \longmapsto S(F) = \mathcal{L}(\tau X + (1-\tau)\bar{X} + C(\tau))$$

est contractante de (M, d_2) *dans* (M, d_2) *car*

$$d_2^2(S(F^0), S(G^0)) \leq \mathbb{E}[\tau^2 + (1-\tau)^2] \; d_2^2(F^0, G^0)$$

où F^0 *est la loi* F *translatée de telle sorte que* $\mathbb{E}(F^0) = 0$.

Remarque : si τ est de loi uniforme sur $[0,1]$,

$$\mathbb{E}[\tau^2 + (1-\tau)^2] = \int_0^1 x^2 \; dx = \frac{2}{3} < 1$$

Démonstration du lemme de contraction : Rappelons que

$$d_2^2(S(F), S(G)) = \inf\{\|X - Y\|_2, X \text{ de loi } S(F), Y \text{ de loi } S(G)\}.$$

Soient X_1 et X_2 indépendantes de loi F et soient Y_1 et Y_2 indépendantes de loi G. Alors, puisque d_2 est un inf,

$$\begin{aligned} d_2^2(S(F), S(G)) &\leq \|\tau X_1 + (1-\tau)X_2 + C(\tau) - \tau Y_1 - (1-\tau)Y_2 - C(\tau)\|_2 \\ &= \mathbb{E}\big[\tau(X_1 - Y_1) + (1-\tau)(X_2 - Y_2)\big]^2 \\ &= \mathbb{E}\big(\tau\big[X_1 - \mathbb{E}X_1 - (Y_1 - \mathbb{E}Y_1)\big] + \tau[\mathbb{E}X_1 - \mathbb{E}Y_1] + \\ &\quad +(1-\tau)\big[X_2 - \mathbb{E}X_2 - (Y_2 - \mathbb{E}Y_2)\big] + (1-\tau)[\mathbb{E}X_2 - \mathbb{E}Y_2]\big)^2 \end{aligned}$$

et en développant le carré, grâce à l'indépendance et aux espérances nulles des variables recentrées, on obtient :

$$d_2^2(S(F), S(G)) \leq \mathbb{E}[\tau^2 + (1-\tau)^2]\ \mathbb{E}(X_1 - \mathbb{E}X_1 - (Y_1 - \mathbb{E}Y_1))^2 + (\mathbb{E}X - \mathbb{E}Y)^2$$

ce qui permet de se ramener à F^0 et G^0 et donc

$$d_2^2(S(F^0), S(G^0)) \leq \mathbb{E}[\tau^2 + (1-\tau)^2]\ d_2^2(F^0, G^0)$$

□

<u>Etape 4 :</u> Le théorème du point fixe (théorème de Banach) pour une contraction S de (M, d_2) dans (M, d_2) , de taux de contraction $K < 1$, dit que la suite $S^n(F), n \in \mathbb{N}$, est une suite de Cauchy qui converge (car (M, d_2) est complet) à la vitesse K^n vers une limite qui est l'unique solution de l'équation de point fixe $S(F) = F$.

Ce qui termine la démonstration du théorème de Roesler. □

<u>Représentation du point fixe :</u>

Nous avons déjà dit que le théorème et plus généralement la méthode de contraction, n'implique pas de relation entre $\mathcal{L}(Y_n)$ et $S^n(F)$. Dans le cas de Quicksort, tout ce que l'on peut faire est d'avoir une représentation de $S^n(F)$ comme loi d'une variable aléatoire, disons, F_n, qui est une somme de variables aléatoires. Le lemme suivant s'obtient sans difficulté, par récurrence.

Lemme 10

$$S^n(F) = \mathcal{L}(F_n)$$

où

$$F_n = C_\emptyset + \sum_{k<n} \sum_{|u|=k} C_u A_u + \sum_{|u|=n} A_u X_u$$

où les $(X_u, u \in U)$ *sont i.i.d. de loi* F*, les* (C_u, U_u) *sont i.i.d. avec* U_u *de loi uniforme sur* $[0,1]$ *et* $C_u = C(U_u)$*, indépendants des* X_u*, et où*

$$A_u = \prod_{v \leq u} \tilde{U}_v$$

avec la notation habituelle des $\tilde{U}$*, déjà utilisée au paragraphe 6.2.*

Bibliographie

[1] J. D. Biggins. (1997). How fast does a general branching random walk spread ?. Classical and modern branching processes. IMA Vol. Math. Appl. **84** 19–39.

[2] B. Chauvin, M. Drmota and J. Jabbour-Hattab. (2001) The profile of binary search trees. Annals of Applied Probability, **11**, 1042-1062.

[3] B. Chauvin, T. Klein, J.F. Marckert, and A. Rouault. (2004) Martingales and Profile of Binary Search Trees. Preprint available at http ://fermat.math.uvsq.fr/∼marckert/papers.html

[4] B. Chauvin and A. Rouault. (2004) Connecting Yule process, Bisection and Binary Search Tree via Martingales. JIRSS.

[5] L. Devroye. (1987). Branching processes in the analysis of the height of trees. Acta Inf. **24** 277–298.

[6] L. Devroye and B. Reed. (1995) On the variance of the height of random binary search trees, SIAM J. Comput. **24**, 1157-1162.

[7] M. Drmota. (2000) The Variance of the Height of Digital Search Trees, Acta Inf., submitted.

[8] J. Jabbour-Hattab. (2001). Martingales and large deviations for the binary search trees. Random Structures and Algorithms. **19**, 2, 112-127.

[9] H. M. Mahmoud. (1992) Evolution of Random Search Trees. Wiley.

[10] P. Major. (1978) On the Invariance Principle for Sums of Independent Identically Distributed Random Variables. J. Multivariate Analysis. **8**, 487-517.

[11] U. Roesler. (1991) A limit theorem for Quicksort. Theoritical Informatics and Applications. Vol 25, n 1, 85-100

[12] U. Roesler. (1992) A fixed point theorem for distributions. Stochastic Processes and their Applications. 42, 195-214

Chapitre 7

Examens des années antérieures

7.1 Examen de Janvier 2001

Dans tout le problème, on considère un processus de Galton-Watson $(Z_n)_{n\in\mathbb{N}}$, sous-critique, donné par sa loi de reproduction $(p_k)_{k\in\mathbb{N}}$, de fonction génératrice f et de moyenne

$$m = \sum_{k\geq 0} k\ p_k < 1$$

On notera f_n la fonction génératrice de Z_n.

Partie A :

1) Rappeler la relation entre les fonctions f_n et f.

2) Posons pour tout réel $s \in [0,1]$

$$g_n(s) := \mathbb{E}(s^{Z_n} \mid Z_n > 0)$$

Exprimer g_n en fonction de f_n.

3) Posons pour tout réel $s \in [0,1]$

$$k(s) := \frac{1-f(s)}{1-s}$$

Montrer que

$$1 - g_n(s) = \frac{k(f_{n-1}(s))}{k(f_{n-1}(0))}\ (1 - g_{n-1}(s))$$

en déduire que la suite $(1-g_n(s))_{n\in\mathbb{N}}$ est croissante et montrer la convergence de g_n vers une fonction g, quand $n \to +\infty$.

4) Montrer que

$$1 - g_n(f(s)) = (1 - g_{n+1}(s))\ k(f_n(0))$$

et en déduire que $g(1) = 1$.

5) En déduire le théorème de Yaglom : pour un processus de Galton-Watson sous-critique, il existe une loi limite du processus conditionné par la non-extinction, c'est-à-dire

$$\mathbb{P}(Z_n = j \mid Z_n > 0) \overset{n\to\infty}{\longrightarrow} Q(j)$$

où Q est une loi de probabilité sur $\mathbb{N}$ non dégénérée.

Partie B :

Pour tout entier $n \geq 1$, appelons ξ_n le sommet de la première génération de l'arbre de Galton-Watson, qui est le plus à gauche parmi les sommets qui ont des descendants dans la n-ème génération. Autrement dit (avec les notations du cours)

$$\xi_n := inf\{j \in \mathbb{N} \text{ tels que } (z_n \cap T_j) \text{ non vide} \mid Z_n > 0\}$$

Appelons A_n le nombre de descendants de ξ_n à la n-ème génération.

1) Faire un dessin.

2) Comparer la loi de $A_n \mid Z_n > 0$ et la loi de $Z_{n-1} \mid Z_{n-1} > 0$.

3) Appelons μ_n la loi de $Z_n \mid Z_n > 0$.

a) Montrer que

$$\frac{\mathbb{P}(Z_n > 0)}{m^n} = \frac{1}{\int x d\mu_n(x)}$$

b) Montrer que la suite $(\frac{\mathbb{P}(Z_n>0)}{m^n})_n$ est décroissante et que

$$\lim_{n \to +\infty} \frac{\mathbb{P}(Z_n > 0)}{m^n} > 0 \iff \sup_n \mathbb{E}(Z_n \mid Z_n > 0) < \infty$$

4) On admettra le théorème d'immigration suivant : soit Z_n les tailles des générations dans un processus de Galton-Watson avec immigration Y_n. On suppose que m, moyenne de la loi de reproduction est < 1. Si $\mathbb{E}(log^+ Y) < \infty$, alors Z_n converge en loi vers Z_∞. Si $\mathbb{E}(log^+ Y) = \infty$, alors Z_n converge en probabilité vers $+\infty$.

Montrer alors l'équivalence

$$\sum_{k \geq 1} (k \ \log k) \ p_k < \infty \iff \sup_n \mathbb{E}(Z_n \mid Z_n > 0) < \infty$$

On pourra utiliser l'arbre biaisé et sa traduction en termes de processus avec immigration, puis calculer $\widehat{\mathbb{P}}(Z_n = k)$ pour tout $k \geq 1$.

7.2 Examen de Janvier 2002

Exercice 1

Soit la famille des arbres planaires généraux non vides, définie par

$$\mathcal{A} = \emptyset \oplus (\bullet, \mathcal{A}) \oplus (\bullet, \mathcal{A}, \mathcal{A}) \oplus (\bullet, \mathcal{A}, \mathcal{A}, \mathcal{A}) \oplus ...$$

On a vu en cours que la fonction génératrice dénombrant ces arbres est

$$A(z) = \sum_{n \geq 1} C_{n-1} z^n = zb(z),$$

où $b(z)$ est la fonction génératrice des arbres binaires.

1. Quel est le nombre moyen de fils de la racine dans un arbre de $\mathcal{A}$ à 4 nœuds ? à 5 nœuds ?
2. Quel est le nombre moyen de fils de la racine dans un arbre de $\mathcal{A}$ de taille n ? Donnez les valeurs exacte, et asymptotique pour $n \to +\infty$. *Attention :* il ne s'agit pas ici d'une valuation additive.
3. Quel est le nombre de feuilles d'un arbre de $\mathcal{A}$ de taille n ?
4. Quel est le nombre moyen, exact et asymptotique, de nœuds simples (à un seul fils) dans un arbre de $\mathcal{A}$ de taille n ?
5. Pouvez-vous étendre le résultat précédent pour obtenir le nombre moyen, exact et asymptotique, de nœuds d'arité p dans un arbre de $\mathcal{A}$ de taille n ? Quelle est la proportion asymptotique du nombre moyen de nœuds d'arité p ? Reconnaissez-vous une loi connue ?

Exercice 2

(les notations sont celles du polycopié)

Dans tout le problème, on considère une marche aléatoire avec branchement particulière, partant d'un ancêtre situé en 0, donnée par une loi de reproduction $(p_k)_{k\in\mathbb{N}}$, et par un processus ponctuel Z sur $\mathbb{R}$:

$$Z = \sum_{j=1}^{N} \delta_{X_j}$$

où X_j désigne la position de la j-ième particule créée et N désigne le nombre de descendants de l'ancêtre. On suppose que les déplacements des particules sont 0 ou 1 avec probabilité 1/2, autrement dit

$$\mathbb{P}(X_j = 0) = \mathbb{P}(X_j = 1) = 1/2$$

autrement dit encore, une particule créée à partir d'un point x de $\mathbb{R}$ se déplace en $x+1$ avec probabilité 1/2 et ne bouge pas avec probabilité 1/2. On supposera que la reproduction et les déplacements sont indépendants entre eux. On suppose que la reproduction vérifie

$$m := \mathbb{E}(N) > 2$$

et

$$\mathbb{E}(N^2) < \infty$$

et on note f la fonction génératrice de la reproduction :

$$f(s) = \mathbb{E}(s^N) = \sum_{k\geq 0} p_k s^k \ .$$

On appelle Z_n le cardinal de la n-ième génération ($Z_1 = N$) et on appelle $Z^{(n)}$ la mesure ponctuelle des positions des particules de la n-ième génération ($Z^{(1)} = Z$) :

$$Z^{(n)} = \sum_{|u|=n} \delta_{X_u}$$

La position de la particule u est appelée X_u et le déplacement de la particule u est appelé γ_u. Les déplacements valent 0 ou 1 et la position d'une particule $u = i_1 i_2 \dots i_n$ est

$$X_u = \gamma_{i_1} + \gamma_{i_1 i_2} + \dots \gamma_{i_1 i_2 \dots i_n}$$

L'objet du problème est de comprendre où sont les particules, quand n devient grand.

(1) Faire un dessin de cette marche aléatoire avec branchement.

(2) Y a-t-il extinction du processus ? Avec quelle probabilité ? (réponse sans preuve acceptée)

(3) **(a)** Montrer que le support de la mesure $Z^{(n)}$, c'est-à-dire $Z^{(n)}(\mathbb{R})$ est contenu dans $\mathbb{N} \cap [0, n]$.

(b) pour tout entier k, on appelle

$$\lambda(n, k) = Z^{(n)}(\{k\}) = \sum_{|u|=n} 1_{\{X_u = k\}} i$$

le nombre de particules situées en k à la n-ième génération et on note

$$G_{n,k}(s) = \mathbb{E}(s^{\lambda(n,k)})$$

la fonction génératrice de $\lambda(n, k)$.

Montrer que l'on a la relation de récurrence

$$(R) \qquad G_{n,k}(s) = f\Big(\frac{G_{n-1,k}(s) + G_{n-1,k-1}(s)}{2}\Big)$$

(4) On pose pour tout entier k,

$$p_{n,k} := \mathbb{P}(\lambda(n, k) = 0).$$

Déduire de la relation (R) que $p_{n,k}$ vérifie la relation de récurrence

$$(r) \qquad p_{n,k} = f\Big(\frac{p_{n-1,k} + p_{n-1,k-1}}{2}\Big)$$

(5) On pose pour tout entier k,

$$q_{n,k} := \mathbb{P}(\forall j \le k,\ \lambda(n, j) = 0)$$

et

$$L_n := \inf\{k \in \mathbb{N} \text{ tels que } \lambda(n, k) > 0\} = \inf \operatorname{support}(Z^{(n)})$$

(a) Montrer que $q_{n,k}$ vérifie la même relation de récurrence (r) (on pourra faire un raisonnement analogue à celui fait à la question (3)(b)).

(b) Quel est le sens de variation de $q_{n,k}$ par rapport à n ? par rapport à k ?

(c) Reporter L_n sur votre dessin de la question (1). Montrer que L_n est croissant en n et montrer que

$$\mathbb{P}(L_n > k) = q_{n,k}$$

(d) Montrer que L_n converge (dans un sens que l'on précisera) vers une limite L telle que

$$\mathbb{P}(L > k) = q_k$$

où q_k est solution de l'équation

$$q_k = f(\frac{q_k + q_{k-1}}{2})$$

(6)

On rappelle la terminologie du poly : pour tout $\theta > 0$,

$$m(\theta) = \mathbb{E}(\sum_{j=1}^{N} e^{-\theta X_j})$$

(a) Calculer $m(\theta)$ dans le cas de la marche aléatoire traitée ici.

(b) Montrer que pour tout $\theta \geq 0$,

$$\log m(\theta) - \theta \frac{m'(\theta)}{m(\theta)} > 0$$

Dessiner $m(\theta)$ et $\log m(\theta)$ pour $\theta \geq 0$. Que dit alors le théorème de Biggins ?

(c) Ecrire

$$W_n(\theta) := \sum_{|u|=n} \frac{e^{-\theta X_u}}{m(\theta)^n}$$

en fonction de m et des $\lambda(n, k)$.

(d) On définit pour tout entier k,

$$\mu(n, k) := \left(\frac{m}{2}\right)^{-n} \sum_{j=0}^{k} \binom{-n}{j} \lambda(n, k-j)$$

Montrer que pour tout $\theta \geq 0$,

$$\sum_{k \geq 0} \mu(n, k) e^{-\theta k} = W_n(\theta)$$

En déduire que pour tout entier k fixé, la suite $(\mu(n, k))_n$ est une $\mathcal{F}_n$-martingale. On appellera μ_k sa limite presque sûre quand n tend vers $+\infty$. Que peut-on dire de μ_k ?

(e) Appliquer (d) à $k = L$ pour obtenir

$$\lambda(n, L) \sim \text{Constante}\left(\frac{m}{2}\right)^n$$

quand $n \to +\infty$.

Que signifie ce résultat ?

7.3 Examen de Mai 2002

Exercice 1

Il est bien connu que les expressions arithmétiques peuvent être représentées sous forme arborescente. Nous nous intéresserons ici à la classe des expressions formées à l'aide de constantes ou variables, d'opérateurs unaires tels que *exp* ou *log*, et d'opérateurs binaires tels que $+$ ou $*$. Les arbres correspondants sont dits *unaires-binaires* : les nœuds externes, ou feuilles, contiennent les variables ou constantes, certains nœuds internes (ceux correspondant aux opérateurs unaires) ont un fils, les autres ont deux fils.

Le but de l'exercice est de dénombrer les formes d'arbres, ou arbres non étiquetés, possibles. Soit u_n le nombre d'arbres unaires-binaires à n sommets.

1. Dessinez les arbres à n sommets, pour $n \leq 3$. Calculez u_0, u_1, u_2, u_3.
2. Donnez une équation de récurrence satisfaite par les u_n.
3. On pose $U(z) := \sum_{n\geq 0} u_n z^n$. Donnez l'équation satisfaite par cette fonction génératrice.
4. Déduisez de cette équation une expression de $U(z)$ en fonction de z.
5. Comment pourriez-vous calculer de deux manières différentes u_{10} ?
6. En étudiant la valuation *nombre de fils d'un sommet interne,* donnez le nombre moyen de fils d'un sommet interne dans un arbre unaire-binaire.

Exercice 2

Soit (Z_n) un processus de Galton-Watson de moyenne $m > 1$ et de loi de reproduction $(p_k)_{k\in\mathbb{N}}$. On appelle N le nombre d'enfants de l'ancêtre, de sorte que

$$\mathbb{P}(N = k) = p_k$$

$$m = \sum_{k=0}^{+\infty} kp_k \ .$$

On suppose dans cet exercice que

$$Var(N) := \mathbb{E}(N^2) - m^2 < \infty \ .$$

On appelle $W_n := \frac{Z_n}{m^n}$ la martingale associée. On sait déjà que W_n converge presque sûrement vers une limite W. Le but de l'exercice est de montrer la convergence de W_n dans L^2.

1) Calculer l'espérance conditionnelle $\mathbb{E}(Z_n^2/\mathcal{F}_{n-1})$. En déduire une relation de récurrence sur $\mathbb{E}(Z_n^2)$. Calculer $\mathbb{E}(Z_n^2)$ en fonction de m et $Var(N)$.

2) Montrer que $\mathbb{E}(W_n^2)$ est borné. En déduire que la martingale W_n converge dans L^2 vers W.

3) Ecrire le théorème ainsi montré. Comment reliez-vous ce résultat au théorème de Kesten-Stigum ?

7.4 Examen de Janvier 2003

Exercice I

On s'intéresse dans ce problème au nombre de manières de construire une expression bien parenthésée, obtenue à partir des règles suivantes :

1. x est une expression bien parenthésée.
2. Si σ_1, σ_2, ... , σ_k sont des expressions bien parenthésées, avec $k \geq 2$, alors $(\sigma_1)(\sigma_2)$... (σ_k) est une expression bien parenthésée.
3. Les seules expressions bien parenthésées sont obtenues par les deux règles ci-dessus.

Par exemple, $(x)(x)$, $((x)(x)(x))(x)$ et $((x)(x))(x)(((x)(x))(x))$ sont des expressions bien parenthésées, mais non (x) ou $((x)(x))$.

On définira la taille d'une expression bien parenthésée comme le nombre d'occurrences de la variable x qu'elle contient.

1. Donnez toutes les expresssions bien parenthésées construites suivant les règles ci-dessus, pour $n \leq 4$.
2. Proposez une représentation arborescente des expressions bien parenthésées, et donnez les arbres correspondant aux trois expressions données en exemple.
3. Soit S_n le nombre d'expressions bien parenthésées de taille n, et soit $S(z) := \sum_n S_n z^n$ la série génératrice associée. Montrez que la série $S(z)$ satisfait une équation algébrique de degré 2, que vous résoudrez.
4. En récrivant l'équation définissant $S(z)$ sous la forme $S(z) = z\Phi(S(z))$, pour une fonction Φ convenable, puis en utilisant la formule de Lagrange, donnez une expression des S_n sous forme de somme simple.
5. Quel est le comportement asymptotique des S_n lorsque $n \to +\infty$?
6. Montrez que $S(z)$ vérifie l'équation différentielle linéaire

$$(1 - 6z + z^2)\, S^{'}(z) + (3 - z)\, S(z) = 1 - z.$$

Vous pourrez par exemple différencier l'expression exacte de $S(z)$ obtenue à la question 3. Tirez de l'équation différentielle une relation de récurrence linéaire entre S_n, S_{n+1} et S_{n+2}, dont vous vous servirez pour calculer les S_n pour $n \leq 8$.

Exercice II

Dans tout le problème, on considère un processus de Galton-Watson $(Z_n)_{n\in\mathbb{N}}$, critique, donné par sa loi de reproduction $(p_k)_{k\in\mathbb{N}}$, de fonction génératrice f et de moyenne

$$m = \sum_{k\geq 0} k\ p_k = 1$$

On notera f_n la fonction génératrice de Z_n.

1. (a) Rappeler la relation entre les fonctions f_n et f. Dessiner la fonction f.
   (b) Quelle est, pour tout t fixé, $t \in [0, 1]$, la limite quand $n \to +\infty$ de $f_n(t)$? En déduire la limite quand $n \to +\infty$ de $\mathbb{P}(Z_n > 0)$.
   (c) Montrer que $\mathbb{E}(Z_n/Z_n > 0) = \frac{1}{\mathbb{P}(Z_n>0)}$.

2. Dans cette question, on traite un cas particulier de processus de Galton-Watson. Pour $p \in]0,1[$ fixé, on pose $p_0 = p$ et pour $k \geq 1$, $p_k = (1-p)^2 p^{k-1}$.
   (a) Calculer la moyenne m de cette loi et la variance que l'on notera σ^2.
   (b) Montrer que la fonction génératrice est
   $$f(s) = \frac{p-(2p-1)s}{1-ps}$$
   et que
   $$f_n(s) = \frac{np-(np+p-1)s}{1-p+np-nps}$$
   (c) En déduire l'égalité suivante
   $$\frac{1}{n}\Big[\frac{1}{1-f_n(t)} - \frac{1}{1-t}\Big] = \frac{\sigma^2}{2}\ . \tag{7.1}$$
   On admettra dans la suite que dans le cas général d'un Galton-Watson critique de variance finie σ^2, on a toujours
   $$\lim_{n\to\infty} \frac{1}{n}\Big[\frac{1}{1-f_n(t)} - \frac{1}{1-t}\Big] = \frac{\sigma^2}{2}\ . \tag{7.2}$$
3. (a) Soit X une variable aléatoire réelle de loi exponentielle de paramètre λ, c'est-à-dire de densité $\lambda e^{-\lambda u} 1_{[0,+\infty[}(u)du$, soit encore
   $$\mathbb{P}(X > x) = e^{-\lambda x}\ .$$
   Calculer la transformée de Laplace de X : $\varphi(\alpha) := \mathbb{E}(e^{-\alpha X})$.
   (b) Montrer que
   $$\mathbb{E}(e^{-\alpha(\frac{Z_n}{n})}/Z_n > 0) = \frac{f_n(e^{-\alpha/n}) - f_n(0)}{1-f_n(0)}.$$
   (c) Déduire de l'égalité (7.2) la limite quand $n \to +\infty$ de $\mathbb{E}(e^{-\alpha(\frac{Z_n}{n})}/Z_n > 0)$.
   (d) Que peut-on dire de la loi conditionnelle limite de Z_n/n trouvée à la question précédente par sa transformée de Laplace ?
   (e) Trouver de deux manières différentes un équivalent de $\mathbb{E}(Z_n/Z_n > 0)$.

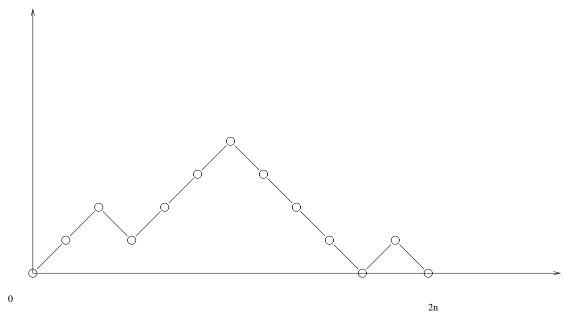
7.5 Examen de Mai 2003

On s'intéresse au dénombrement des possibilités lors d'une élection à deux candidats A et B, lorsque les votes (qui ont lieu séquentiellement) sont comptabilisés dès qu'ils sont formulés ; tous les résultats intermédiaires sont donc disponibles.

1. On suppose tout d'abord qu'il y a un nombre pair $2n$ de votants, et que les votes sont également répartis sur les deux candidats (il n'y a pas de vainqueur). Soit d_n le nombre d'élections possibles, tels que A aie toujours autant ou plus de voix que B. Calculez d_n pour $n \leq 4$.

2. Soit $d(z)$ la fonction génératrice des d_n : $d(z) = \sum_{n\geq 0} d_n z^n$. Donnez une formule explicite pour $d(z)$. Pour cela, vous pourrez chercher une équation de récurrence (sur d_n ou $d(z)$) en considérant le premier instant, lors de l'élection et après le premier vote, où les candidats A et B se retrouvent à égalité. Déduisez-en la valeur de d_n pour n quelconque, et son comportement asymptotique.
3. Avec quels objets informatiques bien connus les élections énumérées dans la question précédente peuvent-elles être mises en bijection ? Pourquoi ?
4. On s'intéresse maintenant aux élections dans lesquelles le candidat A a constamment au moins autant de voix que le candidat B, mais ne gagne que par une voix. Il y aura $n+1$ voix pour A et n voix pour B, soit $2n+1$ votants en tout. Soit e_n le nombre d'élections sur $2n+1$ votants ; donnez la valeur de e_n pour $n \leq 4$.
5. Soit $e(z) = \sum_{n\geq 0} e_n z^n$; calculez $e(z)$. Vous pourrez, comme pour le calcul de $d(z)$, obtenir d'abord une équation de récurrence, ici en regardant la dernière (ou la première) fois où les deux candidats sont à égalité. Déduisez-en la valeur de e_n pour n quelconque.
6. On construit maintenant un arbre à partir d'une suite de votes, i.e. d'un mot $w \in \{A, B\}^*$, comme suit (on ne considère que les élections dans lesquelles A est constamment en tête) :
   - on part d'une racine, donnée ;
   - on va lire le mot de gauche à droite, et construire l'arbre en parallèle ;
   - lorsqu'on lit A, on crée un fils du noeud courant et on va en ce fils (lorsqu'un noeud a plusieurs fils, on considere que ceux-ci sont crées de gauche à droite) ;
   - lorsqu'on lit B, on remonte du noeud courant vers son père ;
   - on terminera donc la lecture à un niveau en dessous de la racine.

   Par exemple, le mot $AABAABAABBABBBABAAABBAB$ donne l'arbre

   Quel est l'arbre associé au mot $AABAABABABBABBAAAABABBABAABBABB$?
7. Quels types d'arbres obtenez-vous ? Est-ce bien une bijection entre les mots considérés et cette classe d'arbres ? Ceci vous permet-il de retrouver la valeur de e_n obtenue à la question précédente ?

7.6 Examen de Janvier 2004

Exercice 1.

Le but de cet exercice est de comparer les complexités respectives de quelques algorithmes de tri, afin de déterminer les champs d'utilisation de chacun. Ces algorithme de tri sont exécutés sur un tableau de n clés, qu'on supposera distinctes : le tableau est mis en bijection avec une permutation aléatoire de $\{1...n\}$.

1. Rappeler les coûts des algorithmes de tri à bulles et de tri par sélection, comptés en nombre de comparaisons de clés, puis en nombre de transferts de clés (rappel : un échange de clés vaut trois transferts).
2. On considère maintenant l'algorithme de tri par insertion. Quel est le nombre moyen de comparaisons entre clés fait par cet algorithme ?
3. Pour ce même algorithme, quel est le nombre moyen de transferts ?
4. Supposons que, dans une situation donnée, le coût d'un transfert de clé soit égal à 100 fois le coût de comparaisons de deux clés, et que les coûts des autres opérations soient négligeables par rapport à ces deux coûts. Comment pouvez-vous utiliser cette information pour choisir un algorithme parmi les trois tris proposés ?
5. Si au contraire le coût d'une comparaison entre clés vaut 100 fois le coût d'un transfert, quel serait votre choix ?

Exercice 2.

On s'intéresse ici aux arbres unaires-binaires, i.e. aux arbres dont les nœuds sont, soit une feuille, soit un nœud double, soit un nœud simple d'arité 1 ; *à la différence des arbres binaires on ne parle pas de sous-arbre gauche ou droit dans ce cas ;* en d'autres termes il y a un seul type de nœud d'arité 1, et non deux.

1. Quelle est l'équation satisfaite par la série génératrice de dénombrement de ces arbres ? Tirez-en une expression pour le nombre u_n d'arbres unaires-binaires à n nœuds.
2. Quel est le comportement asymptotique de u_n ?

Table des matières