HAL
open science

De l'usage des opérateurs en combinatoire : construction, analyse et génération aléatoire

Nicolas Rolin

▶ **To cite this version:**

Nicolas Rolin. De l'usage des opérateurs en combinatoire : construction, analyse et génération aléatoire. Analyse numérique [cs.NA]. Université Sorbonne Paris Cité, 2016. Français. NNT : 2016USPCD022 . tel-01799559

HAL Id: tel-01799559

https://theses.hal.science/tel-01799559

Submitted on 24 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PARIS 13

Discipline : **Informatique**

École doctorale Galilée
Laboratoire d'informatique Paris Nord

Présentée et soutenue publiquement par

Nicolas ROLIN

le 6 octobre 2016

De l'usage des opérateurs en combinatoire : construction, analyse et génération aléatoire.

JURY

Olivier BODINI	Directeur de thèse
Antoine GENITRINI	Co-encadrant de thèse
Julien CLEMENT	Rapporteur
Alfredo VIOLA	Rapporteur
Frédérique BASSINO	Examinatrice
Cyril NICAUD	Examinateur
Vlady RAVELOMANANA	Examinateur

Table des matières

Introduction **ix**

Préliminaires **xiii**
0.1 Combinatoire énumérative . . . xiii
0.2 Séries génératrices . . . xiv
0.2.1 Constructions admissibles . . . xiv
0.2.2 Méthode symbolique . . . xiv
0.2.3 Séries génératrices multivariées . . . xvi
0.2.4 Séries génératrices exponentielles . . . xvi
0.2.5 Théorèmes de transfert et combinatoire analytique . . . xvii
0.3 Génération aléatoire . . . xvii
0.3.1 Génération récursive . . . xvii
0.3.2 Génération de Boltzmann . . . xviii
0.4 Convolution de Dirichlet . . . xix
0.4.1 Définitions et propriétés classiques . . . xix
0.4.2 Séries formelles et convolution de Dirichlet . . . xix
0.4.3 Extension de la convolution de Dirichlet . . . xx

I Méthode symbolique **1**

1 Arbres de processus binaires **3**
1.1 Contexte . . . 3
1.1.1 Arbres syntaxiques . . . 4
1.1.2 Arbres sémantiques . . . 5
1.2 Processus binaire typique . . . 5
1.2.1 Nombre typique d'exécutions . . . 6
1.2.2 Taille typique des arbres sémantiques . . . 8
1.3 Processus binaire typique commutatif . . . 11
1.3.1 Arbres de processus non-planaires . . . 11
1.3.2 Nombre typique d'exécutions et taille de processus non-planaires . . . 11
1.3.3 Taille typique des arbres de calculs . . . 12
1.4 Conclusion . . . 12

2 Arbres de processus avec synchronisation **13**
2.1 Approche de Stanley enrichie . . . 13
2.2 Produit partiellement ordonné . . . 15
2.3 Spécification des arbres synchronisés . . . 17
2.4 Conclusion . . . 20

3 Constructions décomposables en réplications **21**
3.1 Introduction . . . 21
3.2 Constructions . . . 21
3.2.1 Définitions . . . 22
3.2.2 Constructions de base . . . 22
3.3 Opérations sur les constructions . . . 23

3.3.1 Opérations basiques sur les constructions . . . 23
3.3.2 Opérations avancées sur les constructions . . . 25
3.4 Constructions classiques . . . 25
3.4.1 Séquence . . . 25
3.4.2 Puissance . . . 25
3.4.3 Factorisation . . . 26
3.4.4 Inverse de séquence compositionnelle de réplication . . . 26
3.4.5 Cycle . . . 26
3.4.6 Constructions ensemblistes . . . 27
3.4.7 Constructions miroir . . . 28
3.5 Séries indicatrices de réplication . . . 29
3.5.1 Constructions décomposables en réplications . . . 29
3.5.2 Constructions de base . . . 30
3.6 Séries indicatrices de réplication sur les opérations . . . 30
3.6.1 Opérations basiques sur les constructions . . . 31
3.6.2 Opérations avancées sur les constructions . . . 33
3.7 Séries indicatrices de réplication de constructions classiques . . . 34
3.7.1 Séquence . . . 34
3.7.2 Séquence compositionnelle inverse de réplications . . . 34
3.7.3 Cycles . . . 34
3.7.4 Constructions ensemblistes . . . 35
3.7.5 Constructions miroir . . . 36
3.8 Conclusion . . . 36

II Génération aléatoire 39

4 Génération récursive de structures étiquetées avec contraintes 41
4.1 Génération d'objets étiquetés : silhouettes et étiquettes . . . 41
4.1.1 Choix de taille multivarié . . . 42
4.2 Génération aléatoire de silhouettes . . . 43
4.2.1 Génération de constructeurs admissibles . . . 44
4.3 Génération d'étiquetages . . . 45
4.3.1 Étiquetages croissants . . . 45
4.3.2 Étiquetages croissants partiels . . . 45
4.4 Exemples de génération : arbres étiquetés avec contraintes d'ordre . . . 45
4.4.1 Génération aléatoire pondérée de silhouettes . . . 46
4.4.2 Génération aléatoire uniforme d'étiquetages . . . 48
4.5 Conclusion . . . 50

5 Analyse quantitative de différentes versions de la génération de Boltzmann 51
5.1 Contexte des générateurs de Boltzmann . . . 51
5.1.1 Générateur de Boltzmann . . . 51
5.2 Complexité des générateurs de Boltzmann . . . 53
5.3 Objets à distribution plate . . . 56
5.3.1 Générateur de Boltzmann classique sans rejet anticipé . . . 56
5.3.2 Générateur de Boltzmann classique avec rejet anticipé . . . 58
5.4 Objets avec distribution piquée . . . 59
5.4.1 Générateur de Boltzmann pointé . . . 59
5.4.2 Générateur de Boltzmann singulier . . . 59
5.5 Comparaison des différents méthodes pour les distributions piquées . . . 60
5.5.1 Une classe d'objet générale avec $0 < -\alpha < 1$ . . . 60
5.5.2 Grammaires apériodiques et fortement connectées : $-\alpha = 1/2$ . . . 61
5.5.3 Comparaison expérimentale des différents générateurs . . . 62
5.6 Conclusion . . . 63

6 Génération de Boltzmann mortelle **65**
6.1 Générateurs analytiques . . . 66
6.1.1 Définitions principales . . . 66
6.1.2 Constructions . . . 68
6.1.3 Illustration du taux de “mortalité” avec les arbres de Cayley . . . 69
6.2 Arbres simplement générés . . . 71
6.2.1 Approches existantes . . . 71
6.2.2 Générateurs analytiques : Maximiser un polynôme . . . 71
6.3 Constructions additionnelles . . . 72
6.3.1 Construction de réplications . . . 72
6.3.2 Construction des paires non ordonnées . . . 72
6.3.3 Arbres d’Otter . . . 73
6.3.4 L’opérateur de pointage (différentiation) . . . 74
6.4 Conclusion . . . 75

7 Conclusion et perspectives **79**

Bibliographie **81**

A Pavages 1 et 2 **85**
A.1 Settings, definitions . . . 85
A.1.1 Bivariate generating function . . . 85
A.1.2 Entropy . . . 86
A.1.3 Combinatorical results . . . 87
A.2 Automaton representation . . . 88
A.2.1 Essential, non-essential and additional states . . . 89
A.2.2 Simplified automata . . . 91
A.3 Matrix representation . . . 93
A.4 Cylindrical case . . . 93
A.5 Conclusion . . . 94

B Algorithme évolutionaire 1+1 **97**

Remerciements

Je remercie de manière générale toutes les personnes qui ont pu m'accompagner pendant trois ans, et je m'excuse par avance pour ceux que je vais oublier dans l'énumération suivante.

Je remercie donc :

— Antoine et Olivier, pour leur accompagnement synergique de ma thèse. Vous m'avez recadré quand j'en ai eu besoin, mis la pression quand c'était nécessaire, amené sur plusieurs continents (et Guéret).

— Afredo et Julien pour avoir accepté de rapporter mon manuscrit, et d'avoir donné de nombreuses remarques qui ont amélioré le résultat final.

— Cyril, Frédérique, Vlady, Alfredo et Julien d'avoir accepté de se déplacer pour être dans mon jury.

— tout mes co-auteurs : Antoine, Oliver, Jérémie, Alexandra, Frédéric, Emmanuel, Hsien-Kuei, Alois, Tsung-Hsi et Wei-Mei.

— tout les membres de mon bureau (à trois places), qui ont pu supporter la surcharge pendant ces trois ans : Pierre, Quentin, Cyril, Mario et Axel

— tout le reste de l'équipe pour les nombreuses discussions : Lionel, Julien (David & Courtiel), Andréa, Alexandra, Thierry, Axel ,Thomas, Valentin, Alice et les autres.

— ceux qui ont partagé mes TD et les autres personnalités du labo : Catherine, Mohamed, Christine, Sophie pour son sourire, Camille pour son soutien informatique, Laure pour avoir rattrapé mes gaffes, et j'en oublie.

— tout le personel administratif : Brigitte, Nathalie, Séverine,...

— tout le groupe aléa et les chercheurs associés : Anne avec qui il est agréable de se perdre dans Bath, Vincent et ses blagues mathématiques qui trompent énormément, Michael the regular of international conferences, Jérôme, Élie, Matthieu, et tout ceux dont j'ai oublié le prénom (j'ai une mauvaise mémoire).

Mais j'ai aussi une vie en dehors de la recherche, je remercie donc mes collocs : Ara, Émile, Poussinet, Martin, Ping, Marion, Ariane, Émeline, Nowleen et Pat. Je remercie aussi Maud et Daniel qui ont bien voulu venir goûter ma cuisine régulièrement, ce qui comporte des risques j'en suis bien conscient. Je remercie enfin tout ceux que j'ai pu croiser sur le campus de l'ENS Cachan : Harry, Jonas, Cpa, David, Alban, NeK, Jérémie, Marc, Medthieux, Gaétan, Adèle, ...

Je tiens enfin en dernier lieu à remercier mes parents et à ma famille pour leur soutien pendant toute cette aventure.

Introduction

Le sujet de cette thèse est la combinatoire, on s'intéresse donc à des objets intrinsèquement finis, définis par des règles d'assemblage simples. Le principal objet d'étude est celui des *classes* combinatoires, c'est à dire des ensembles d'objets sur lesquels on peut définir une notion de taille, et comportant un nombre fini d'objets d'une taille donnée. Ce modèle suffit à décrire exactement les structures provenant des domaines informatiques, où la mémoire prise par une structure est une taille qui ne peut physiquement correspondre qu'à un nombre fini d'éléments. La combinatoire s'intéresse à l'étude extensive d'une classe combinatoire donnée. Dans ce cadre, le premier problème rencontré, l'*énumération*, consiste à trouver une spécification pour une classe donnée. Cela veut dire trouver une bijection (souvent récursive) qui permette de construire tous les éléments de la classe de manière non ambiguë. Une fois la spécification trouvée, un autre problème consiste à trouver la *série génératrice* associée à la classe combinatoire. La série génératrice d'une classe nous renseigne sur la distribution de ses objets selon leur taille et plus exactement le nombre d'objets ayant une taille donnée, et des techniques d'analyse permettent d'obtenir automatiquement des équivalents asymptotiques de ces quantités pour nombre de classes combinatoires. Pour faire une étude fine de ces objets on peut adjoindre un poids à chaque objet (la hauteur ou la largeur dans le cas des arbres, si la taille correspond au nombre de nœuds), et utiliser les méthodes de combinatoire analytique pour en déduire par exemple le poids typique d'un objet. Le dernier centre d'intérêt est la *génération aléatoire uniforme* d'objets dans une classe combinatoire à taille donnée, où chaque objet a une probabilité $1/T_n$ d'être tiré, où T_n est le nombre d'objets de la taille choisie. Bien que demander l'uniformité puisse sembler arbitraire, il est possible d'obtenir des objets tirés selon la plupart des distributions en adaptant des techniques de génération uniforme, comme par exemple en générant des objets uniformément dans un espace modifié. Un relâchement de ce paradigme consiste à tirer un objet non plus uniformément pour une taille donnée, mais de tirer un objet selon une distribution qui conserve l'uniformité pour une taille donnée, mais n'impose pas de taille à priori.

Dans cette thèse, on va particulièrement s'intéresser à l'énumération et à la génération, avec un accent sur opérations qui transforment une classe en une autre, les *constructions*.

Résumé de la thèse :

Chapitre 1 : Arbres de processus binaires

Le premier chapitre de cette thèse s'appuie sur l'article [7], écrit en collaboration avec Olivier Bodini, Antoine Genitrini et Frédéric Peschanski, et accepté à CALDAM.

On se fonde dans ce chapitre sur un modèle de concurrence qui peut se traduire en terme d'arbres étiquetés, avec des contraintes de croissance. Dans l'article [6], les auteurs considèrent des arbres planaires d'arité quelconque, et on l'étend à des arbres binaires non-planaires, qui sont plus proches des modèles issus de la théorie de la concurrence. On étudie d'abord un modèle intermédiaire, avec des nœuds binaires mais planaires, dont l'analyse se révèle assez difficile. Le passage aux arbres non-planaires devient par contre assez direct.

Chapitre 2 : Arbres de processus avec synchronisation

Le deuxième chapitre de cette thèse s'appuie sur l'article [9], écrit en collaboration avec Olivier Bodini et Antoine Genitrini, et accepté à GASCOM.

Il s'agit d'une extension du modèle précédent, où les contraintes d'ordre ne suivent plus une structure arborescente. Ces objets se révèlent durs à spécifier avec les outils existants, on introduit donc un nouvel opérateur, qui nous permet d'obtenir une caractérisation succincte de nos objets, et de parvenir à obtenir la série génératrice. On montrera comment appliquer ce nouvel opérateur dans le cas des arbres de processus avec synchronisation, ou *arbres synchronisés*.

Chapitre 3 : Constructions décomposables en réplications

Le troisième chapitre n'a pas fait l'objet d'une publication.

Le chapitre vise à bâtir des constructions telles que les opérateurs de Pólya pour les objets non-étiquetés, en se basant sur des opérateurs basiques. On propose d'étendre la méthode symbolique aux *constructions*, en commençant par les opérateurs qui découlent naturellement de la méthode

symbolique sur les classes, et en ajoutant des extensions spécifiques aux constructions. La nouveauté consiste à ajouter des réplications élémentaires et l'inversion compositionnelle d'une construction aux briques de base de notre modèle. Cela permet des interprétations combinatoires plus puissantes, mais il y a un contrecoup, puisque l'inverse combinatoire d'une construction ne va pas de manière générale être une construction définie pour toute classe. Le résultat principal de ce chapitre consiste à retrouver de manière naturelle les expressions formelles des opérateurs de Pólya.

Chapitre 4 : Génération récursive de structures étiquetées avec contraintes

Le quatrième chapitre de cette thèse s'appuie sur des parties des articles [7] et [9], écrits en collaboration avec Olivier Bodini, Antoine Genitrini et Frédéric Peschanski.

Ce chapitre regroupe des techniques de génération qui permettent de générer les classes vues au chapitres 1 et 2, et plus généralement à des objets étiquetés avec contraintes d'ordres. Les techniques de génération récursive sur des objets étiquetés sont déjà bien connues [26], mais la nouveauté dans ce cas est que les étiquetages ne sont pas classiques, et nécessitent plus d'attention que dans les cas habituels. On va notamment séparer la génération d'un objet en deux parties bien distinctes et indépendantes : la génération pondérée de la silhouette, et la génération uniforme de l'étiquetage d'une silhouette.

Chapitre 5 : Analyse quantitative de différentes versions de la génération de Boltzmann

Le cinquième chapitre de cette thèse s'appuie l'article [8], écrit en collaboration avec Olivier Bodini et Antoine Genitrini, et accepté à GASCOM.

On compare dans ce chapitre l'efficacité de deux méthodes de génération de Boltzmann. Les deux méthodes consistent en deux approches différentes de traiter le cas de classes d'objets ayant des distributions désavantageuses pour la génération de Boltzmann. La première méthode, la génération de Boltzmann singulière, consiste à profiter d'être dans un cas désavantageux pour faire une approximation et utiliser la singularité de la fonction génératrice comme paramètre. La deuxième méthode, la génération de Boltzmann pointée, consiste à changer la classe à générer à l'aide d'une habile bijection, afin de se retrouver dans un cas plus simple. On conserve ainsi les pré-calculs habituels, et on doit y ajouter le coût nécessaire pour effectuer la bijection. La deuxième méthode est évidemment plus efficace que la première si on ne prend pas en compte les coûts de pré-calculs et de transformation de l'objet final. On réalise une analyse quantitative de cette différence d'efficacité, afin d'avoir une évaluation précise de ce gain.

Chapitre 6 : Génération de Boltzmann mortelle

Le sixième chapitre de cette thèse reprend l'article [10] avec Olivier Bodini et Jérémie Lumbroso, et soumis à ANALCO.

On y décrit une extension de la génération de Boltzmann, où l'on montre de manière précise comment on peut approximer les valeurs de l'Oracle en conservant l'uniformité exacte. Les approximations se payent en rajoutant au générateur une probabilité de faire mourir l'objet pendant la construction, obligeant le générateur à en construire un autre. Cette technique reste néanmoins utile, puisque le prix à payer est directement corrélé à la qualité de l'approximation, ce qui dans certains cas permet d'obtenir des probabilités de mourir négligeables, en conservant l'uniformité exacte et en s'épargnant des implémentations complexes pour la garantir.

Annexes

Ce sont des travaux sur lesquels j'ai collaboré pendant ma thèse.

Annexe A : Pavages 1 et 2

Le premier annexe est l'article [48] avec Alexandra Ugolinkova, accepté à *RAIRO - Theor. Inf. and Applic.* On réalise ici une étude combinatoire d'un problème dont l'exposé est simple, puisqu'il

s'agit de compter le nombre de pavages possible d'un rectangle de taille fixée avec des carrés de coté 1 et 2. L'analyse se révèle assez difficile dans le cas général, et on se restreint donc à des rectangles avec un coté de taille petite fixée, et l'autre coté avec une taille quelconque.

Annexe B : Algorithme évolutionnaire 1+1

Le deuxième annexe est l'article [32] avec Hsien-Kuei Hwang, Alois Panholzer, Tsung-Hsi Tsai et Wei-Mei Chen, en cours de révision pour *Evolutionary Computation Journal.* C'est une analyse extrêmement fine d'un algorithme génétique. Les algorithmes génétiques sont issus d'une méta-heuristique dont l'étude est très complexe, ainsi les premiers travaux les étudiant ont essayé de prendre les instances les plus simples, pour arriver à un algorithme qu'il est possible d'étudier. Un modèle simple classique est celui où l'on ne fait pas de crossover (l'intérêt principal de la reproduction sexuée, qui consiste à faire un mélange de deux solutions), uniquement des mutations. On restreint encore en limitant l'espace des solutions à $\{0,1\}^n$, et en se plaçant dans le cas où la fonction à optimiser est la somme de la valeur des éléments. Il se trouve que même dans ce cas excessivement simplifié, l'étude du temps que l'algorithme met à trouver la bonne solution se révèle particulièrement complexe, bien que des premiers résultats asymptotiques ait été montrés dans [27]. L'annexe montre des résultats très fins sur la complexité de cet algorithme, améliorant tous les résultats trouvés précédemment.

Préliminaires

0.1 Combinatoire énumérative

Le domaine d'intérêt de la combinatoire est l'étude des structures discrètes, en vue de pouvoir les mesurer et les construire. On dit qu'un ensemble $\mathcal{C}$ est une *classe* si l'on peut la munir d'une fonction de taille $|.| : \mathcal{C} \mapsto \mathbb{N}$, tel que le nombre d'éléments de $\mathcal{C}$ ayant une taille n fixé soit fini. Cette définition est cohérente avec l'envie d'étudier des structures discrètes, puisque qu'une classe est naturellement contrainte à être dénombrable. La définition d'une classe $\mathcal{C}$ implique qu'on peut définir la *suite énumérative* associée $(c_n)_{n\in\mathbb{N}}$. On dit qu'un objet de taille 1 est un *atome*, ainsi on décrit les classes comme une décomposition d'un objet de grande taille en atomes.

Les questions usuelles en combinatoire sont *combien ai-je d'objets dans ma classe pour une taille donnée ?* et *pour une taille donnée, combien ai-je d'objets dans ma classe qui vérifient une propriété donnée ?* La première question peut être abordée en construisant des bijections entre la classe de départ et d'autres classes qui induisent une relation entre leurs tailles.

Exemple 0.1. *Mots binaires :*

Soit la classe $\mathcal{B}$ des mots composés uniquement de 0 et de 1, avec $|.|$ la fonction de taille qui a un mot renvoie le nombre de lettres contenues dans ce mot. On peut facilement construire une bijection entre $\mathcal{B}$ et $\mathcal{E} \cup (\{0,1\} \times \mathcal{B})$, où $\mathcal{E}$ représente un ensemble avec un élément de taille 0, puisque un mot binaire est soit le mot vide, soit une lettre puis un mot binaire. On en déduit une relation de récurrence sur les b_n, car le mot vide appartient à l'ensemble, ce qui implique $b_0 = 1$, et un mot de taille $n \neq 0$ se décompose en une première lettre de taille 1, où l'on a deux choix, et une deuxième partie de taille $n-1$, pour lequel on a b_{n-1} choix. On en déduit que $b_n = 2b_{n-1}$, donc $b_n = 2^n$.

Exemple 0.2. *Arbres binaires :*

Soit la classe $\mathcal{B}$ des arbres binaires planaires, définis comme étant les objets étant soit une feuille soit un nœud avec un arbre binaire droit et un arbre binaire gauche, et la taille est égale au nombre de nœud internes. On a donc une claire bijection entre $\mathcal{B}$ et $\mathcal{E} \cup \mathcal{Z} \times (\mathcal{B} \times \mathcal{B})$, où $\mathcal{Z}$ représente un ensemble avec un élément de taille 1, ce qui cette fois se traduit en une équation de récurrence :

$$b_0 = 1 \text{ et } b_n = \sum_{0\leq k\leq n-1} b_k b_{n-k-1}.$$

La résolution de cette équation donne $c_n = \frac{1}{n+1}\binom{2n}{n} = Cat(n)$, qui sont les célèbres *nombres de Catalan*. On remarque ici que le fait que l'on distingue branche droite et branche gauche est important, car il nous permet d'obtenir une bijection simple. Le modèle où les deux fils peuvent ne sont pas ordonnés existe aussi – il s'agit des arbres d'Otter – mais est combinatoirement plus compliqué.

On remarque que dans nos bijections certaines opérations apparaissent souvent, comme le produit ou l'union de deux classes, et qu'on peut facilement décrire l'effet qu'ils ont sur la récurrence associée. Par exemple, on remarque que

$$\text{si } \mathcal{A} = \mathcal{B} \times \mathcal{C}, \text{ alors } a_n = \sum_{0\leq k\leq n} b_k c_{n-k}.$$

Il est néanmoins un peu étrange de se retrouver avec un produit aussi "compliqué" pour une opération aussi banale, et cela nous donne l'intuition qu'il existe peut être un modèle plus pertinent pour réaliser des études quantitatives de classes combinatoires.

0.2 Séries génératrices

Définition 0.1. *Deux classes combinatoires $\mathcal{A}$ et $\mathcal{B}$ sont dites* isomorphes*, que l'on notera $\mathcal{A} \cong \mathcal{B}$, si leurs suites énumératives sont égales.*

Cette définition implique donc que l'on ne va considérer pour une classe donnée uniquement sa suite énumérative, ce qui justifie la définition suivante :

Définition 0.2. *Soit une classe $\mathcal{C}$, on définit sa* fonction génératrice ordinaire *$C(z)$:*

$$C(z) \triangleq \sum_{n\in\mathbb{N}} c_n z^n = \sum_{\gamma\in\mathcal{C}} z^{|\gamma|}$$

Notations. On notera en lettre cursive $\mathcal{C}$ une classe, en lettre droite sa série génératrice $C(z)$, et en petit sa suite énumérative $(c_n)_{n\in\mathbb{N}}$. On notera par défaut [1] $\mathcal{C}_i$ la classe $\mathcal{C}$ réduit à ses éléments de taille i. On introduit la notation $[z^n]C(z) \triangleq c_n$ qui nous permet d'extraire le $n-$ième coefficient d'une fonction génératrice.

0.2.1 Constructions admissibles

On va définir la notion de construction admissible, qui nous permet de définir le cadre des constructions pouvant être étudiées par des méthodes reposant sur des séries génératrices.

Définition 0.3. *Soit Ψ une construction m-aire qui associe à la collection de classes $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \ldots, \mathcal{A}^{(m)}$ une nouvelle classe*

$$\mathcal{B} = \Psi[\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \ldots, \mathcal{A}^{(m)}].$$

Ψ est admissible *si et seulement si sa suite énumérative (B_n) de $\mathcal{B}$ dépend uniquement des suites énumératives $\left({A_n}^{(1)}\right), \left({A_n}^{(2)}\right), \ldots, \left({A_n}^{(m)}\right)$ de $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \ldots, \mathcal{A}^{(m)}$*

La notion d'admissibilité relie donc fortement une construction et les propriétés combinatoires des objets considérés, et constitue donc une condition nécessaire pour analyser combinatoirement une construction. On ne considérera par la suite que des constructions admissibles. La conséquence immédiate est que si une construction Ψ est admissible, alors il existe un opérateur ψ tel que :

$$B(z) = \psi[A^{(1)}(z), A^{(2)}(z), \ldots, A^{(m)}(z)].$$

Exemple 0.3. *Construction non admissible. $\mathcal{X}_1, \mathcal{X}_2 \mapsto \mathcal{X}_1 \bigcup \mathcal{X}_2$ n'est pas un constructeur admissible. En effet soit $\mathcal{X} = \{a\}$ une classe contenant un élément a de taille 1, et $\mathcal{Y} = \{b\}$ une classe contenant un élément b de taille 1, avec $a \neq b$. Alors $\mathcal{X}\bigcup\mathcal{X}$ existe (et en l'occurrence est égal $\mathcal{X}$), mais $card\{\mathcal{X}\bigcup\mathcal{X}\} = 1 \neq 2 = card\{\mathcal{X}\bigcup\mathcal{Y}\}$. $\bigcup$ n'est donc pas une construction admissible.*

0.2.2 Méthode symbolique

On veut étudier quel effet ont les constructions admissibles naturelles sur les séries génératrices associées, afin de pouvoir automatiquement trouver la série génératrice d'une classe en utilisant sa définition recursive utilisant des constructions.

Classes élémentaires. On considère les classes suivantes :

— La classe $\emptyset$ n'ayant aucun élément.

— La classe $\mathcal{E}$ appelée la classe neutre, avec uniquement un élément de taille 0.

— La classe $\mathcal{Z}$ appelée la classe atomique, avec uniquement un élément de taille 1.

Les fonctions génératrices $\varnothing(z)$, $E(z)$ et $Z(z)$ correspondant respectivement aux classes $\emptyset$, $\mathcal{E}$ et $\mathcal{Z}$ sont :

$$\varnothing(z) = 0 \ , \ E(z) = 1 \text{ et } Z(z) = z$$

La classe atomique va servir à représenter les parties d'un objet qui portent la taille, comme par exemple les lettres pour un mot, les noeuds pour un arbre, etc.

1. On notera que cette convention ne sera pas respectée dans le chapitre 2 et certaines parties du chapitre 4.

Union disjointe. On voit d'après l'exemple 0.3 que l'union de deux classes n'a pas de sens combinatoire, on considérera donc toujours l'union disjointe, que l'on notera avec le symbole $+$. Dans le cas où $\mathcal{A}$ et $\mathcal{B}$ ont des éléments communs, on introduit deux marqueurs distincts $\heartsuit$ et $\diamondsuit$ de taille 0, et ont définit :

$$\mathcal{A} + \mathcal{B} \triangleq (\{\heartsuit\} \times \mathcal{A}) \cup (\{\diamondsuit\} \times \mathcal{B})$$

On peut reprendre les observations de la section précédente, et en déduire que :

$$\mathcal{A} = \mathcal{B} + \mathcal{C} \Rightarrow A(z) = B(z) + C(z)$$

ce qui donne un dictionnaire qui permet de spécifier les classes combinatoires les plus simples.

Produit cartésien. La construction $\mathcal{A} = \mathcal{B} \times \mathcal{C}$ donne toute les paires ordonnées . On peut traduire cette construction en terme de série génératrice :

$$\mathcal{A} = \mathcal{B} \times \mathcal{C} \Rightarrow A(z) = B(z)C(z).$$

Exemple 0.4. *Arbres binaires :*

On reprend la classe $\mathcal{B}$ des arbres binaires planaires de l'exemple 0.2. La bijection entre $\mathcal{B}$ et $\mathcal{E} \cup \mathcal{Z} \times (\mathcal{B} \times \mathcal{B})$ se traduit en une équation sur sa série génératrice :

$$B(z) = 1 + zB(z) \cdot B(z).$$

On peut donc résoudre cette équation, ce qui donne :

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

Autres constructions. On peut rajouter quelques autres opérations telles que :

— la *séquence* $\textsc{Seq}(\mathcal{C})$ d'une classe $\mathcal{C}$, dont les éléments sont les successions d'un nombre quelconque d'éléments de $\mathcal{C}$.

— l'*ensemble* d'une classe $\textsc{PSet}(\mathcal{C})$, dont les éléments sont les sous ensembles de la classe $\mathcal{C}$.

— le *multiensemble* d'une classe $\textsc{MSet}(\mathcal{C})$, dont les éléments sont les sous ensembles de la classe $\mathcal{C}$, où l'on s'autorise la répétition d'éléments.

— le *cycle* d'une classe $\textsc{Cyc}(\mathcal{C})$, dont les éléments sont les successions d'un nombre d'éléments de $\mathcal{C}$, à symétrie circulaire près.

Les opérateurs sur les séries génératrices correspondant à ces constructions sont :

Union disjointe :	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$\Rightarrow A(z) = B(z) + C(z)$
Produit cartésien :	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$\Rightarrow A(z) = B(z) \cdot C(z)$
Séquence :	$\mathcal{A} = \textsc{Seq}(\mathcal{B})$	$\Rightarrow A(z) = \dfrac{1}{1 - B(z)}$
Ensemble :	$\mathcal{A} = \textsc{PSet}(\mathcal{B})$	$\Rightarrow A(z) = \exp\left(\sum_{k=1}^{\infty} \dfrac{(-1)^{k-1}}{k} B(z^k)\right)$
Multiensemble :	$\mathcal{A} = \textsc{MSet}(\mathcal{B})$	$\Rightarrow A(z) = \exp\left(\sum_{k=1}^{\infty} \dfrac{1}{k} B(z^k)\right)$
Cycle :	$\mathcal{A} = \textsc{Cyc}(\mathcal{B})$	$\Rightarrow A(z) = \sum_{k=1}^{\infty} \dfrac{\phi(k)}{k} \log\left(\dfrac{1}{1 - B(z^k)}\right)$

Pointage de classe. Soit $\{\epsilon_1, \epsilon_2, \ldots\}$ un ensemble infini d'objets de taille 0 distincts. Le pointage de la classe $\mathcal{B}$, noté $\mathcal{A} = \Theta\mathcal{B}$ est :

$$\Theta\mathcal{B} \triangleq \sum_{n\geq 0} \mathcal{B}_n \times \{\epsilon_1, \ldots, \epsilon_n\}$$

Cela permet de distinguer un élément atomique en particulier dans un objet, en lui ajoutant une marque. On a alors un résultat sur les fonctions génératrices :

$$\mathcal{A} = \Theta\mathcal{B} \Rightarrow A(z) = z\frac{\partial B(z)}{\partial z}$$

0.2.3 Séries génératrices multivariées

Dans le but d'obtenir des résultats plus fins, on peut munir une classe d'un (ou plusieurs) poids, qui est une fonction qui à tout objet de la classe associe un entier. On peut par exemple dans le cas d'un arbre avoir une taille qui correspond au nombre de nœud, et un poids qui reflète la hauteur de l'arbre. Cela se traduit en série génératrice. Pour une classe $\mathcal{C}$ on définit

$$C(z, u) \triangleq \sum_{n,p\in\mathbb{N}} c_{n,p} z^n u^p,$$

où $c_{n,p}$ est le nombre d'objet de $\mathcal{C}$ de taille n et de poids p.

0.2.4 Séries génératrices exponentielles

On peut décider de donner une autre définition pour nos séries génératrices, notamment pour que notre produit de séries génératrices soit le plus simple possible. On définit une classe étiquetée comme une classe où chaque élément est constitué d'une *silhouette* contenant les atomes, et où chaque atome possède une étiquette dans $\{1, \ldots, n\}$, et chaque étiquette est utilisée exactement une fois. Alors si on considère les fonctions génératrices ordinaires associés à ces objets, le produit des fonctions ne correspond plus au produit cartésien des objets étiquetés.

Dans ce cas on préfère utiliser la *fonction génératrice exponentielle* $C(z)$:

$$C(z) \triangleq \sum_{n\in\mathbb{N}} c_n \frac{z^n}{n!}.$$

Ainsi

$$B(z)C(z) = \sum_{n\in\mathbb{N}} \sum_{0\leq k\leq n} \binom{n}{k} b_k c_{n-k} z^n,$$

ce qui correspond au produit deux deux structures étiquetées, puisque l'on mélange les étiquettes des deux classes pour ré-étiqueter sur la taille totale.

La méthode symbolique marche aussi dans ce cas, avec néanmoins une modification du dictionnaire, la distinction entre ensemble et multiensemble disparaissant avec l'étiquetage. Les opérateurs sur les séries génératrices correspondant sont :

Union disjointe :	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$\Rightarrow A(z) = B(z) + C(z)$
Produit cartésien :	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$\Rightarrow A(z) = B(z) \cdot C(z)$
Séquence :	$\mathcal{A} = \textsc{Seq}(\mathcal{B})$	$\Rightarrow A(z) = \dfrac{1}{1 - B(z)}$
Ensemble :	$\mathcal{A} = \textsc{Set}(\mathcal{B})$	$\Rightarrow A(z) = \exp(B(z))$
Cycle :	$\mathcal{A} = \textsc{Cyc}(\mathcal{B})$	$\Rightarrow A(z) = \log \dfrac{1}{1 - B(z)}$

Produit de Greene. Dans le cas des objets étiquetés, on introduit aussi une construction qui permet de spécifier un ordre sur les étiquettes des différents atomes. Le produit avec contrainte d'ordre binaire $\mathcal{A}^{\square} \star \mathcal{B}$, introduit par Greene [29], permet d'introduire des contraintes d'ordre sur les structures. $\mathcal{A}^{\square} \star \mathcal{B}$ encode le produit des classes étiquetées $\mathcal{A}$ et $\mathcal{B}$ tel que la plus petite étiquette apparaisse dans la composante de $\mathcal{A}$. Cela peut être exprimé avec la méthode symbolique, puisque $\mathcal{A}^{\square} \star \mathcal{B}$ se traduit en termes de séries génératrices par

$$\int_0^z A'(t) \cdot B(t)dt.$$

0.2.5 Théorèmes de transfert et combinatoire analytique

Un des atouts de la méthode symbolique est de pouvoir compter sur des théorèmes d'analyse puissants, pour passer efficacement de l'expression d'une fonction génératrice à un équivalent asymptotique de ses coefficients. Les théorèmes sont décris en détail par Phillipe Flajolet et Andrew Odlysko [22], on va donner ici une définition essentielle et un résultat de transfert d'équivalent.

Définition 0.4. *Soit deux nombres ψ, R avec $R > 1$ et $0 < \psi < \frac{\pi}{2}$, le domaine ouvert $\Delta(\psi, R)$ est défini comme :*

$$\Delta(\psi, R) = \{z \in \mathbb{C} | |z| < R, z \neq 1, |arg(z-1)| > \psi\}.$$

Un domaine est un Δ-domaine en 1 *si il existe R et ψ tel que c'est un $\Delta(\psi, R)$. Pour un nombre complexe $\zeta \neq 0$, un* Δ-domaine en ζ *est l'image de la fonction $z \mapsto \zeta z$ d'un Δ-domaine en 1. Une fonction est* Δ-analytique *si elle est analytique dans un Δ-domaine.*

Théorème 0.1. *Soit une fonction $f(z)$ Δ-analytique telle que*

$$f(z) \sim (1-z)^{-\alpha} \text{ lorsque } z \to 1, z \in \Delta,$$

avec $\alpha \notin \{0, -1, -2, \ldots\}$. Alors les coefficients de f satisfont

$$[z^n]f(z) = \frac{n^{\alpha-1}}{\Gamma(\alpha)}.$$

Il est aussi possible de traiter les cas où $f(z) \sim (1-z)^{-\alpha} \log(1-z)^{-\beta}$, cf. [24, p388].

0.3 Génération aléatoire

Un générateur aléatoire d'une classe $\mathcal{C}$ est un algorithme qui tire aléatoirement un objet $a \in \mathcal{C}$ selon une distribution donnée. Le plus classique est la génération aléatoire uniforme à taille exacte, où l'on choisit une taille n, et l'algorithme tire uniformément des objets de la taille donnée appartenant à la classe. Ces générateurs permettent d'avoir accès à des objets de grande taille, pour par exemple établir ou vérifier des conjectures sur des mesures typiques, ou mesurer l'efficacité d'un algorithme sur de grandes instances.

0.3.1 Génération récursive

La méthode la plus naturelle est la méthode récursive [26] [43], qui consiste à décomposer un objet en objets de plus petite taille, décider d'une répartition de taille de ces objets, de les tirer aléatoirement, puis de les recombiner

Elle est automatique dans le cas des classes générées spécifiables avec la méthode symbolique. L'exemple illustrant le mieux la méthode est celui du produit : si l'on veut générer un objet de taille n d'une classe $\mathcal{A}$ qui vérifie $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, on doit considérer toutes les tailles possibles k pour l'objet provenant de $\mathcal{B}$. La probabilité que l'objet de $\mathcal{B}$ soit de taille k est

$$\mathbb{P}[X = k] = \frac{b_k c_{n-k}}{\sum_{p=0}^{n} b_p c_{n-p}}.$$

On doit donc calculer et stocker tous les coefficients a_k et b_k, jusqu'à la taille voulue.

0.3.2 Génération de Boltzmann

L'idée principale de la génération de Boltzmann, introduite par Duchon *et al.* [20], est que l'on va vouloir générer un objet selon une distribution de taille qui conserve l'uniformité à taille fixée, au lieu de générer juste une taille fixée. On garde l'objet généré si la taille nous convient (généralement un intervalle autour d'une taille donnée), on le jette et en retire un autre sinon. Le coup de force de cette technique de génération est que le rejet s'avère assez efficace et simple pour de nombreuses applications. On choisit pour la génération de Boltzmann la distribution de Boltzmann, qui est la distribution la plus simple et naturelle pour une classe, puisqu'il s'agit de la distribution de taille que l'on obtient en normalisant par la série génératrice de la classe.

Définition 0.5. *Soit $\mathcal{C}$ une classe combinatoire non étiquetée, de fonction génératrice $C(z)$. Son modèle de Boltzmann associé, dépendant d'un paramètre réel $0 < z < \rho$ ou $z = \rho$ dans certains cas, est la distribution sur les objets de $\mathcal{C}$, telle que la probabilité d'un objet γ est*

$$\mathbb{P}_z(\gamma) = \frac{z^{|\gamma|}}{C(z)},$$

On remarque que la distribution est uniforme sur tous les objets de même taille. Ainsi la probabilité d'avoir un objet de taille n, notée $\mathbb{P}_z(N = n)$ et l'espérance de la taille de l'objet, notée $\mathbb{E}_z(N)$, pour la paramètre z, vérifient :

$$\mathbb{P}_z(N = n) = \frac{C_n z^n}{C(z)}, \qquad \text{et} \qquad \mathbb{E}_z(N) = \frac{z\ C'(z)}{C(z)}.$$

Dans le cas des arbres binaires, on obtient donc :

$$\mathbb{P}_z(N = n) = \binom{2n}{n} \frac{2z^{n+1}}{(n+1)(1-\sqrt{1-4z})} \qquad \text{et} \qquad \mathbb{E}_z(N) = \frac{1-2z-\sqrt{1-4z}}{(1-\sqrt{1-4z})\sqrt{1-4z}}.$$

Un *générateur de Boltzmann* est un constructeur qui génère des objets selon leur distribution de Boltzmann. Puisque la taille de l'objet généré est aléatoire, pour tirer un objet de taille n, les générations sont répétées jusqu'à obtenir un objet de la bonne taille.

On remarque que la méthode de génération se voit affublé d'un paramètre de contrôle z qui va changer la distribution de taille qui sera adoptée.

Avec un générateur de Boltzmann pour une classe combinatoire non étiquetée $\mathcal{C}$, pour laquelle il y a c_n éléments de taille n, la probabilité de tirer un objet $\gamma \in \mathcal{C}$ est :

$$\mathbb{P}_z[\gamma] = \frac{z^{|\gamma|}}{C(z)} \qquad \text{avec} \qquad C(z) := \sum_{n=1}^{\infty} c_n z^n = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|} \tag{1}$$

Où $|\gamma|$ est la taille de l'objet γ et x est un paramètre de contrôle à choisir. Donc la probabilité de tirer un objet de taille n est

$$\mathbb{P}_z[|\gamma| = n] = \frac{c_n z^n}{C(z)} \qquad \mathbb{P}_z[\gamma \mid |\gamma| = n] = \frac{1}{c_n},$$

tandis que la probabilité de tirer un objet sachant sa taille donnée est uniforme.

Le nom de la méthode évoque le modèle de Boltzmann de physique statistique, qui assigne à chaque état possible d'un système une probabilité $\mathrm{e}^{-\beta E}/Z$, où E est l'énergie de l'état, $\beta = 1/T$ est une constante, et Z est une constante normalisante.

Pour obtenir une génération [20] plus efficace, on s'accorde généralement une marge d'erreur sur la taille des objets. Par exemple, on n'acceptera que les objets dont la taille est dans les bornes $n(1-\varepsilon)$ et $n(1+\varepsilon)$, où ε est la marge d'erreur.

Les opérateurs de la méthode symbolique se traduisent directement en opérateur sur les générateurs de Boltzmann, rendant quasiment automatique la génération dans des classes spécifiables avec la méthode symbolique.

0.4 Convolution de Dirichlet

Ce chapitre donne des bases sur les convolutions de Dirichlet, qui seront nécessaires pour le chapitre 3.

0.4.1 Définitions et propriétés classiques

Definition 1. *La convolution de Dirichlet de deux fonctions arithmétiques (de $\mathbb{N}$ dans $\mathbb{N}$) f et g est la fonction arithmétique $f \star g$ définie par :*

$$\forall n \in \mathbb{N}^*, (f \star g)(n) = \sum_{d|n} f(d)g(n/d)$$

où $d|n$ signifie que d divise n.

L'ensemble des fonctions arithmétiques, muni de l'addition et de la convolution de Dirichlet, forme un anneau commutatif, ce qui veut dire que $\star$ est associatif, commutatif, et distributif par rapport à l'addition.

On dispose de plus de propriétés additionnelles lorsque la fonction f a une image de 1 non nulle, puisqu'elle possède alors un inverse, et qu'il est calculable :

$$f^{\{-1\}}(n) = \begin{cases} \frac{1}{f(1)} & \text{quand } n = 1 \\ -\sum\limits_{k|n, k\neq 1} \frac{f^{\{-1\}}(k)f(n/k)}{f(1)} & \text{sinon} \end{cases}$$

On va noter $\delta(n)$ la fonction qui vérifie :

$$\delta(n) = \begin{cases} 1 & \text{quand } n = 1 \\ 0 & \text{sinon} \end{cases}$$

Nombre de fonctions arithmétiques s'expriment avec les fonctions δ, 1 (fonction constante égale à 1), id (fonction qui a un nombre renvoie lui-même), des convolutions de Dirichlet et des inversions de Dirichlet. On définit la fonction de Möbius μ comme l'inverse de Dirichlet de 1, c'est à dire que $\mu \star 1 = 1 \star \mu = \delta$. La fonction φ d'Euler peut être définie comme l'unique fonction arithmétique vérifiant $\varphi \star 1 = id$, ou alors de manière équivalente $\phi = id \star \mu$. On dispose aussi de définitions explicites pour ces fonctions, puisque :

$$\mu(n) = \begin{cases} 1 & \text{quand } n \text{ est un entier sans facteur carré et un nombre pair de facteurs premiers.} \\ -1 & \text{quand } n \text{ est un entier sans facteur carré et un nombre impair de facteurs premiers.} \\ 0 & \text{quand } n \text{ a un facteur premier qui est un carré.} \end{cases}$$

et

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

0.4.2 Séries formelles et convolution de Dirichlet

Un des intérêts de la convolution de Dirichlet est qu'elle permet de transcrire au niveau des coefficients l'effet de la multiplication de deux séries.

Soit V un espace vectoriel de base $(v_n)_{n\in\mathbb{N}}$, et le produit tensoriel $\otimes$ induit par la multiplication des entiers tel que $\forall k, p \in \mathbb{N}, v_k \otimes v_p = v_{k\cdot p}$.
On peut ensuite calculer le produit tensoriel de deux vecteurs $F = \sum\limits_{n\in\mathbb{N}} f(n)v_n$ et $G = \sum\limits_{n\in\mathbb{N}} g(n)v_n$, où f et g sont des fonctions arithmétiques :

$$F \otimes G = \sum_{n\in\mathbb{N}} \sum_{kp=n} f(k)g(p)v_n = \sum_{n\in\mathbb{N}} \sum_{d|n} f(d)g(n/d)v_n.$$

On remarque alors que

$$F \otimes G = \sum_{n\in\mathbb{N}} (f \star g)(n)v_n, \tag{2}$$

où $\star$ désigne la convolution de Dirichlet. Dans le cas des séries de Dirichlet, ce produit tensoriel correspond à la multiplication.

On peut remarquer que cette dynamique n'est pas exclusive à la convolution de Dirichlet, et peut s'étendre à plus d'algèbres d'incidence [49], qui sont une généralisation des convolutions de Dirichlet. Le principe de télescopage (qui donne la formule classique d'inversion pour les séquences) peut notamment aussi être vu comme l'inverse pour une convolution.

0.4.3 Extension de la convolution de Dirichlet

Un cas particulier (et utile) d'extension de la convolution de Dirichlet est le cas où l'une des fonction (par exemple dans l'équation (2)) n'est pas arithmétique mais peut s'écrire sous la forme $n \mapsto \frac{f(n)}{n}$, où f est une fonction arithmétique. Alors un rapide calcul montre que

$$\sum_{d|n} \frac{f(d)}{d} g(n/d) = \frac{1}{n} \sum_{d|n} f(d) g(n/d) \frac{n}{d} = \frac{(f \star (id \times g))(n)}{n}, \tag{3}$$

où $\times$ désigne la multiplication. En remarquant que pour toute fonction arithmétique f et g, on a

$$(f \times id) \star (g \times id) = (f \star g) \times id,$$

on s'assure que la précédente formule est bien cohérente avec la définition de la convolution de Dirichlet, même dans le cas où $f = id \times h$, où h est une fonction arithmétique.

Première partie

Méthode symbolique

Chapitre 1

Arbres de processus binaires

Ce chapitre s'appuie sur l'article [7], écrit en collaboration avec Olivier Bodini, Antoine Genitrini et Frédéric Peschanski.

Dans ce chapitre on introduit un modèle issu de la concurrence, en vue de faire une étude quantitative de certaines de ses caractéristiques. On considère des processus qui peuvent s'effectuer possiblement en parallèle, et qui ont des contraintes logiques et temporelles qui peuvent inhiber leur exécution. Par exemple on peut avoir une règle qui énonce que le processus b ne peut s'effectuer que si le processus a s'est déjà lancé, ou une règle qui énonce que si le processus c s'effectue le processus d ne pourra être effectué, etc. Le modèle que nous allons regarder est l'interprétation des processus parallèles en terme d'*arbres de calculs*. Malgré leur caractérisation algébrique directe, les structures sous-jacentes construites sur des *étiquetages croissants* sont plutôt complexes et leur étude demande une utilisation fine de la combinatoire analytique.

Ce chapitre représente une extension de travaux effectués en [4], qui montrait une bijection entre une sous-classe des processus parallèles purs et la classe des *arbres croissants généraux* [2]. Cette bijection a donné de nombreux résultats quantitatifs, notamment le nombre moyen d'exécutions concurrentes pour des processus, i.e une analyse en moyenne de l'effet connu sous le nom *d'explosion combinatoire*. L'opérateur parallèle y est considéré comme d'arité arbitraire, afin de pouvoir utiliser des *arbres généraux* dans les interprétations combinatoires. Dans la plupart des modèles d'algèbres de processus (cf. e.g. [40, 19]), au contraire, l'opérateur parallèle est considéré comme binaire, ce qui complexifie l'analyse combinatoire des objets. On peut en effet facilement transformer un arbre binaire en arbre d'arité quelconque en conservant la même sémantique, en revanche la transformation des arbres d'arité quelconque à binaire demande de faire un choix arbitraire. Cela modifie la répartition des différents arbres, les arbres comportant des nœuds d'arité élevée étant plus fréquents dans le cas binaire, puisque qu'à un nœud d'arité n correspond plus de 2^{n-2} arbres binaires équivalents.

On va fournir une caractérisation précise et des approximations asymptotiques en moyenne de mesures diverses sur les processus parallèles, notamment : (1) le nombre moyen d'exécutions possibles, et (2) la taille moyenne des arbres sémantiques. Cela donne une signification plus précise de l'*explosion combinatoire*, qui auparavant était juste connu comme étant exponentiel. On arrive de manière plus intéressante à caractériser la forme typique des arbres syntaxiques, à partir d'une décomposition par niveau qui se révèle utilisable. Cela donne une interprétation intéressante sur la notion de *profondeur de processus*. Le dernier résultat est une extension de l'analyse dans le cas où l'on considère l'opérateur parallèle sur des processus binaires qui commutent, et qui s'obtient en suivant le même schéma que l'analyse précédente.

1.1 Contexte

Le modèle présenté vient le la théorie de la concurrence. On dispose de différents acteurs qui peuvent accomplir des actions en parallèle, mais certaines actions sont contraintes à être effectuées après d'autres actions. On va modéliser l'ensemble des actions réalisables ainsi que leurs relations de précédence, par un arbre de dépendance que l'on nommera arbre syntaxique.

1.1.1 Arbres syntaxiques

On spécifie les processus parallèles purs selon la grammaire suivante :

— une action atomique, notée $a, b, c, \ldots$ est un processus ;

— le préfixe $a.P$ d'une action a et d'un processus P est un processus ;

— la composition $P_1 \parallel P_2$ d'exactement deux processus P_1 et P_2, est un processus.

Le processus suivant sera l'*exemple de référence* que l'on utilisera régulièrement dans ce chapitre :

$$(a.b) \parallel [(c \parallel d) \parallel (e.(f \parallel g))].$$

Ces termes de processus peuvent s'interpréter naturellement en tant que structures d'arbres, ainsi que l'on peut le voir sur la gauche de la Fig. 1.1. Des processus parallèles purs sont composés de nœuds d'action unaires, de feuilles d'action et de nœuds binaires parallèles. Des termes qui sont conformes à la grammaire ci-dessus seront dorénavant appelés *arbres syntaxiques*.

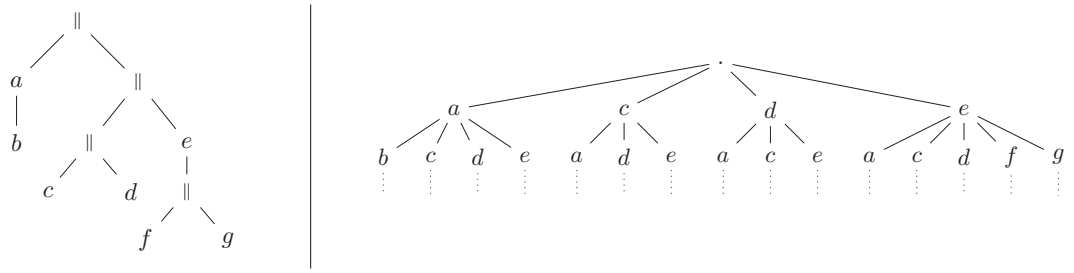


FIGURE 1.1 – Un arbre syntaxique de taille 7 (à gauche) et les premiers niveaux de son arbre sémantique.

On peut interpréter directement une grammaire pour une structure en forme d'arbre (fini) comme une *classe combinatoire*, à condition de décider d'une fonction de taille pour les objets appartenant à la classe. Puisque le nom des actions atomiques ne nous intéresse pas (ils sont supposés distincts), nos spécifications combinatoire vont les oublier. Cela donne la spécification suivante :

Définition 1.1. *La classe combinatoire des arbres de processus est :*

$$\mathcal{P} = \mathcal{Z} + \mathcal{Z} \times \mathcal{P} + \mathcal{P} \times \mathcal{P}, \text{ où } \mathcal{Z} \text{ marque le nombre d'actions.}$$

Cela peut se lire comme une grammaire : un objet dans la classe $\mathcal{P}$ est soit une feuille marquée par $\mathcal{Z}$, soit un nœud interne unitaire marqué par $\mathcal{Z}$ avec un sous-arbre dans $\mathcal{P}$, soit un nœud binaire non marqué avec deux sous-arbres dans $\mathcal{P}$. Le marqueur $\mathcal{Z}$ indique quels nœuds doivent être comptés dans la taille d'un objet. Ici on marque les nœuds correspondant aux actions de la grammaire. Ainsi la taille d'un arbre de processus P est le nombre d'occurrences d'actions dans l'arbre, et est noté $|P|$. Par exemple, la taille de l'exemple de référence est 7.

Nous conservons les notations habituelles de combinatoire analytique [1] pour la classe $\mathcal{P}$. Soit P_n le nombre de processus de taille n et $P(z)$ la fonction génératrice ordinaire [2] associée à la classe $\mathcal{P}$: elle vérifie $P(z) = \sum_{n>0} P_n\ z^n$. Et la notation pour l'extraction de coefficients d'une fonction génératrice est $[z^n]P(z) = P_n$.

En utilisant la définition 1.1 et la *méthode symbolique* (voir Section 0.2.2), on en déduit une équation fonctionnelle sur la spécification des processus :

$$P(z) = z + z \cdot P(z) + P(z) \cdot P(z).$$

On peut ainsi donner les premiers termes : $P(z) = z + 2\ z^2 + 6\ z^3 + 22\ z^4 + 90\ z^5 + 394\ z^6 + \ldots$
Il y a par exemple 90 arbres de taille 5 dans la classe $\mathcal{P}$.

1. C'est à dire que pour une classe $\mathcal{C}$ on note C_n les termes de sa suite énumérative et $C(z)$ sa fonction génératrice.
2. La fonction génératrice exponentielle $G(z)$ associée à la suite $(G_n)_n$ satisfait $G(z) = \sum_n G_n \frac{z^n}{n!}$.

Proposition 1.1.1. *La classe combinatoire $\mathcal{P}$ satisfait :*

$$P(z) = \frac{1 - z - \sqrt{1 - 6z + z^2}}{2} \qquad \textit{et} \qquad P_n \sim_{n\to\infty} \sqrt{\frac{3\sqrt{2} - 4}{4\ \pi\ n^3}} \cdot \left(3 - 2\sqrt{2}\right)^{-n}.$$

C'est une conséquence directe de l'utilisation de la méthode symbolique (cf. [24]), la méthode pour compter les nœuds est distincte[3] de [4]. Rappelons que $\rho_P = 3 - 2\sqrt{2}$ est la singularité dominante de P, qui est directement associée à la croissance exponentielle de $(P_n)_{n\in\mathbb{N}}$.

Dans [4, 5] il est décrit une variante des processus parallèles purs dans lesquels l'opérateur parallèle est d'arité quelconque. En terme de combinatoire, c'est un modèle plus simple que le cas binaire puisque les arbres de processus peuvent être identifiés avec les *arbres généraux* avec un seul type de nœuds : un nœud d'action suivi par un ensemble de sous-processus. En revanche l'opérateur binaire utilisé dans cette partie est plus proche des représentations algébriques des algèbres de processus (cf. e.g. [40, 19]).

1.1.2 Arbres sémantiques

Les arbres syntaxiques sont des objets qui doivent être interprétés sur un domaine *sémantique*. Notre modèle combinatoire vise à interpréter les comportements des processus comme des *arbres sémantiques* [14]. Une exécution est le résultat de la fusion des branches d'un arbre syntaxique. Par exemple, en utilisant notre exemple de référence de la Fig. 1.1, on note que l'exécution $\langle a, b, c, d, e, f, g\rangle$ est une exécution valide du processus mais $\langle a, b, c, d, f, e, g\rangle$ ne l'est pas car il n'est pas le résultat de la fusion de processus (l'action f ne peut pas précéder l'action e).

Le comportement complet d'un processus peut être mis sous la forme d'un arbre avec toutes les exécutions possibles avec des préfixes partagés : c'est l'*arbre* sémantique du processus.

L'arbre de droite de la Fig. 1.1 montre les premiers niveaux de l'arbre sémantique engendré par l'exemple de référence. C'est un fait connu de la littérature que les arbres sémantiques de processus purement parallèles sont exponentiellement plus larges que les arbres syntaxiques associés. On peut observer ce phénomène d'*explosion combinatoire* sur notre exemple de référence, en effet, malgré la petite taille de son arbre syntaxique (il a 7 nœuds comptés), son arbre sémantique induit est de taille 2360.

Les questions qui nous concernent sont en premier de nature quantitative : nous voulons trouver une signification mathématique précise de l'expression "explosion combinatoire". La mesure la plus significative du comportement est indubitablement le nombre *moyen* d'exécutions pour un processus donné. Une mesure plus fine et plus technique est requise pour déterminer proprement le nombre de partage de préfixe dans les arbres sémantiques. Une question plus qualitative est ensuite posée : quelle est la *forme typique* des arbres sémantiques ? Pour cela on exploite une décomposition des arbres sémantiques par niveaux. Un *niveau* ℓ d'un arbre sémantique est l'ensemble des nœuds qui correspondent à la ℓ-ième occurrence d'une action dans chacune de ses branches. Par exemple, Fig. 1.1 montre le premier et le deuxième niveau de l'arbre sémantique pour l'exemple de référence. Cette étude combinatoire permet notamment la création d'algorithmes efficaces pour génération d'arbres sémantiques, qui seront développés dans le chapitre 4.

On peut remarquer que les arbres sémantiques sont par nature d'arité quelconque, et il n'est du coup pas possible de retrouver l'arbre syntaxique à l'origine d'un arbre sémantique, puisque qu'un arbre sémantique peut correspondre à des arbres syntaxiques différents (par exemple $(P_1 \parallel P_2) \parallel P_3$ et $P_1 \parallel (P_2 \parallel P_3)$ induisent le même arbre sémantique). Ce modèle est en cela très différent de celui développé en [6], où la correspondance arbres syntaxique-sémantique est une bijection. On va donc avoir une modification par rapport au modèle d'arité quelconque, puisque certains arbres sémantiques seront comptés en plusieurs exemplaires (un pour chaque arbre syntaxique correspondant).

1.2 Processus binaire typique

Dans cette section, on va s'intéresser aux mesures typiques des arbres sémantiques dans le contexte de processus binaires parallèles associatifs. On donne en premier le nombre asymptotique

3. Généralement, tous les nœuds sont comptés, ce qui n'est pas le cas ici pour ceux d'arité binaire.

d'exécutions de processus de taille n, quand n tends vers $+\infty$. On raffine ensuite l'étude quantitative en considérant la taille totale des arbres sémantiques.

1.2.1 Nombre typique d'exécutions

Théorème 1.1. *Le nombre asymptotique moyen d'exécutions, noté $\bar{G}_n$, induits par des processus binaires de taille n, satisfait quand n tends vers $+\infty$:*

$$\bar{G}_n \sim_{n\to\infty} 3 \cdot \sqrt{\frac{\ln\frac{3}{2}-\frac{1}{3}}{6\sqrt{2}-8}} \cdot \left(\frac{3-2\sqrt{2}}{3\left(\ln\frac{3}{2}-\frac{1}{3}\right)}\right)^n \cdot n!.$$

On peut remarquer que $(3-2\sqrt{2})/(3\left(\ln\frac{3}{2}-\frac{1}{3}\right)) \approx 0.79287$, et ainsi la moyenne $\bar{G}_n$ est bien plus petite que $n!$ qui correspond au nombre d'exécutions dans le pire cas (où toutes les actions sont en parallèle). Le nombre moyen d'exécutions dans [4] était équivalent à $\frac{n!}{2^{n-1}}$. On en déduit que le modèle binaire provoque un nombre d'exécutions moyen plus important. La preuve de ce théorème suit le même plan de preuve que celle suivie dans [4] pour les arbres de processus n-aires. En revanche l'approche consistant à se reposer sur l'holonomie des fonctions pour dériver des équations simples n'est pas utilisable dans ce cas, et on est contraint de se débrouiller avec d'autres techniques de calcul.

Afin de calculer le nombre d'exécutions d'un processus, on exploite un isomorphisme entre les exécutions d'un arbre syntaxique et ses *étiquetages croissants*. On rappelle que l'étiquetage croissant d'un arbre est un étiquetage de chaque nœud avec un entier entre 1 et la taille de l'arbre tel que tout les nœuds ont une étiquette strictement plus grande que celle de leurs parents. Ces arbres appelés *arbres croissants* sont étudiés en détail dans [18, Chapitre 1].

Bien que dans notre modèle de processus binaire seul les nœuds étiquetés par une action sont pris en compte dans le calcul de taille, l'isomorphisme reste valide.

Lemme 1.1. *Soit P un processus binaire. Il y a une isomorphisme entre les exécutions de P et les étiquetages croissants des nœuds de P contenant une action.*

La preuve pour ce lemme peut directement être adaptée de [4]. Puisque les nœuds parallèles ne sont pas considérés pour l'étiquetage croissant, l'associativité de l'opération parallèle ne joue aucun rôle ici.

Sur la Fig. 1.2 les deux arbres croissants à gauche correspondent respectivement aux exécutions $\langle a,b,c,d,e,f,g\rangle$ et $\langle a,c,e,f,b,d,g\rangle$ de l'exemple de référence.

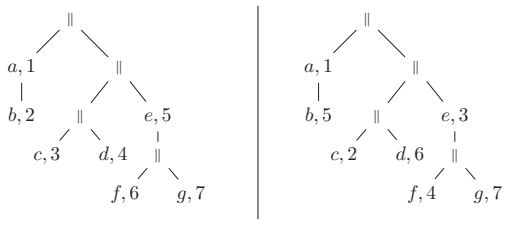


FIGURE 1.2 – Deux processus étiquetés croissants.

En combinatoire analytique, le produit avec contraintes de croissance permet d'encoder des contraintes de croissance sur les étiquettes des nœuds. Soit $\mathcal{A}$ et $\mathcal{B}$ deux classes combinatoires croissantes, alors la classe $\mathcal{A}^{\square} \star \mathcal{B}$ correspond aux objets étiquetés tels que la plus petite étiquette appartient à la première composante (dans $\mathcal{A}$). Pour plus de détail, voir [24, Chapitre II].

Proposition 1.2.1. *Soit $\mathcal{G}$ la classe des processus étiquetés croissants, et $G(z)$ la fonction génératrice exponentielle correspondante. On obtient :*

$$\mathcal{G} = \mathcal{G} \star \mathcal{G} + \mathcal{Z}^{\square} \star (\mathcal{G} + \mathcal{E}), \quad \textit{soit} \quad G(z) = -1 - \frac{3}{2} \cdot W\left(-\frac{2}{3}\exp\left(\frac{z-2}{3}\right)\right),$$

où W désigne la fonction W de Lambert, qui satisfait : $W(z)\cdot\exp(W(z)) = z$.

On peut trouver un bon nombre de résultats sur la fonction W de Lambert dans l'article de Corless *et al* [15]. Entre autre sa fonction génératrice est la suivante :

$$W(z) = \sum_{r\geq 1} w_r\ z^r, \qquad \text{où } w_r = \frac{(-r)^{r-1}}{r!}. \tag{1.1}$$

En prenant en compte que la série $G(z)$ est exponentielle, on peut facilement calculer les premiers termes des processus étiquetés croissants (en fonction du nombre d'actions) :

$$1, 3, 21, 243, 3933, 81819, \ldots$$

Démonstration. En utilisant la traduction du produit avec contrainte de croissance binaire (voir le paragraphe 0.2.4 des préliminaires) sur les séries formelles, on sait que la fonction génératrice exponentielle $G(z)$ est la solution analytique en 0 du système

$$\begin{cases} T'(z) - 2T(z)T'(z) - T(z) = 1 \\ T(0) = 0 \end{cases}$$

En utilisant une décomposition en élément simple de cette équation, on peut l'intégrer et obtenir l'équation

$$\frac{2}{3}(1 + G(z))e^{-\frac{2}{3}(1+G(z))} = e^{\frac{z}{3} - \frac{2}{3}\ln\frac{3}{2}}.$$

En définissant y et x tels que

$$\begin{cases} y = \frac{2}{3}(1 + G(z)) \\ x = e^{\frac{z}{3} - \frac{2}{3} - \ln\frac{3}{2}} \end{cases}$$

l'équation devient $ye^{-y} = x$, d'où l'apparition de la fonction W de Lambert. □

Pour la suite du calcul, l'article [4] utilise l'holonomie de la fonction trouvée pour utiliser des stratégies de "Guess and Prove", et on peut donc essayer de voir si notre résultat est toujours holonome. Puisque nous nous intéressons à des fonctions génératrices qui énumèrent des objets combinatoires, on peut restreindre la définition de fonction holonome de la manière suivante :

Définition 1.2. *Une série formelle $f(z)$ est* holonome *(ou D-finie) si elle satisfait une équation différentielle linéaire dont les coefficients sont rationnels en z :*

$$q_0(z)\frac{d^r}{dz^r}f(z) + q_1(z)\frac{d^{r-1}}{dz^{r-1}}f(z) + \cdots + q_r(z)f(z) = 0, \tag{1.2}$$

pour des polynômes $q_s(X) \in \mathbb{Q}[X]$ (avec q_0 différent de 0).

Or la fonction W de Lambert n'est *pas* holonome, ainsi que démontré par exemple dans [28], donc on peut légitimement supposer que la fonction génératrice $G(z)$ de la proposition 1.2.1 n'est pas non plus holonome. On va le prouver dans la proposition suivante, la non-holonomie ne se propageant pas forcément par la composition de fonction. Ainsi le passage au modèle d'opérateur parallèle binaire fait s'effondrer le schéma de preuve utilisé en [4],

Proposition 1.2.2. *$G(z)$ n'est pas holonome.*

Démonstration. Par les théorèmes classiques d'analyse complexe dans le contexte des équations différentielles, (e.g. [30, Chapitre 9]), toute fonction $f(z)$, analytique en 0, peut être prolongée analytiquement sur n'importe quel chemin qui évite les racines de q_0 (défini par l'équation (1.2)). Une conséquence directe est que toutes les singularités de $f(z)$ sont des racines de q_0.

On fait une preuve par l'absurde : Supposons que $G(z)$ est holonome. Dans notre contexte, une des singularités de la fonction W de Lambert est $-e^{-1}$, donc $G(z)$ a une singularité en $-1 + 3\ln\frac{2}{3}$ qui correspond à une racine du polynôme rationnel q_0. Donc $\ln\frac{2}{3}$ est une racine d'un polynôme rationnel, ce qui contredit le fait que ce nombre est transcendant. □

On va maintenant prouver le théorème 1.1 en utilisant la forme de la fonction génératrice $G(z)$ de la classe des processus étiquetés croissants $\mathcal{G}$ prouvée en 1.2.1.

Démonstration. (théorème 1.1) La singularité dominante de $G(z)$ est atteinte quand la fonction W de Lambert atteint $-e^{-1}$.

Ainsi la singularité dominante de $G(z)$ est $\eta = -1 + 3\ln 3/2$.
Par des calculs classiques sur la fonction W de Lambert, on obtient :

$$W(-e^{-1} - h) =_{h\to 0} -1 + \sqrt{2e\ h} + o(\sqrt{h}).$$

En l'utilisant conjointement avec le développement de Taylor suivant :

$$-\frac{2}{3}\exp\left(\frac{z-2}{3}\right) = -e^{-1}\cdot\left(1 - \frac{\eta}{3}\cdot\left(1 - \frac{z}{\eta}\right)\right) + o\left(1 - \frac{z}{\eta}\right),$$

on obtient :

$$W\left(-\frac{2}{3}\exp\left(\frac{z-2}{3}\right)\right) = -1 + \sqrt{\frac{2\eta}{3}\left(1 - \frac{z}{\eta}\right)} + o\left(\sqrt{1 - \frac{z}{\eta}}\right).$$

La singularité de $G(z)$ est isolée, et est analytique autour dans un $\Delta-$domaine à cause de sa singularité de type racine carré. $G(z)$ est donc $\Delta-$analytique. On peut ainsi utiliser les théorèmes de transfert classiques dus à Flajolet et Odlyzko [22], détaillés dans [24], qui donnent :

$$n!\ [z^n]G(z) \sim_{n\to\infty} n!\cdot\frac{3}{2}\cdot\sqrt{\frac{\ln\frac{3}{2} - \frac{1}{3}}{2\pi\ n^3}}\cdot\left(\frac{1}{3\left(\ln\frac{3}{2} - \frac{1}{3}\right)}\right)^n.$$

La moyenne est finalement obtenue en normalisant par P_n. □

1.2.2 Taille typique des arbres sémantiques

Les arbres sémantiques partagent des préfixes de calculs en commun, ce qui peut se voir en comptant les feuilles, ainsi que cela a été fait précédemment dans la section 1.2.1. Le but de cette sous-section est de calculer le nombre asymptotique moyen de noeuds à un niveau donné d'un arbre sémantique, et plus précisément le nombre moyen de nœuds pour chaque niveau.

Théorème 1.2. *Soit $\overline{L}_n$ la taille moyenne des arbres sémantiques induits par des processus binaires de taille n, et $\overline{L}_n^{n-\ell}$ le nombre moyen de nœuds au niveau $n-\ell-1$ des arbres sémantique ($\ell \in \{0,\ldots,n-1\}$). Les valeurs asymptotiques de ces quantités, quand n tends vers $+\infty$, satisfont :*

$$\overline{L}_n \sim_{n\to\infty} e\cdot\overline{G}_n, \qquad \text{et} \qquad \overline{L}_n^{n-\ell} \sim_{n\to\infty} \frac{\overline{G}_n}{\ell!}.$$

Définition 1.3. *Soit P un arbre de processus. Commençant par P, enlevons itérativement des feuilles de la structure d'arbre. Si la structure restante C ne contient pas de feuilles étiquetées par l'opérateur $\|$, alors C est appelé une* coupe admissible *de P. La* taille *d'une coupe admissible est le nombre d'actions qu'elle contient.*

L'arbre de processus complet T est défini comme une coupe admissible, mais le processus vide (après avoir enlevé tout les nœuds de T) n'est pas une coupe admissible. Sur la Fig. 1.3, deux coupes admissibles obtenues depuis notre exemple de référence (Fig. 1.1) sont dessinées.

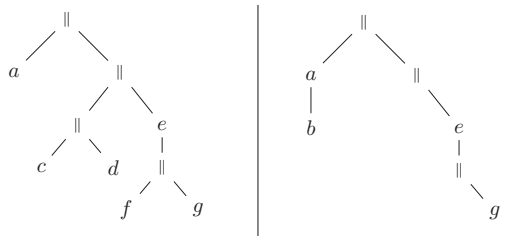


FIGURE 1.3 – Deux coupes admissibles.

Un étiquetage croissant des actions d'une coupe admissible donne une *coupe admissible croissante.*

Lemme 1.2. *Soit P un arbre de processus. Le nombre de nœuds au niveau $i-1$, pour $i \in \{1, \dots, |P|\}$, est égal au nombre de coupes admissibles de taille i du processus P.*

Cette bijection va nous permettre d'utiliser les coupes admissibles pour étudier la taille moyenne des arbres sémantiques.

Proposition 1.2.3. *La spécification suivante énumère toutes les coupes admissibles induites par les arbres syntaxiques de la même taille.*

$$\mathcal{C} = \mathcal{C} \times \mathcal{C} + \mathcal{C} \times \mathcal{P} + \mathcal{P} \times \mathcal{C} + \mathcal{U}^{\square} \star \mathcal{Z} \times (\mathcal{C} + \mathcal{P} + \mathcal{E}),$$

où $\mathcal{Z}$ marque tous les nœuds et $\mathcal{U}$ les nœuds de la coupe admissible croissante. Ainsi,

$$\frac{\partial C(z,u)}{du} = \frac{\partial C(z,u)}{du}\left(2C(z,u) + 2P(z)\right) + z\left(C(z,u) + P(z) + 1\right).$$

D'où

$$C(z,u) = -\left(1 + P(z)\right) - \frac{3}{2} \cdot W\left(-\frac{2}{3} \cdot (1 + P(z)) \cdot \exp\left(\frac{uz}{3} - \frac{2}{3}\left(1 + P(z)\right)\right)\right).$$

Une approche analogue à celle présentée dans la preuve de la proposition 1.2.1 donne le résultat.

Proposition 1.2.4. *$C(z,u)$ n'est pas holonome.*

Démonstration. de la proposition 1.2.4 On a $C(z,u) = \sum_{\ell,n} C_{\ell,n} u^{\ell} z^n$. La diagonale ΔC est telle que $\Delta C(z) = \sum_n C_{n,n} z^n$. On reconnaît $\Delta C(z) = G(z)$. Ainsi par le résultat de Lipschitz [36], on en conclut que $C(u,z)$ ne peut pas être holonome (sinon $G(z)$ serait elle aussi holonome). $\square$

À cause de ce résultat, la preuve du théorème 1.2 n'est pas évidente, et il sera prouvé par une analyse détaillée de l'équation satisfaite par $C(z,u)$.

Preuve du théorème 1.2

Le nombre de noeuds au niveau ℓ d'un arbre sémantique est égal au nombre de coupes croissantes avec l noeuds, ce qui nous donne :

$$L_n^{n-\ell} = (n-\ell)![u^{n-\ell}\ z^n]C(u,z), \qquad \text{d'où} \qquad L_n = \sum_{\ell=1}^{n} L_n^{n-\ell}.$$

Remarquons que les nombres moyen $\overline{L}_n^{n-\ell}$ et $\overline{L}_n$ sont obtenus en divisant ces valeurs par T_n.

Soit $\ell \in \{0, \dots n-1\}$. En utilisant l'équation (1.1) on prouve les formules suivantes :

$$\begin{aligned}
L_n^{n-\ell} &= (n-\ell)![u^{n-\ell}\ z^n]C(z,u) \\
&= -\frac{3}{2}\sum_{r\geq 1} w_r (n-\ell)![u^{n-\ell}\ z^n]\left(-\frac{2}{3} \cdot (1+P(z)) \cdot \exp\left(\frac{uz}{3} - \frac{2}{3}\left(1+P(z)\right)\right)\right)^r \\
&= -\frac{3}{2}\sum_{r\geq 1} w_r (n-\ell)![u^{n-\ell}\ z^n]\left(\sum_{s\geq 0}\frac{1}{s!}\left(\frac{ruz}{3}\right)^s\right)\left(-\frac{2}{3} \cdot (1+P(z)) \cdot \exp\left(-\frac{2}{3}\left(1+P(z)\right)\right)\right)^r \\
&= -\frac{3}{2}\sum_{r\geq 1} w_r [z^n]\left(\frac{rz}{3}\right)^{n-\ell}\left(-\frac{2}{3} \cdot (1+P(z)) \cdot \exp\left(-\frac{2}{3}\left(1+P(z)\right)\right)\right)^r \\
&= -\frac{3}{2}\sum_{r\geq 1} w_r \left(\frac{r}{3}\right)^{n-\ell}\left(-\frac{2}{3}e^{-\frac{2}{3}}\right)^r [z^{\ell}]\left((1+P(z)) \cdot \exp\left(-\frac{2}{3}P(z)\right)\right)^r
\end{aligned}$$

Pour compléter la preuve, il reste à réaliser l'extraction de $[z^{\ell}]\left((1+P(z)) \cdot \exp\left(-\frac{2}{3}P(z)\right)\right)^r$. Le résultat de cette extraction sera vu comme un polynôme en r dont le degré dépends de ℓ. Une remarque important est que ce polynôme ne dépend pas de n. Pour réaliser l'extraction on va donc prouver que $[z^{\ell}]\left((1+P(z)) \cdot \exp\left(-\frac{2}{3}P(z)\right)\right)^r = \frac{1}{\ell!}\left(\frac{r}{3}\right)^{\ell} + p_{\ell-1}(r)$, où $p_{\ell-1}(r)$ est un polynôme de degré au plus $\ell - 1$, avec des coefficients positifs. Soit $f(z) = \left((1+P(z)) \cdot \exp\left(-\frac{2}{3}P(z)\right)\right)^r$. On trouve les coefficients de $f(z)$ par récurrence :

— $[z]f(z) = \frac{r}{3}$.

— Soit $[z^\ell]f(z) = \frac{1}{\ell!}\left(\frac{r}{3}\right)^\ell + p_{\ell-1}(r)$, avec $p_{\ell-1}(r)$ un polynôme de degré au plus $\ell - 1$, avec des coefficients positifs

— On s'intéresse maintenant à $[z^{\ell+1}]f(z)$. On note qu'il y a la relation sympathique suivante entre $f(z)$ et sa dérivée :

$$f'(z) = f(z) \cdot \frac{\ell}{3}\left(1 + \frac{z}{1 - z - 2P(z)}\right).$$

Ainsi,

$$[z^{\ell+1}]f(z) = \frac{1}{\ell+1}[z^\ell]f'(z) = \frac{1}{\ell+1}[z^\ell]f(z)\cdot\frac{\ell}{3}\left(1 + \frac{z}{1 - z - 2P(z)}\right) = \frac{1}{(\ell+1)!}\left(\frac{r}{3}\right)^{\ell+1} + \tilde{p}_\ell(r),$$

avec $\tilde{p}_\ell(r)$ un polynôme de degré au plus ℓ, avec des coefficients positifs.

Ceci conclut la preuve par récurrence.

Puisque tout les polynômes que l'on considère ont des coefficients positifs, on en déduit :

$$L_n^{n-\ell} = (n-\ell)![u^\ell\ z^n]C(u,z) \geq -\frac{3}{2}\sum_{r\geq 1}\frac{w_r}{l!}\left(-\frac{2}{3}e^{\frac{-2}{3}}\right)^r\left(\frac{r}{3}\right)^n. \tag{1.3}$$

On calcule G_n :

$$\begin{aligned} G(z) &= -1 - \frac{3}{2}\cdot W\left(-\frac{2}{3}\exp\left(\frac{z-2}{3}\right)\right), \\ &= -1 - \frac{3}{2}\cdot\sum_{r\geq 1} w_r\ \left(-\frac{2}{3}\exp\left(\frac{z-2}{3}\right)\right)^r, \text{(en appliquant l'équation (1.1))} \\ &= -1 - \frac{3}{2}\cdot\sum_{r\geq 1} w_r\ \left(-\frac{2}{3}\exp\left(\frac{-2}{3}\right)\right)^r \exp\left(\frac{rz}{3}\right), \end{aligned}$$

Donc

$$G_n = n![z^n]G(z) = -n!\frac{3}{2}\cdot\sum_{r\geq 1} w_r\ \left(-\frac{2}{3}e^{\frac{-2}{3}}\right)^r\frac{\left(\frac{r}{3}\right)^n}{n!}, \tag{1.4}$$

$$= -\frac{3}{2}\cdot\sum_{r\geq 1} w_r\ \left(-\frac{2}{3}e^{\frac{-2}{3}}\right)^r\left(\frac{r}{3}\right)^n. \tag{1.5}$$

On en déduit enb utilisant (1.5) et (1.3) que :

$$L_n^{n-\ell} \geq \frac{G_n}{\ell!}.$$

De plus, si l'on fixe ℓ, chaque terme n'a pas été pris en compte peut être vu comme le produit d'une constante et d'un spécifique G_k, pour $k < n$. On remarque que $\sum_{k<n} G_k = o(G_n)$ pour $k < n$ (en utilisant la preuve du théorème 1.1) ce qui nous permet de conclure que

$$L_n^{n-\ell} \sim_{n\to\infty} \frac{G_n}{\ell!}.$$

En prenant la moyenne on en déduit les résultats annoncés :

$$\bar{L}_n \sim_{n\to\infty} e\cdot\bar{G}_n, \qquad \text{et} \qquad \bar{L}_n^{n-\ell} \sim_{n\to\infty} \frac{\bar{G}_n}{\ell!}.$$

Le premier résultat peut être obtenu en sommant les $\bar{L}_n^{n-\ell}$ avec précaution, en séparant la somme en deux parties : les ℓ petits, qui vont apporter l'équivalent, et les ℓ grands, que l'on va montrer négligeables.

1.3 Processus binaire typique commutatif

Dans l'article [4] les méthodes de résolutions utilisées s'adaptent très mal dans le cas où l'opération parallèle comme commutative. Bien entendu, considérer des arbres de processus non-planaires, i.e identifier les arbres de processus à des commutations sur les branches parallèles près, a un impact important sur les calculs, et par conséquent les mesures typiques peuvent varier beaucoup. Le fait que l'on ai réussi à analyser nos arbres croissants binaires-unaires sans utiliser les théorèmes sur les fonctions holonomes apporte un sous-produit intéressant : les techniques de calculs utilisées pour le cas non-commutatif s'adaptent naturellement dans le cas commutatif. Ainsi d'un point de vue technique la commutativité devient une simple variation du travail précédent, ainsi qu'on va le montrer dans cette section.

1.3.1 Arbres de processus non-planaires

Définition 1.4. *La classe combinatoire des arbres non-plans est spécifiée comme suit :*

$$\mathcal{Q} = \mathcal{Z} + \mathcal{Z} \times \mathcal{Q} + MSet_2[\mathcal{Q}], \qquad \text{où } \mathcal{Z} \text{ marque les nœuds contenant une action.}$$

$$\text{Par conséquent,} \qquad Q(z) = z + z \cdot Q(z) + \frac{1}{2}\left(Q^2(z) + Q(z^2)\right).$$

L'opérateur MSet est un outil classique de la combinatoire pour construire des multiensembles (i.e des ensembles où l'on autorise la répétition des éléments). La variante MSet_2 construit des multiensembles deux deux éléments (potentiellement identiques). Donc les deux élément commutent, ce qui fournit une manière naturelle d'encoder la commutativité directement dans la syntaxe. Une remarque importante est que MSet_2 opère sur des structures non étiquetées. Les premiers coefficients de la fonction génératrice peuvent être facilement calculés : $Q(z) = z + 2\ z^2 + 4\ z^3 + 11\ z^4 + 30\ z^5 + 92\ z^6 + \dots$.

Une difficulté ici est que Q ne peut être résolu explicitement, et une analyse de singularité semble inévitable.

Proposition 1.3.1. *Soit ρ_Q la singularité dominante de Q, alors*

$$Q_n \sim_{n\to\infty} \frac{\gamma_Q}{2\sqrt{\pi}} \cdot \frac{\rho_Q^{-n}}{n\sqrt{n}},$$

avec les approximations suivantes : $\rho_Q \approx 0.247834710963325$ et $\gamma_Q \approx 0.942857697692029$.

Démonstration. Des méthodes de bootstrap données en [24, P.475–477] permettent d'estimer ρ_Q. Une autre technique basée sur de la dichotomie converge quadratiquement vers la singularité, et nous donne une approximation plus précise. On remarque que $\rho_Q^{-1} \approx 4.034947$ est plus petit que ρ_P^{-1}. □

1.3.2 Nombre typique d'exécutions et taille de processus non-planaires

Maintenant les résultats quantitatifs pour le cas commutatifs suivent.

Théorème 1.3. *Le nombre asymptotique moyen du nombre d'exécutions, noté $\bar{H}_n$, induit par un processus non-planaire de taille n, satisfait quand n tend vers $+\infty$:*

$$\bar{H}_n \sim_{n\to\infty} 2\sqrt{2\left(\ln 2 - \frac{1}{2}\right)}\left(\frac{\rho_Q}{2\left(\ln 2 - \frac{1}{2}\right)}\right)^n \cdot n! \sim 4\sqrt{\pi\ n\left(\ln 2 - \frac{1}{2}\right)}\left(\frac{\rho_Q\ n}{2e\left(\ln 2 - \frac{1}{2}\right)}\right)^n.$$

Afin de prouver le Théorème 1.3 on suit la même stratégie que dans le cas binaire planaire.

Proposition 1.3.2. *Soit $\mathcal{H}$ la classe des processus étiquetés non-planaires croissants, et $H(z)$ la fonction génératrice exponentielle correspondante. On obtient :*

$$\mathcal{H} = Set_2\mathcal{H} + \mathcal{Z}^{\square} \star (\mathcal{H} + \mathcal{E}), \qquad \text{donc} \qquad H(z) = -1 - 2 \cdot W\left(-\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right).$$

La fonction $H(z)$ n'est pas holonome.

On rappelle que l'opérateur Set_2 construit les ensembles de deux objets étiquetés. A cause de l'étiquetage, cela correspond directement à une liste non-ordonnée de deux éléments. On peut trouver toutes les informations nécessaires sur cette construction dans [24, Chapitre II]. Dans la proposition ci-dessus, on contraint les étiquettes à être croissantes.

Démonstration. La spécification se traduit en l'équation différentielle $H'(z) = H'(z) \cdot H(z) + H(z) + 1$. Le reste de la preuve est similaire à celle de la proposition 1.2.1. □

Les premiers termes de la fonction génératrice exponentielle $H(z)$ sont $1, 2, 8, 52, 472, 5504, \ldots$, qui ne figure pas dans l'encyclopédie des suites de Sloane (OEIS). Ces termes doivent être comparés aux termes relativement plus grands de $G(z)$.

1.3.3 Taille typique des arbres de calculs

Les résultats suivants sont obtenus d'une manière similaire à ceux de la section 1.2

Théorème 1.4. *Soit $\overline{M_n}$ la taille moyenne des arbres sémantiques induits par des arbres de processus non-planaires de taille n, et $\overline{M_n^{n-\ell}}$ le nombre moyen de nœuds au niveau $n-\ell-1$ de l'arbre sémantique ($\ell \in \{0, \ldots, n-1\}$). Les valeurs asymptotiques de ces moyennes, quand n tend vers $+\infty$, satisfont :* $\overline{M}_n \sim_{n\to\infty} e \cdot \overline{H}_n$, *et* $\overline{M}_n^{n-\ell} \sim_{n\to\infty} \frac{\overline{H}_n}{\ell!}$.

Proposition 1.3.3. *La spécification suivante énumère toutes les coupes admissibles croissantes induites par des processus non-planaires de la même taille.*

$$\mathcal{D} = Set_2\mathcal{D} + \mathcal{D} \star \mathcal{Q} + \mathcal{U}^{\square} \star \mathcal{Z} \star (\mathcal{D} + \mathcal{Q} + 1),$$

où $\mathcal{Z}$ marque tous les nœuds du processus et $\mathcal{U}$ les nœuds de la coupe croissante admissible. En conclusion,

$$D(z,u) = -(1 + Q(z)) - 2 \cdot W\left(-\frac{1}{2} \cdot (1 + Q(z)) \cdot \exp\left(\frac{uz}{2} - \frac{1}{2}(1 + Q(z))\right)\right).$$

La fonction $D(u,z)$ n'est pas holonome.

Rappelons les mesures typiques pour les processus purement parallèles :

Modèle	Arité arbitraire [4]	Binaire 1.2	Binaire commutatif 1.3
Nombre d'exécutions	$\Theta(0.5^n \cdot n!)$	$\Theta(\beta^n \cdot n!)$; $\beta \approx 0.79287$	$\Theta(\gamma^n \cdot n!)$; $\gamma \approx 0.64156$

Ce tableau renforce l'idée que passer sur le cas non-planaire ne change pas la *nature* de l'objet étudié : l'explosion combinatoire arrive dans tout les cas. On peut cependant remarquer que les valeurs asymptotiques ci-dessus diffèrent d'un facteur exponentiel, mais le "gain" de quotienter par commutativité ne compense pas – loin de là – la croissance exponentielle.

1.4 Conclusion

Nous avons dans ce chapitre montré l'influence de la binarité du parallélisme de processus sur le nombre d'exécutions moyen des processus. On peut donc constater que le nombre asymptotique moyen d'exécutions est plus élevé d'un facteur exponentiel dans ce cas, mais que le profil moyen des arbres sémantiques est similaire dans les deux cas, avec l'essentiel des nœuds dans les derniers niveaux. On a eu aussi l'occasion d'étudier la modification apportée par le fait de considérer que l'opérateur de parallélisme binaire des processus commute. On trouve que le nombre asymptotique moyen d'exécutions est encore identique aux deux cas précédents à un facteur exponentiel près, et le profil moyen des arbres sémantiques associés est aussi similaire.

Le chapitre montre donc que les changements de modèles sur l'opérateur de parallélisme des processus ne modifient pas intrinsèquement les mesures sur les exécutions associés au modèles.

Chapitre 2

Arbres de processus avec synchronisation

Ce chapitre s'appuie sur l'article [9], écrit en collaboration avec Antoine Genitrini et Olivier Bodini, qui constitue une extension du modèle présenté dans le chapitre précédent. On cherche à rajouter au modèle de processus parallèles purs une notion de *synchronisation*, qui implique que plusieurs actions doivent se réaliser *en même temps*. On se trouve donc en présence de structures plus complexes, puisque la relation d'ordre entre les actions ne peut plus être exprimée uniquement avec un arbre, mais avec un graphe orienté cyclique (DAG).

La principale difficulté pour spécifier ce type de structure vient d'objets de la forme suivante : deux séquences croissantes parallèles, avec un ancêtre commun (qui a une étiquette plus petite que tout le monde), et un descendant commun (qui a une étiquette plus grande que tout le monde). Cela se spécifie très mal en utilisant l'opérateur de contrainte de croissance binaire de Greene (défini dans les préliminaires dans le paragraphe 0.2.4), qui est le produit avec contrainte de croissance binaire, puisqu'il faut spécifier tous les ordres possibles d'entremêlement des deux séquences, et nécessite un nombre exponentiel de termes. On va donc essayer de trouver une meilleure méthode pour spécifier ce type de classes combinatoires. Une des difficultés est que le problème sous-jacent, compter les extensions linéaires d'un ordre partiel, est dur dans le cas général [13], ce qui implique que l'on ne réussira probablement pas à trouver une spécification générale et efficace.

On va notamment s'inspirer pour cela des travaux de Stanley [53], qui dans un article a introduit une interprétation géométrique pour compter le nombre d'ordres totaux satisfaisant un ordre partiel donné. Les éléments du poset définissent un hypercube unitaire dans lequel chaque ordre total correspond à un polytope spécifique de l'hypercube. Ainsi le nombre d'ordres compatibles est directement relié à un volume et peut être calculé en utilisant une intégration multivariée. On va pouvoir par exemple encoder l'exemple précédent (le mélange de deux séquences avec un ancêtre et un descendant) avec la méthode de Stanley et le transformer en une équation vérifiée par la fonction génératrice. Grâce à cette approche, on arrive à éviter l'explosion en taille de la spécification, aux prix d'un calcul d'intégrale possiblement complexe. Cela nous permet tout de même dans un nombre de cas particuliers de déterminer une formule exacte pour la série génératrice.

Nous allons présenter en détail dans ce chapitre une manière d'intégrer l'approche de Stanley dans le contexte de la méthode symbolique, et l'illustrer dans un cas particulier : énumérer les arbres croissants avec *exactement* une répétition d'étiquettes, qu'on appellera *arbres synchronisés*.

2.1 Approche de Stanley enrichie

Dans son article de 1986 [53] puis dans son livre [52, chapitre 3], Stanley développe une stratégie pour énumérer le nombre d'extensions linéaires, ou chaînes, d'un ensemble partiellement ordonné ou *poset*. Pour cela il a réussi à donner une interprétation géométrique d'un poset et ses extensions linéaires en tant que sous-espace d'un hypercube unitaire, et des simplexes qui le partitionne. Donc le nombre d'extensions linéaires d'un poset est réduit à un calcul d'intégrale avec comme domaine un hypercube (donc plusieurs variables). On rappelle ici que le calcul exact du nombre d'extensions

linéaires d'un poset a été prouvé $\#P$ complet par Brightwell et Winkler [13]. La difficulté de ce calcul a été transformé par Stanley en la difficulté du calcul exact de certaines intégrales multiples.

Je vais donc présenter par la suite des adaptations de l'approche de Stanley qui vont nous permettre d'avoir plus d'information que le nombre d'extensions linéaires. En effet, en insérant des variables formelles dans les intégrales on arrive à séparer les extensions linéaires selon leurs propriétés structurelles.

Pour le reste du chapitre, on notera l'ordre naturel sur les entiers $\leq$ lorsque l'on voudra ordonner les entiers $\{1, \ldots, n\}$ pour un quelconque entier positif n.

Définition 2.1. *Un ordre partiel $\leq_\alpha$ (ou poset) sur E est une relation entre des éléments de E tel que :*

- $\leq_\alpha$ *est* réflexive *:* $\forall x \in E, x \leq_\alpha x$
- $\leq_\alpha$ *est* antisymétrique *:* $\forall x, y \in E, x \leq_\alpha y, y \leq_\alpha x \Rightarrow x = y$
- $\leq_\alpha$ *est* transitive *:* $\forall x, y, z \in E, x \leq_\alpha y, y \leq_\alpha z \Rightarrow x \leq_\alpha z$

On rappellera qu'une *extension linéaire* (ou chaîne) d'un ordre partiel $\leq_A$ est un ordre total compatible avec $\leq_\alpha$, i.e., un ordre $\leq_A$ sur E tel que $\forall x, y \in E, x \leq_\alpha y \Rightarrow x \leq_A y$. On note $\mathcal{LE}(\leq_\alpha)$ l'ensemble des extensions linéaires de $\leq_\alpha$.

Le principal résultat de Stanley pour l'énumération des extensions linéaires est le suivant :

$$|\mathcal{LE}(\leq_\alpha)| = n! \cdot \mathrm{Vol}(\leq_\alpha),$$

où $\mathrm{Vol}(\leq_\alpha)$ est le volume du polytope associé à $\leq_\alpha$, qui est défini dans la définition 2.2.

Exemple 2.1. *Considérons l'ordre partiel $\succ$: $z \succ a, z \succ b, a \succ t, b \succ t$. Alors le nombre d'extensions de $\succ$ est $4! \int_{z=0}^{1} \int_{t=0}^{z} \int_{x=t}^{z} \int_{y=t}^{z} dzdtdxdy = 2$.*

La première adaptation de la stratégie de Stanley consiste à voir l'hypercube unitaire en tant qu'un hypercube formel de $[0, z]^n$, avec z une variable dans $[0, 1]$, et l'utiliser comme domaine d'intégration. Cela va nous permettre de pouvoir étendre le travail de Stanley sur les fonctions génératrices. On retrouve bien entendu tous les résultats de Stanley en évaluant pour $z = 1$.

On définit donc naturellement le domaine d'intégration formel qui est associé au domaine d'intégration d'un poset dans l'approche de Stanley.

Définition 2.2. *Soit $\leq_\alpha$ un ordre partiel sur un ensemble fini de taille n, et z une variable dans $[0, 1]$. On définit $E_{\leq_\alpha}(z) = \{(z_1, \ldots, z_n) \in [0, z]^n \mid \forall i \neq j \in \{1, \ldots, n\},\ i \leq_\alpha j \Rightarrow z_i \leq z_j\}$ comme le* domaine associé *à $\leq_\alpha$.*

En utilisant le contexte formel, on obtient immédiatement les deux résultats suivants :

Lemme 2.1. *Soit $\leq_\alpha$ un ordre partiel sur un ensemble fini de taille n, et z une variable dans $[0, 1]$.* $\bigcup_{\leq_A \in \mathcal{LE}(\leq_\alpha)} E_{\leq_A}(z) = E_{\leq_\alpha}(z)$

Ce premier lemme décrit simplement le fait tous les domaines associés aux extensions linéaires d'un ordre partiel sont inclus dans le domaine de l'ordre partiel, et que pour tout élément appartenant au domaine d'un ordre partiel, il existe une extension linéaire dont le domaine associé contient cet élément.

Lemme 2.2. *Soit $\leq_A$ et $\leq_B$ deux différentes extensions linéaires de $\mathcal{LE}(\leq_\alpha)$, et μ la mesure de Lebesgue sur $[0, z]^n$. Alors $\mu(E_{\leq_B}(z) \cap E_{\leq_A}(z)) = 0$.*

Ce deuxième lemme énonce le fait que l'intersection des domaines associés de deux ordres totaux quelconques est de dimension inférieure à celle du domaine d'un quelconque ordre partiel qui satisfait les deux ordres totaux.

Exemple 2.2. *Considérons l'ensemble $\{1, 2, 3\}$, et l'ordre partiel $2 \leq_\alpha 1$. Alors le domaine de $\leq_\alpha$ est $E_{\leq_A}(z) = \{z_1, z_2, z_3 \in [0, z]^3 : z_2 \leq z_1\}$. Les ordres totaux compatibles avec $\leq_\alpha$ sont $2 \leq 1 \leq 3$,*

$2 \leq 3 \leq 1$ et $3 \leq 2 \leq 1$, et l'union de leurs domaines fait bien $E_{\leq_A}(z)$ conformément au lemme 2.1. On remarque aussi que

$$\{z_1, z_2, z_3 \in [0, z]^3 \ : z_3 \leq z_2 \leq z_1\} \cap \{z_1, z_2, z_3 \in [0, z]^3 \ : z_2 \leq z_3 \leq z_1\} = \{z_1, z_2, z_3 \in [0, z]^3 \ : z_3 = z_2 \leq z_1\},$$

qui est de dimension inférieure à celle de $E_{\leq_A}(z)$.

A l'aide de cette approche formelle, on est maintenant prêt à énumérer des structures croissantes et en particulier la famille qui nous intéresse : les arbres croissants avec exactement une étiquette répétée.

2.2 Produit partiellement ordonné

On va motiver les résultats de cette section avec un exemple. On veut compter le nombre d'*arbres synchronisés*, qui sont des arbres étiquetés tel que n'importe quel chemin de la racine à une feuille soit strictement croissant et qu'il y ait exactement deux nœuds avec la même étiquette. Le nombre de ces objets est compté par la suite suivante :

$$1, 11, 122, 1518, 21423, 340869, 6058980, 119218860, 2575293165, 60628447215, \ldots$$

Pour pouvoir les énumérer, on va légèrement modifier leur représentation en rassemblant les deux nœuds avec la même étiquette, ainsi qu'on peut le voir sur l'exemple de la Figure 2.1. Afin que la notion de taille soit cohérente entre les deux représentations, on va définir comme taille le nombre d'étiquettes différentes utilisés (donc 10 dans le cas de l'exemple de la Figure 2.1).

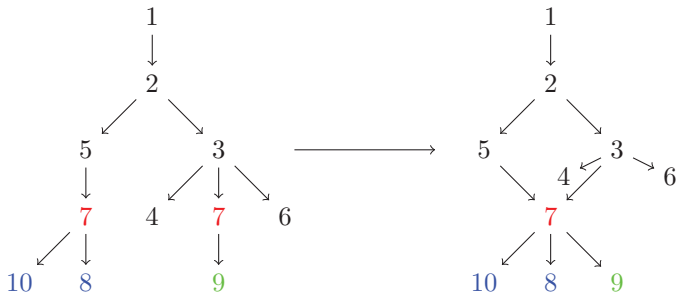


FIGURE 2.1 – Un arbre synchronisé (gauche) et sa représentation réduite (droite)

Le produit avec contrainte de croissance binaire n'est pas suffisamment expressif pour spécifier convenablement les relations entre différents nœuds plus petits qu'un même nœud (par exemple les nœuds $3, 5, 7$ de la Figure. 2.1). On peut le spécifier en réalisant une liste exhaustive de tous les ordres compatibles, mais la taille de cette liste peut devenir exponentielle.

Pour spécifier les arbres synchronisés, il nous faut donc définir un produit partiellement ordonné de classes combinatoires, qui spécifie un ordre partiel sur les plus petits éléments des différentes classes.

Définition 2.3. *Soit $\mathcal{A} = \star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)$ la construction qui prend n classes étiquetées*[1] *$\mathcal{X}_1$, $\mathcal{X}_2, \ldots, \mathcal{X}_n$ et un ordre partiel $\leq_\alpha$ sur $\{1, \ldots, n\}$, et retourne le sous-ensemble du produit de ces classes, dans lequel pour chaque $i, j \in E$ tel que $i \leq_\alpha j$, la plus petite étiquette entre la composante $\mathcal{X}_i$ et $\mathcal{X}_j$ soit dans la composante de $\mathcal{X}_i$.*

Pour que la définition soit cohérente, on va supposer qu'aucune classe n'a d'élément de taille 0 ($\forall i, X_i(0) = 0$).

1. On s'écarte dans ce chapitre des notations habituelles, puisque $\mathcal{X}_i$ ne désigne pas les éléments de taille i d'une éventuelle classe $\mathcal{X}$, mais juste une i-ème classe indépendante des autres. Le choix de cette notation (au lieu de $\mathcal{X}^{(i)}$) a été fait pour éviter de surcharger les formules.

Dans le cas où $\leq_\alpha = 1 \leq 2$, la définition classique du produit avec contrainte de croissance binaire de [29] peut être utilisée :

$$\mathcal{A} = \star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2) = \mathcal{X}_1^{\square} \star \mathcal{X}_2 \qquad \text{et} \qquad A(z) = \int_{t=0}^{z} X_1'(t)X_2(t)dt.$$

A partir du produit avec contrainte de croissance binaire, on peut en déduire comment trouver une spécification pour tout ordre total. On prouve en effet aisément que $\star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)$, dans le cas où $\leq_\alpha = 1 < 2 < 3 \cdots < n$, s'écrit facilement en fonction du produit avec contrainte de croissance binaire :

$$\star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) = \mathcal{X}_1{}^{\square} \star \left(\mathcal{X}_2{}^{\square} \star \left(\mathcal{X}_3{}^{\square} \star (\ldots \mathcal{X}_n)\right)\right).$$

L'autre idée principale est qu'un élément satisfaisant un ordre partiel satisfait au moins un des ordres totaux compatibles. Ainsi on peut écrire l'ensemble des éléments satisfaisant un ordre partiel comme l'union des ensembles d'éléments satisfaisant les ordres totaux compatibles. Donc le produit de la définition 2.3 peut se traduire à l'aide de l'opérateur de croissance contrainte binaire :

Lemme 2.3. $\mathcal{A} = \star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) = \sum_{\leq_A \in \mathcal{LE}(\leq_\alpha)} \star^{\square}_{\leq_A}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)$

Le problème ici est le nombre de termes dans la somme, qui peut très rapidement devenir exponentiel : on cherche une manière plus concise d'exprimer la traduction de cet opérateur sur les séries génératrices. Le théorème suivant montre que le produit partiellement ordonné sera une intégrale, où la seule influence de l'ordre partiel choisi est sur le domaine d'intégration que l'on va utiliser.

Notation : On note $d\mathbf{z} = dz_1 dz_2 \ldots dz_n$.

Théorème 2.1. *Soit $\mathcal{A} = \star^{\square}_{\leq_\alpha}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)$, et z plus petit que le rayon de convergence du produit $X_1'X_2' \ldots X_n'$, alors :*

$$A(z) = \int_{(z_1,\ldots,z_n) \in E_{\leq_\alpha}(z)} X_1'(z_1)X_2'(z_2)X_3'(z_3) \ldots X_n'(z_n) d\mathbf{z}.$$

Démonstration. En utilisant le lemme 2.3, on obtient :

$$A(z) = \sum_{\leq_A \in \mathcal{LE}(\leq_\alpha)} \int_{(z_1,\ldots,z_n) \in E_{\leq_A}(z)} X_1'(z_1)X_2'(z_2)X_3'(z_3) \ldots X_n'(z_n) d\mathbf{z}.$$

On peut ensuite utiliser le lemme 2.2 pour obtenir :

$$A(z) = \int_{\bigcup_{(z_1,\ldots,z_n) \in \leq_A \in \mathcal{LE}(\leq_\alpha)} E_{\leq_A}} X_1'(z_1)X_2'(z_2)X_3'(z_3) \ldots X_n'(z_n) d\mathbf{z},$$

et le lemme 2.1 donne le résultat. □

Ce théorème donne une description précise et concise de la valeur de la fonction génératrice associée à ce produit, mais ne donne pas de réponse précise quand à la manière de calculer l'intégrale, en particulier l'ordre d'intégration qu'il faut utiliser. Le problème sous-jacent (compter le nombre d'extensions linéaires d'un poset) étant intrinsèquement compliqué, on ne va pas pouvoir disposer de méthode générale pour calculer ces intégrales efficacement.

Un choix canonique pour le choix de l'ordre consiste à être cohérent avec le produit avec contrainte de croissance binaire, et faire partir toutes les variables des intégrales de 0 :

$$A(z) = \int_{z_1=0}^{\min(\{z_i | i \leq_\alpha 1\})} \int_{z_2=0}^{\min(\{z_i | i \leq_\alpha 2\})} \cdots \int_{z_n=0}^{\min(\{z_i | i \leq_\alpha n\})} X_1'(z_1)X_2'(z_2)X_3'(z_3) \ldots X_n'(z_n) d\mathbf{z}.$$

Dans le cas où l'ordre partiel est un arbre avec comme racine le nœud le plus petit (ce qui correspond aux arbres croissants du chapitre 1), les `min` se réduisent toujours à une seule variable, ce qui rend l'intégration simple.

De manière générale, quelques techniques permettent de transformer les intégrales de la forme canonique en une forme plus pratique. Les deux techniques les plus utiles sont :

— Changer l'ordre des intégrations, de telle manière que les variables centrales soient calculées en premier. Le prix à payer est que la borne inférieure de ces intégrale ne sera plus 0. Voici un exemple ci-dessous :

$$\int_{z_1=0}^{z}\int_{z_2=0}^{z_1}\int_{z_3=0}^{z_2} X_1(z_1)X_2(z_2)X_3(z_3)d\mathbf{z} = \int_{z_1=0}^{z}\int_{z_3=0}^{z_1} X_1(z_1)X_3(z_3)\int_{z_2=z_3}^{z_1} X_2(z_2)d\mathbf{z} \tag{2.1}$$

— Séparer l'intégration en deux cas, ce qui va se payer en augmentant le nombre d'intégrales, mais possiblement plus faciles à calculer. Cela revient à augmenter le nombre d'intégrales pour diminuer la complexité individuelle du calcul d'une intégrale. Il faut choisir ses variables avec soin, puisque si on le fait pour toutes les variables cela revient à la formule du lemme 2.3. Voici un exemple où l'on a séparé les cas $z_1 > z_2$ et $z_1 < z_2$:

$$\int_{z_1=0}^{z}\int_{z_2=0}^{z} X_1(z_1)X_2(z_2)d\mathbf{z} = \int_{z_1=0}^{z}\int_{z_2=0}^{z_1} X_1(z_1)X_2(z_2) + \int_{z_2=0}^{z}\int_{z_1=0}^{z_2} X_1(z_1)X_2(z_2)d\mathbf{z} \tag{2.2}$$

En définitive, l'utilisation de ce nouvel opérateur demande une adaptation au cas par cas de la mise en forme de l'équation, pour avoir le calcul le plus pratique. On va montrer dans la section suivante un cas non trivial où l'on peut spécifier une classe en utilisant ce produit.

2.3 Spécification des arbres synchronisés

On va se servir du nouvel opérateur défini dans la section précédente pour décrire la spécification des arbres synchronisés. On peut reprendre la remarque faite plus tôt, qui est que l'on peut prendre une représentation des arbres synchronisés où l'on fusionne les deux nœuds qui ont la même étiquette. On peut donc spécifier un arbre synchronisé de la manière suivante : on a tout d'abord un fil croissant, que l'on appellera par la suite le *tronc*, où des forêt croissantes sont greffées à gauche et à droite de chaque nœud, suivi d'un embranchement. De cet embranchement on fait partir deux branches, qui sont des fils croissants avec deux forêts croissantes greffées à chaque nœud, que l'on nommera *branche gauche* et *branche droite*, qui se rejoignent en un unique nœud, d'où partent deux forêts croissantes. On greffe à la séparation des branches gauche et droite trois forêts croissantes (voir Figure 4.3).

On peut résumer cela en un simple système pour décrire la classe des arbres synchronisés $\mathcal{S}$:

$$\begin{aligned}
&\mathcal{S} = \mathcal{T}^{\square} \star \mathcal{S} + \mathcal{P}\\
&\mathcal{P} = \sum_{p,q\in\mathbb{N}} \star^{\square}_{\leq_{p,q}}(\mathcal{R}, \mathcal{X}_1, \ldots, \mathcal{X}_p, \mathcal{Y}_1, \ldots, \mathcal{Y}_q, \mathcal{H})\\
&\mathcal{X}_i = \mathcal{Y}_i = \mathcal{H} = \mathcal{T} = \mathcal{Z}^{\square} \star (\mathcal{F}^2)\\
&\mathcal{R} = \mathcal{Z}^{\square} \star (\mathcal{F}^3)
\end{aligned}$$

Avec les notations suivantes : $\leq_{p,q}$ est l'ordre partiel tel que le plus petit élément correspond à $\mathcal{R}$, chaque $\mathcal{X}_i$ (resp. $\mathcal{Y}_i$) est plus petit que $\mathcal{X}_{i+1}$ (resp. $\mathcal{Y}_{i+1}$), et l'élément maximal correspond à $\mathcal{H}$. Plus

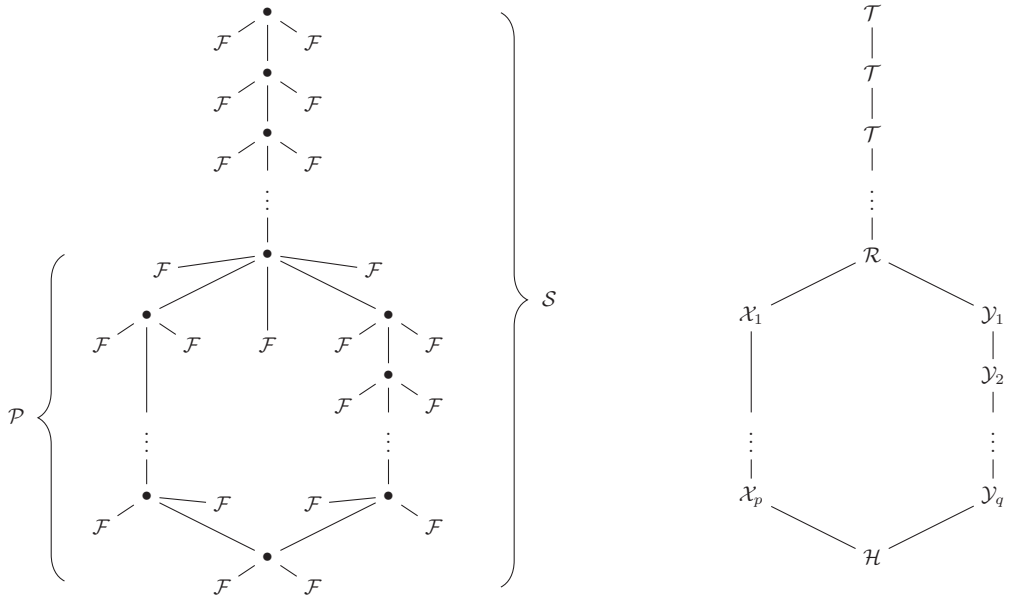


FIGURE 2.2 – Spécification combinatoire des arbres synchronisés

précisément $\leq_{p,q}$ est l'ordre partiel sur $\{1, 2, \ldots, p+q+2\}$ tel que :

$$\begin{aligned}
&1 \leq_{p,q} 2 \text{ et } 1 \leq_{p,q} p+2 \\
&\forall i \text{ tel que } i \leq p, i \leq_{p,q} i+1 \\
&\forall i \text{ tel que } p+2 \leq i \leq p+q+1, i \leq_{p,q} i+1 \\
&p+1 \leq_{p,q} p+q+2 \text{ et } p+q+1 \leq_{p,q} p+q+2
\end{aligned}$$

La classe $\mathcal{F}$ est celle des forêts croissantes telles que $\mathcal{A} = \mathcal{Z}^{\square} \star \mathcal{F}$, où $\mathcal{A}$ est la classe des arbres croissants spécifiés de la manière suivante : $\mathcal{A} = \mathcal{Z}^{\square} \star Seq(\mathcal{A})$. Donc $\mathcal{A}$ vérifie le système :

$$A'(z) = \frac{1}{1 - A(z)} A(0) = 0$$

Donc $A(z) = 1 - \sqrt{1-2z}$, ce qui donne en dérivant la fonction génératrice exponentielle F de $\mathcal{F}$ qui est $F(z) = \frac{1}{\sqrt{1-2z}}$. On en déduit en utilisant le théorème 2.1 pour $\mathcal{P}$

$$\begin{aligned}
S(z) &= \int T'S + P(z) \Rightarrow S'(z) = T'(z)S(z) + P'(z) \\
P(z) &= \sum_{p,q \in \mathbb{N}} \int_{r=0}^{z} \int_{x_1=0}^{r} \ldots \int_{x_p=0}^{x_{p-1}} \int_{y_1=0}^{r} \ldots \int_{y_q=0}^{y_{q-1}} \int_{h=0}^{\min(x_p, y_q)} R'(r) X_1'(x_1) \ldots Y_1'(y_1) H'(h) dr dh dx_1 \ldots, \\
X_i(z) &= Y_i(z) = H(z) = T(z) = \int F(z)^2 = \int \frac{1}{1-2z} \\
&\text{et} \\
R(z) &= \frac{1}{(1-2z)\sqrt{1-2z}}
\end{aligned}$$

En choisissant un bon ordre d'intégration (cf. équation (2.1)),

$$P(z) = \int_{r=0}^{z} \int_{h=0}^{r} R'(r)H'(h) \left(\sum_{p\in\mathbb{N}} B_p(r,h) \right) \left(\sum_{q\in\mathbb{N}} B_q(r,h) \right) drdh$$

$$\text{où} \quad B_p(r,h) = \int_{t_1=h}^{r} \dots \int_{t_p=h}^{t_{p-1}} T'(t_1)T'(t_2)\dots T'(t_p)dt1dt2\dots$$

Les $B_q(r,h)$ vérifient l'équation de récurrence

$$B_{q+1}(r,h) = \int_{t_1=h}^{r} T'(t_1)B_q(t_1,h)$$
$$B_0(r,h) = 1$$

On remarque donc que $B(r,h) = \left(\sum_{q\in\mathbb{N}} B_q(r,h) \right)$ vérifie l'équation différentielle :

$$B(r,h) = \int_{t_1=h}^{r} T'(t_1)B(t_1,h) + 1.$$

Donc en dérivant :

$$\frac{dB(r,h)}{dr} = T'(r)B(r,h).$$

On peut donc obtenir les expression de $S(z)$ et $P(z)$ en résolvant des équations différentielles ordinaires :

$$S(z) = \int_{t=0}^{z} \frac{\sqrt{1-2t}}{\sqrt{1-2z}} P'(t)dt$$

et

$$P(z) = \int_{r=0}^{z} \int_{h=0}^{r} \frac{1}{(1-2r)\sqrt{1-2r}} \frac{1}{1-2h} \left(\frac{\sqrt{1-2h}}{\sqrt{1-2r}} \right)^2 drdh.$$

On peut finalement trouver une formule close pour la série génératrice exponentielle de $\mathcal{S}$:

$$S(z) = \frac{\frac{2z}{1-2\,z} - \log\left(\frac{1}{1-2\,z}\right)}{(4\sqrt{1-2\,z})}.$$

La singularité de $S(z)$ est isolée, $S(z)$ est donc $\Delta-$analytique, ce qui donne, en utilisant un théorème de transfert ([22]) :

Théorème 2.2. *Le nombre d'arbres synchronisés avec n étiquettes, noté S_n, satisfait quand n tend vers ∞ :*

$$S_n \sim 2^n n! \sqrt{\frac{n}{\pi}} \left(\frac{1}{2} - \frac{\log(n)}{4n} \right)$$

On remarque que l'ajout de la synchronisation au modèle n'a pas modifié la nature de la croissance, ni même le facteur exponentiel, qui reste le même que dans le cas sans synchronisation.

2.4 Conclusion

Ce chapitre introduit un nouvel opérateur sur les classes étiquetée, qui permet de faire des produits avec des contraintes d'ordres complexes, qui généralise le produit avec contrainte d'ordre binaire traditionnel. L'opérateur résultant se traduit sur les séries génératrices en une intégrale multiple, où l'ordre désiré se retrouve dans le domaine d'intégration. Ce produit nous permet de spécifier une classe d'objets résultant d'une extension des modèles de concurrence abordés dans le chapitre précédent, puisque maintenant plusieurs actions peuvent s'effectuer simultanément. On le modélise comme des arbres croissants où deux nœuds se partagent une étiquette, en utilisant le produit introduit plus tôt. On obtient ainsi une formule close directement utilisable par les théorèmes classiques de la combinatoire analytique, ce qui nous permet de réaliser une étude asymptotique. Le nombre asymptotique moyen d'exécutions dans ce cas est donc de l'ordre de $2^n n!$, ce qui est une modification d'un facteur exponentiel par rapport aux autres modèles. On peut aussi spécifier des objets plus complexes (par exemple des arbres croissants avec deux partages d'étiquettes), mais la modélisation et les calculs deviennent plus difficiles.

Chapitre 3

Constructions décomposables en réplications

3.1 Introduction

Ce chapitre est né de questionnements sur les constructions. En effet, on dispose d'un cadre général pour spécifier des classes combinatoires, mais on se contente d'un nombre limité de constructions classiques pour cela (cf. [24]). La théorie des espèces permet de décrire des constructions, mais elle ne s'appuie pas directement sur les classes combinatoires, et n'intègrent pas certaines constructions que l'on peut qualifier de naturelles (telle que la construction qui a une classe renvoie la classe composé des doublons d'éléments de cette classe). Ce cheminement m'a amené à définir un cadre simple pour pouvoir spécifier les constructions classiques.

Motivation

On va donc chercher à spécifier des constructions compliquées comme le cycle, et trouver une expression de cette série sous forme de série multivariée. Puisque le cycle vérifie

$$\mathcal{A} = \textsc{Cyc}[\mathcal{B}] \Rightarrow A(z) = \sum_{p\geq 1} \frac{\varphi(p)}{p} \ln\left(\frac{1}{1-B(z^p)}\right),$$

il est naturel de définir des séries multivariées de variables $s_1, s_2, s_3, \dots$ où chaque s_i représente $B(z^i)$. On cherche donc à définir la série $\mathrm{Cyc}(s_1, s_2, \dots)$ pour que $\mathrm{Cyc}(s_1, s_2, \dots) = \sum_{p\geq 1} \frac{\varphi(p)}{p} \ln\left(\frac{1}{1-s_p}\right)$

On va dans un premier temps définir le cadre permettant de spécifier les constructions classiques, puis dans un deuxième temps on va introduire des séries formelles, similaires aux fonctions génératrices, qui nous permettrons de traduire la spécification d'une construction en une équation sur des séries formelles. On va notamment retrouver par ce moyen les formules des constructions classiques.

3.2 Constructions

L'objectif de ce chapitre est d'avoir un moyen de fabriquer de nouvelles constructions à partir de constructions de base, et de leur associer une série formelle à l'image de la fonction génératrice pour les classes, que l'on appelle *série indicatrice de réplication*. On se restreindra à un certain type de constructions, les constructions *réplicables*, qui couvre les constructions classiques dont les constructions de Pólya (cycles, ensembles,...). La série indicatrice de réplication d'une construction est cohérente avec sa série indicatrice de cycle de la théorie des espèces [3], dans le cas où cette construction peut être interprétée en tant qu'espèce. Les constructions réplicables forment un ensemble de constructions plus vastes que les espèces, avec des constructions qui sont réplicables mais non représentables en tant qu'espèce (non virtuelle) dans la théories des espèces.

3.2.1 Définitions

On va porter un grand soin pour chaque construction à définir précisément l'ensemble de classes sur lequel elle est définie. On note $\mathbf{C}$ l'ensemble constitué de toutes les classes combinatoires, et $\mathbf{C}^*$ l'ensemble de toutes les classes n'ayant aucun objet de taille 0.

Définition 3.1. *Une* construction Ψ *est une fonction d'un ensemble de classes* $\mathbf{E} \subset \mathbf{C}$ *vers l'ensemble des classes. On appelle* domaine de Ψ *l'ensemble de définition* $\mathbf{E}$ *de* Ψ*. On appelle* domaine image *de* Ψ *l'ensemble des classes* $\Psi(\mathbf{E})$ *qui sont l'image par* Ψ *d'une classe dans* $\mathbf{E}$*, le domaine de* Ψ*.*

Définition 3.2. *Une construction est dite* totale *si son domaine est* $\mathbf{C}$*. Une construction est dite* infinie *si son domaine est* $\mathbf{C}^*$*.*

Toutes les constructions classiques sont soit totales (comme le produit cartésien d'une classe avec elle-même) soit infinies (comme la séquence ou les constructions de Pólya). Les inverses compositionnels des constructions classiques ne sont par contre en général ni des constructions totales, ni des constructions infinies, mais possèdent un domaine qui reste suffisant pour les caractériser si l'on ajoute des hypothèses de régularité, tel que celui que nous allons présenter ci-dessous.

On va définir la notion de *domaine premier*, qui est un domaine suffisamment vaste pour être significatif. Pour une classe $\mathcal{C}$, on note [1] $[n]\mathcal{C} = \#(\mathcal{C}_n)$ le nombre d'éléments de $\mathcal{C}$ ayant pour taille n.

Définition 3.3. *Un ensemble de classe* $\mathbf{E} \subset \mathbf{C}$ *est un* domaine premier *si il existe un ensemble infini de nombres premiers* $\mathbf{P} = \{p_1, p_2, p_3, \ldots\}$ *tel que :*
Pour tout n*,* $\exists \mathbf{A} \subset \mathbf{E}$ *tel que :*

- $\mathbf{A}$ *est infini*
- $\forall \mathcal{C}, \mathcal{C}' \in \mathbf{A}$ *tel que* $\mathcal{C} \neq \mathcal{C}'$*, alors* $\forall p \in \{p_1, \ldots, p_n\}, [p]\mathcal{C} \neq [p]\mathcal{C}'$

Définition 3.4. *Une construction est dite* significative *si son domaine est un domaine premier.*

Une conséquence est que l'on peut évaluer une construction significative sur une infinité de classes où le nombre d'objets de taille $p_1, p_2, \ldots, p_{n-1}$ et p_n est distinct entre les différentes classes. Cette définition est avant tout technique, et est introduite pour pouvoir prouver le théorème 3.1.

3.2.2 Constructions de base

On va maintenant présenter les constructions de base, qui seront les briques de base pour fabriquer des constructions.

Classes élémentaires Les premières constructions de base sont les constructions qui correspondent aux classes élémentaires. Ces constructions ignorent leur entrée et renvoient respectivement la classe vide $\emptyset$, la classe neutre $\mathcal{E}$ et la classe atomique $\mathcal{Z}$. On fera un abus de notation et notera les constructions comme les classes, c'est-à-dire que pour toute classe $\mathcal{C}$, on va écrire $\emptyset[\mathcal{C}] = \emptyset$, $\mathcal{E}[\mathcal{C}] = \mathcal{E}$ et $\mathcal{Z}[\mathcal{C}] = \mathcal{Z}$. Ces constructions sont des constructions totales.

Réplications On va rajouter aux précédentes constructions d'autres constructions de base, les opérations de réplication. Pour un i entier positif, on note $\mathcal{S}_i$ la construction qui à une classe $\mathcal{C}$ renvoie la classe

$$\mathcal{A} = \mathcal{S}_i[\mathcal{C}],$$

constituée de i copies de la classe $\mathcal{C}$, ou de manière équivalente, une copie de la classe $\mathcal{C}$ où tout toutes les classes atomiques la constituant sont remplacées par un produit de i classes atomiques. La traduction sur les séries génératrices est simple : la copie en i exemplaires de la classe $\mathcal{C}$ a pour série génératrice $S_i[\mathcal{C}](z) = C(z^i)$. La construction $\mathcal{S}_1$ correspond donc à l'identité sur les classes. Les réplications sont des constructions totales.

Exemple 3.1. *Soit* $\mathcal{C}$ *la classe des mots sur un alphabet* Σ *fixé. Alors* $\mathcal{S}_2[\mathcal{C}]$ *est la classe des palindromes de taille paire de mots de* Σ^**.*

1. Cette notation est utile dans le cas où la classe s'écrit comme l'image d'une classe par une construction, et où la notation habituelle C_n est moins pratique.

3.3 Opérations sur les constructions

On regarde les opérations sur les constructions, c'est-à-dire les fonctions qui transforment des constructions.

3.3.1 Opérations basiques sur les constructions

On définit des opérations basiques sur les constructions, en étendant la somme disjointe, le produit cartésien et la composition comme opérateurs sur les constructions. On va aussi ajouter le pointage de construction et l'inverse compositionnel.

Union disjointe et produit cartésien

Soit deux constructions Ψ et Φ, alors $\Psi + \Phi$ et $\Psi \times \Phi$ sont définies respectivement comme la construction qui à la classe $\mathcal{C}$ associe la classe $\Psi[\mathcal{C}] + \Phi[\mathcal{C}]$ et $\Psi[\mathcal{C}] \times \Phi[\mathcal{C}]$. On définit alors simplement la multiplication et la division d'une construction par un entier positif, et l'exponentiation par un entier positif.
Le domaine de définition de l'union disjointe et le produit cartésien de deux construction est clairement égal à l'intersection des domaines des deux constructions.

Exemple 3.2. *Arbres d'Otter*

Un arbre d'Otter est un arbre binaire non-plan. Un manière simple de le spécifier est de dire qu'il s'agit d'un arbre binaire classique, où l'on divise par deux le nombre de choix possibles sur les racines des sous-arbres (puisque l'ordre n'a plus d'importance). Il faut néanmoins rajouter un terme pour compenser les arbres qui ont deux sous-arbres identiques. Cela donne la spécification :

$$\mathcal{O} = \left(\mathcal{E} + \frac{1}{2}\mathcal{Z} \times \mathcal{S}_1^2 + \frac{1}{2}\mathcal{Z} \times \mathcal{S}_2\right)[\mathcal{O}].$$

Qui correspond à la spécification classique :

$$\mathcal{O} = \mathcal{E} + \mathrm{MSet}_2[\mathcal{O}].$$

On peut donc définir la classe $\mathcal{O}$ comme le point fixe de la construction $\mathcal{E} + \frac{1}{2}\mathcal{Z} \times \left(\mathcal{S}_1^2 + \mathcal{S}_2\right)$.

Composition

On veut maintenant définir l'opération de composition de deux opérateurs. Pour des constructions Ψ et Θ, le composé $\Xi = \Psi \circ \Theta$ est le résultat de la composition classique de deux fonctions. Comme pour toute composition de fonctions, elle n'est définie que si l'ensemble d'arrivée de Θ est compatible avec l'ensemble de départ de Ψ. On considérera ici que $\Xi = \Psi \circ \Theta$ est toujours définie, mais son domaine est réduit aux classes où la composition est compatible, c'est-à-dire aux classes $\mathcal{C}$ telles que $\Theta[\mathcal{C}]$ appartient au domaine de Ψ. On peut donc se retrouver avec des constructions triviales, puisque par exemple $\mathrm{Seq} \circ \mathcal{E}$ est définie, mais son domaine est l'ensemble vide.

Il est important de noter que les réplications élémentaires ont une relation particulière avec la composition, puisqu'elles commutent avec n'importe quelle autre construction :

$$\mathcal{S}_i \circ \Psi = \Psi \circ \mathcal{S}_i.$$

Cela vient du fait qu'il est équivalent pour la taille de répliquer une structure ou de répliquer chaque élément de la structure. De plus la composition entre réplications élémentaires est simple, puisque $\mathcal{S}_i \circ \mathcal{S}_j = \mathcal{S}_{i \cdot j}$.
On remarque aussi que de manière générale la composition n'est pas distributive à gauche vis-à-vis de l'addition, c'est-à-dire qu'il existe une construction Ψ telle que

$$\Psi + \Psi \circ \Psi \neq \Psi \circ (\mathcal{S}_1 + \Psi)$$

Pointage de construction

On souhaite ici étendre la notion de pointage qui existe sur les classes aux constructions. Une façon de le faire consiste à introduire des constructions qui prennent plusieurs classes en entrée, ce qui est fait dans [11] en introduisant la notion de pointage de cycle. On va ici relâcher la contrainte que la notion de pointage se comporte bien vis-à-vis de la composition, afin de pouvoir obtenir qu'une construction pointée reste une construction monovariée.

On fait le choix que l'opération de pointage de construction ∇ vérifie $\nabla(\Phi)[\mathcal{Z}] = \Theta(\Phi[\mathcal{Z}])$, où Θ est l'opération de pointage de classe définie dans les préliminaires au paragraphe 0.2.2. Ainsi le pointage de construction est l'opération qui prend une construction en entrée, et renvoie une construction où l'une des classes d'entrée a une marque, les instances dupliquées étant considérées comme distinctes vis-à-vis de la marque. On définit l'opération de pointage ∇ sur les constructions, et opérations basiques comme suit :

$$\begin{aligned}
\nabla\mathcal{E} &= \emptyset \\
\nabla\mathcal{Z} &= \mathcal{E} \\
\nabla\mathcal{S}_i &= i\mathcal{S}_i \\
\nabla(\Psi + \Phi) &= (\nabla\Psi) + (\nabla\Phi) \\
\nabla(\Psi \times \Phi) &= (\nabla\Psi \times \Phi) \cup (\Psi \times \nabla\Phi)
\end{aligned}$$

On ne possède en revanche pas de formule classique du type $(g \circ f)' = f' \times (g' \circ f)$ qui lie le pointage et la composition. Il serait possible de définir un pointage qui s'adapte à la composition [11], mais nécessite de faire l'usage de constructions prenant plusieurs classes en entrée, ce qui sort du cadre de ce que l'on souhaite définir ici.

Exemple 3.3. *Prenons la construction $\Psi = \mathcal{S}_1 \times \mathcal{S}_1 \times \mathcal{S}_2$. Alors*

$$\begin{aligned}
\nabla\Psi &= \nabla(\mathcal{S}_1) \times \mathcal{S}_1 \times \mathcal{S}_2 + \mathcal{S}_1 \times \nabla(\mathcal{S}_1) \times \mathcal{S}_2 + \mathcal{S}_1 \times \mathcal{S}_1 \times \nabla(\mathcal{S}_2) \\
&= \mathcal{S}_1 \times \mathcal{S}_1 \times \mathcal{S}_2 + \mathcal{S}_1 \times \mathcal{S}_1 \times \mathcal{S}_2 + \mathcal{S}_1 \times \mathcal{S}_1 \times (\mathcal{S}_2 + \mathcal{S}_2) \\
&= 4(\mathcal{S}_1 \times \mathcal{S}_1 \times \mathcal{S}_2)
\end{aligned}$$

La construction $\nabla\Psi$ permet donc de marquer un des 4 objets en entrée, les deux objets identiques étant différenciés.

Inversion compositionnelle

On souhaite être capable de pouvoir exprimer la solution à des équations implicites du type :

$$\Phi = \Psi \circ \mathcal{X},$$

où Φ et Ψ sont deux constructions connues, et $\mathcal{X}$ une construction inconnue. On va donc définir l'inverse d'une construction Ψ comme la seule construction $\Psi^{[-1]}$ telle que :

— Son domaine est égal à celui de l'image du domaine de Ψ par Ψ

— Pour toute classe $\mathcal{C}$ dans le domaine de $\Psi^{[-1]}$:

$$(\Psi \circ \Psi^{[-1]})[\mathcal{C}] = \mathcal{C}.$$

— Pour toute classe $\mathcal{C}$ dans le domaine de Ψ :

$$(\Psi^{[-1]} \circ \Psi)[\mathcal{C}] = \mathcal{C}.$$

Notation : On notera dans la suite les exposants d'inverses compositionnels avec des crochets ($\Psi^{[-1]}$), les inverses multiplicatifs sans crochets (Ψ^{-1}), et les inverses de Dirichlet (sur des fonctions entières) avec des accolades ($f^{\{-1\}}$).

3.3.2 Opérations avancées sur les constructions

Séquence compositionnelle

Par analogie avec la séquence classique, on va définir la séquence compositionnelle d'une construction Ψ, où l'on remplace le produit cartésien par la composition :

$$\textsc{Seq}_\circ(\Psi) = \mathcal{S}_1 + \Psi + \Psi \circ \Psi + \Psi \circ \Psi \circ \Psi + \dots$$

Il est à noter que cette séquence n'est pas définie pour toutes les constructions, notamment pour $\mathcal{S}_1$, et ne possède de manière générale la propriété de récurrence classique des séquences, c'est-à-dire qu'il existe Ψ tel que :

$$\textsc{Seq}_\circ(\Psi) \neq \mathcal{S}_1 + \Psi \circ \textsc{Seq}_\circ(\Psi).$$

En revanche pour les réplications élémentaires (hors $\mathcal{S}_1$) cette séquence est naturelle, puisqu'elle s'écrit simplement sous la forme :

$$\textsc{Seq}_\circ(\mathcal{S}_i) = \sum_{n\in\mathbb{N}} \mathcal{S}_{i^n},$$

et que la propriété suivante est vraie :

$$\textsc{Seq}_\circ(\mathcal{S}_i) = \mathcal{S}_1 + \mathcal{S}_i \circ \textsc{Seq}_\circ(\mathcal{S}_i).$$

Les séquences compositionnelles de réplications (hors $\mathcal{S}_1$) sont des constructions infinies.

Inversion de pointage

On veut définir l'opérateur $\nabla^{[-1]}$ comme l'opérateur inverse de ∇. Puisque le pointage est un analogue de la dérivation, il faut pour intégrer faire un choix de constante. On va choisir de décider que les constructions issues de $\nabla^{[-1]}$ n'engendrent aucune construction constante $\mathcal{E}$.

3.4 Constructions classiques

3.4.1 Séquence

La séquence $\textsc{Seq}$ est une construction infinie classique, qui vérifie :

$$\textsc{Seq} = \mathcal{E} + \mathcal{S}_1 \times \textsc{Seq}$$

On s'intéressera aussi à la séquence non vide, $\textsc{Seq}_{\geq 1} = \mathcal{S}_1 \times \textsc{Seq}$, qui est aussi une construction infinie.

3.4.2 Puissance

On va définir une autre notion de séquence, qui va utiliser les toutes les réplications. La construction puissance $\hat{\mathcal{P}}$, qui prend en entrée une classe et qui donne en sortie la classe de la somme disjointe de toutes les copies possibles depuis cette classe initiale. On a donc

$$\hat{P} = \sum_{n\in\mathbb{N}} \mathcal{S}_n.$$

C'est une construction infinie. On remarque que $\hat{P}[\mathcal{Z}] = \mathrm{Seq}[\mathcal{Z}]$, bien que $\hat{P} \neq \mathrm{Seq}$. C'est une construction qui ne peut être représentée en tant qu'espèce dans la théorie des espèces.

Exemple 3.4. *Puissance d'un mot*

Soit la classe $\mathcal{C}$ composée de deux éléments de taille 1 : le mot a et le mot b. Alors $\hat{P}[C]$ est la classe, telle que pour tout $n \in \mathbb{N}$ donné, elle est composée uniquement de deux éléments de taille n : a^n et b^n.

3.4.3 Factorisation

On peut prendre l'inverse de la construction puissance, ce qui permet d'avoir une construction qui "factorise". Le domaine de $\hat{P}^{[-1]}$ est par définition l'ensemble des classes $\mathcal{B}$ telles qu'il existe une suite $(c_i)_{i\in\mathbb{N}}$ telle que $\forall n \in \mathbb{N}^*, [n]\mathcal{B} = \sum_{i|n} c_i$.

Lemme 3.1. *Le domaine de $\hat{P}^{[-1]}$ est premier.*

Démonstration. Soit $k \in \mathbb{N}^*$, on considère la classe $\mathcal{C}^{(k)}$ indexée par k qui possède exactement k objets de taille j pour tout entier j non nul, c'est-à-dire que $\forall j \in \mathbb{N}^*, [j]\mathcal{C}^{(k)} = k$, et $[0]\mathcal{C}^{(k)} = 0$. On note $\mathbf{K}$ l'ensemble constitué de toutes ces classes :

$$\mathbf{K} = \{\mathcal{C}^{(k)}, k \in \mathbb{N}^*\} \tag{3.1}$$

On va montrer que $\hat{P}(\mathbf{K})$, le domaine image de $\mathbf{K}$ par $\hat{P}$, est un domaine premier. On remarque que si p est premier, alors $[p]\hat{P}[\mathcal{C}] = c_1 + c_P$. Donc si l'on choisit deux classes $\mathcal{C} \neq \mathcal{C}'$ dans $\mathbf{K}$, on a $\mathcal{C} = \mathcal{C}^{(k)}$ et $\mathcal{C}' = \mathcal{C}^{(k')}$, avec $k \neq k'$. Soit p un nombre premier, alors $[p]\hat{P}[\mathcal{C}] = 2k$ et $[p]\hat{P}[\mathcal{C}'] = 2k'$, donc $[p]\hat{P}[\mathcal{C}] \neq [p]\hat{P}[\mathcal{C}']$. □

La factorisation est donc une construction significative.

3.4.4 Inverse de séquence compositionnelle de réplication

Dans la cas où l'on fait la séquence compositionnelle d'une réplication élémentaire, la propriété :

$$\textsc{Seq}_\circ(\mathcal{S}_i) = \mathcal{S}_1 + \mathcal{S}_i \circ \textsc{Seq}_\circ(\mathcal{S}_i)$$

peut se réécrire en composant par $(\textsc{Seq}_\circ(\mathcal{S}_i))^{[-1]}$ à droite :

$$\mathcal{S}_1 = (\textsc{Seq}_\circ(\mathcal{S}_i))^{[-1]} + \mathcal{S}_i.$$

On reconnaît ici un formule du même type que l'inversion multiplicative d'une séquence, dont la série génératrice s'écrit $Seq(z)^{-1} = (1 - z)$.

On prouve que la construction $\textsc{Seq}_\circ(\mathcal{S}_i)^{[-1]}$ est significative en suivant le même schéma de preuve que celui du lemme 3.1, c'est-à-dire en prouvant que le domaine image de $\mathbf{K}$ (défini à l'équation (3.1)) par $\textsc{Seq}_\circ(\mathcal{S}_i)$ est premier.

3.4.5 Cycle

On va montrer comment construire les constructions classiques de Pólya en utilisant les constructions et opérateurs vu précédemment, en commençant par l'opérateur de cycle. On va ici réutiliser le schéma de la preuve du cycle de Phillipe Flajolet et Michèle Soria [25].

Séquence primitive

On commence donc par construire l'opérateur de séquence primitive SeqP, qui a une classe renvoie la classe des séquence non-nulles d'éléments de cette classe, telle que cette séquence ne soit pas une répétition. Toutes les séquences non-nulles $\textsc{Seq}_{\geq 1} = \mathcal{S}_1 \times \textsc{Seq}$ peuvent de manière unique se voir comme une répétition d'une séquence primitive, ainsi :

$$\hat{\mathcal{P}} \circ \textsc{SeqP} = \textsc{Seq}_{\geq 1}. \tag{3.2}$$

Donc on peut exprimer SeqP en fonction de $\textsc{Seq}_{\geq 1}$ et l'inverse de $\hat{\mathcal{P}}$:

$$\textsc{SeqP} = \hat{P}^{[-1]} \circ \textsc{Seq}_{\geq 1}.$$

La construction Séquence primitive SeqP est infinie, d'après la bijection exprimée par l'équation (3.2).

Cycle non-périodique

Un cycle non-périodique PCyc est un cycle sans répétition. Une séquence est un cycle où l'on a pointé un objet, puisque l'on décide d'un premier élément. Ici les objets sont tous primitifs (qui ne s'écrivent pas comme une puissance d'un autre objet), donc le pointage est non-ambigu. On en déduit une relation entre les cycles non-périodiques et les séquences primitives :

$$\nabla(\mathrm{PCyc}) = \mathrm{SeqP}.$$

Ainsi on peut décrire les cycles non-périodiques :

$$\mathrm{PCyc} = \nabla^{[-1]}(\mathrm{SeqP}).$$

La bijection explicitée implique que PCyc est une construction infinie.

Cycle

Enfin on peut voir la construction du cycle comme l'opérateur constitué de répétitions de cycles primitifs :

$$\mathrm{Cyc} = \hat{P} \circ \mathrm{PCyc}$$

On arrive donc à exprimer le cycle en fonction de nos constructions de base, en utilisant des inversions :

$$\mathrm{Cyc} = \hat{P} \circ \nabla^{[-1]}(\hat{P}^{[-1]} \circ \mathrm{Seq}_{\geq 1}).$$

La construction PCyc est infinie, les classes images de PCyc n'ont pas d'objet de taille 0, et $\hat{P}$ est infinie, donc Cyc est une construction infinie.

Il est intéressant de noter que l'on réussit donc à traduire complètement la preuve du cycle de Phillipe Flajolet et Michèle Soria [25] en utilisant des opérations sur des constructions. On verra par la suite comment on en déduit la formule de la construction cycle appliquée aux séries génératrices.

3.4.6 Constructions ensemblistes

Multiensembles

Un multiensemble est un ensemble qui autorise la répétition d'éléments, que l'on peut aussi voir comme une séquence ou l'on enlève l'ordre des éléments. On veut chercher à assembler la construction de multiensemble MSet à partir des opérations déjà existantes.

On utilise la bijection qui lie les séquences et les multiensembles de cycles primitifs (voir [24], p85) : $\mathrm{Seq} = \mathrm{MSet} \circ \mathrm{PCyc}$
On peut voir cette bijection comme un analogue de la propriété que n'importe quel mot peut être écrit de manière unique comme un produit de mots de Lyndon croissants ; elle sert notablement à construire des bases libres d'algèbres de Lie ([41], Ch. 5).
Le problème de cette bijection est qu'elle ne caractérise pas entièrement la construction MSet sur tout son domaine, seulement sur le domaine image de PCyc. On note $\overline{\mathrm{MSet}}$ la construction MSet restreinte au domaine image de PCyc. Alors cette bijection nous permet donc d'exprimer la construction du multiensemble directement en fonction de celle du cycle primitif :

$$\overline{\mathrm{MSet}} = \mathrm{Seq} \circ \mathrm{PCyc}^{[-1]}$$

La construction MSet est infinie, mais la construction $\overline{\mathrm{MSet}}$ n'est que significative. Pour le prouver, on peut montrer que le domaine image de $\mathbf{K}$ (défini à l'équation (3.1)) par PCyc est un domaine premier. Cela découle du fait que le nombre d'éléments de taille n de l'image d'une classe $\mathcal{C}$ par PCyc est croissant en fonction du nombre d'éléments de taille i de $\mathcal{C}$ pour tout i, et strictement croissant pour $i = 1$.

k-Ensembles

On va ici décrire la construction de k-Ensemble, qui correspond aux ensembles avec au maximum $k-1$ occurrences d'un élément donné (avec $k \geq 2$). Le cas $k = 2$ correspond aux ensembles classiques, et le cas $k = \infty$ correspond au multiensemble.

On va exprimer simplement les k-ensembles en fonction des multiensembles. Un multiensemble peut être vu comme une fonction qui à chaque élément de l'ensemble de départ E associe un entier positif, tandis qu'un k-ensemble est une fonction qui à chaque élément de l'ensemble de départ E associe un entier positif strictement inférieur à p. On va construire une bijection entre les multiensembles et les k-ensembles en se servant de l'unicité de l'écriture d'un entier en base k. Soit E' l'ensemble constitué de toutes les copies des éléments de E en k^n exemplaires, pour n entier. On a donc $E' = \textsc{Seq}_\circ(\mathcal{S}_k)[E]$, ce qui se traduit en :

$$k\text{-}\textsc{PSet} \circ \textsc{Seq}_\circ(\mathcal{S}_k) = \textsc{MSet}$$

C'est une bijection semblable à l'identité de Vallée [24, p30].

Exemple 3.5. *Soit $\mathcal{C}$ la classe contenant uniquement 3 mots de taille 1, a, b et c. Prenons le multiensemble $\{a, a, a, a, a, b, b, b, c, c, c, c, c, c\}$ construit à partir de $\mathcal{C}$, contenant 5 fois le mot a, 3 fois le mot b et 6 fois le mot $\mathcal{C}$. La bijection précédente le met donc en relation avec le 3-ensemble $\{a, a, a^3, b^3, c^3, c^3\}$. On retrouve aisément le multiensemble de départ à partir du 3-ensemble, en calculant le nombre total de fois ou chaque lettre est présente.*

On exprime donc la construction $\overline{k\text{-}\textsc{PSet}}$, qui est la construction k-PSet restreinte au domaine image de $\textsc{Seq}_\circ(\mathcal{S}_k)$, en fonction de MSet :

$$\overline{k\text{-}\textsc{PSet}} = \textsc{MSet} \circ \left(\textsc{Seq}_\circ(\mathcal{S}_k)^{[-1]}\right).$$

Les k-Ensembles sont des constructions significatives, puisque $\textsc{Seq}_\circ(\mathcal{S}_k)^{[-1]}$ est défini sur un domaine premier, et que son domaine d'arrivée est inclus dans celui de MSet.

3.4.7 Constructions miroir

On va maintenant regarder des constructions qui traduisent une symétrie centrale : la séquence miroir et le cycle miroir.

Séquence miroir

La séquence miroir USeq est la séquence quotientée par la symétrie centrale. On remarque qu'il y a une bijection entre 2USeq et l'union disjointe de Seq et des séquences égales à leur miroir. Les séquences égales à leurs images miroir s'expriment comme un élément central (éventuellement vide) et une séquence de couples d'éléments identiques : $(\mathcal{S}_1 + \mathcal{E}) \times \textsc{Seq}(\mathcal{S}_2)$. On en déduit donc la spécification de USeq :

$$\textsc{USeq} = \frac{1}{2}\left(\textsc{Seq}(\mathcal{S}_1) + (\mathcal{E} + \mathcal{S}_1) \times \textsc{Seq}(\mathcal{S}_2)\right)$$

La construction USeq est infinie, puisque c'est la somme de constructions infinies.

Cycle miroir

Le cycle miroir USeq est le cycle quotienté par la symétrie centrale. La construction est analogue à celle de la séquence miroir, avec une bijection similaire : il y a une bijection entre 2UCyc et l'union disjointe de Cyc et des cycles égaux à leur miroir.

Les cycles égaux à leur miroir, que l'on va noter CycM, sont un peu plus difficiles à spécifier. Il y a trois différents types de cycles égaux à leurs miroirs :

— les séquences de paires d'éléments identiques, qui sont comptés deux fois : $\frac{1}{2}(\mathcal{E}) \times \textsc{Seq}(\mathcal{S}_2)$,

— les séquences de paires d'éléments identiques, avec un élément central : $\mathcal{S}_1 \times \textsc{Seq}(\mathcal{S}_2)$,

— les séquences de paires d'éléments identiques, avec un élément central et un élément au début, qui sont comptés deux fois, les rôles de premier élément et élément central étant cycliquement identiques : $\frac{1}{2}(\mathcal{S}_1)^2 \times \textsc{Seq}(\mathcal{S}_2)$.

On a alors :

$$\textsc{CycM} = \frac{1}{2}(\mathcal{E} + \mathcal{S}_1)^2 \textsc{Seq}(\mathcal{S}_2)$$

Ce qui donne pour les cycles miroirs :

$$\textsc{UCyc} = \frac{1}{2}\left(\textsc{Cyc} + \frac{1}{2}(\mathcal{E} + \mathcal{S}_1)^2 \textsc{Seq}(\mathcal{S}_2)\right)$$

Le cycle miroir est une construction infinie, puisque somme de constructions infinies.

3.5 Séries indicatrices de réplication

On veut transformer ces constructions en séries formelles de la même manière qu'on manipule des fonctions génératrices lorsque l'on veut étudier des classes. L'extension naturelle des fonctions génératrices aux constructions est de voir nos constructions comme des espèces combinatoires [3], et d'étudier les séries indicatrices de cycles. Néanmoins certaines constructions utilisées ici, en particulier les réplications élémentaires et la constructions puissance, ne se traduisent pas naturellement [2] en espèces combinatoires, et on n'adoptera pas un point de vue de théorie des espèces sur nos constructions. On s'appliquera tout de même à ce que nos séries formelles soient cohérentes avec les séries indicatrices de cycles.

3.5.1 Constructions décomposables en réplications

On ne décrit pas toutes les constructions admissibles possibles, cet ensemble étant trop large et pouvant contenir des constructions trop complexes. On veut décrire les constructions qui utilisent en entier les éléments de la classe prise en entrée.

Définition 3.5. *Soit Ψ une construction admissible. On dit que Ψ est* réplicable *sur un domaine* $\mathbf{D}$ *si il existe une fonction Λ des suites à valeurs dans $\mathbb{Q}$ à support fini telle que pour toute classe $\mathcal{A} \in \mathbf{D}$, si on note $\mathcal{C} = \Psi[\mathcal{A}]$, alors*

$$C(z) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda(b) \prod_i A(z^i)^{b_i},$$

où l'on considère que $0^0 = 1$. Si de plus Λ est unique, on dit que la construction Ψ est uniquement réplicable*, et on définit sa* série indicatrice de réplication*, qui est la série formelle telle que*

$$\psi(s_1, s_2, s_3, \dots) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda(b) \prod_i s_i^{b_i}.$$

On ne précisera pas le domaine si il est égal à celui de la construction. Les constructions n'ont pas automatiquement une décomposition unique, il est nécessaire pour cela d'être défini pour suffisamment de classes, et on va montrer que les domaines premiers conviennent.

Théorème 3.1. *Les constructions significatives réplicables sont uniquement réplicables.*

Démonstration. La preuve sera constructive, et on va décrire une méthode pour retrouver explicitement tous les coefficients de la décomposition. Soit Ψ une construction réplicable, et $\mathcal{C}$ une classe de suite énumérative $(c_n)_{n\in\mathbb{N}}$. Pour toute taille n, on peut regarder le nombre d'éléments de taille n dans $\Psi[\mathcal{C}]$, qui est noté $[n]\Psi[\mathcal{C}]$. Puisque Ψ est réplicable, alors $[n]\Psi[\mathcal{C}]$ est un polynôme multivarié, de variables $c_1, c_2, \dots, c_n$.
Par exemple, si Ψ est une construction réplicable, il existe des coefficients a, b et d de $\mathbb{Q}$ tels que $[2]\Psi[\mathcal{C}] = ac_1^2 + bc_1 + dc_2$, c'est-à-dire qu'un objet de taille 2 de $\Psi[\mathcal{C}]$ est soit un produit de deux

2. On peut en effet les définir avec des espèces virtuelles [3, Chapitre 2.5], mais ce n'est pas vraiment *naturel*.

objets de taille 1 de $\mathcal{C}$ (coefficient a), soit un objet de taille 2 de $\mathcal{C}$ (coefficient b), soit un objet de taille 1 de $\mathcal{C}$ dupliqué une fois (coefficient d).

On veut relier les coefficient d'un multipolynôme aux valeurs de Λ. Par exemple pour les objets de taille 4 le monôme C_1^2 correspond au produit $A(z^1)A(z^3)$ et $A(z^2)A(z^2)$, donc le coefficient devant le monôme est égal à $\Lambda(1,0,1,0,0,\dots) + \Lambda(0,2,0,0,0,\dots)$.

On va montrer que l'on sait trouver toutes valeurs de Λ associée à des suites à deux termes non-nuls, puis nous allons montrer comment l'étendre à n'importe quel nombre de termes non-nuls.

Supposons qu'on veuille trouver la valeur de $\Lambda(b)$, où b est la suite telle que $b(i) = b(j) = 1$ et $b(k) = 0$ si $k \notin \{i,j\}$. Alors on cherche comment repérer un monôme dont le coefficient correspondrait directement à celui de $s_i s_j$, et on cherche pour cela une taille finale n et un monôme $c_{p_1}c_{p_2}$ de $[n]\Psi[\mathcal{C}]$ tel que seule la construction $s_i s_j$ contribue au coefficients de $c_{p_1}c_{p_2}$. On a par construction que $n = ip_1 + jp_2$, et on cherche à montrer l'unicité de (i,j), c'est-à-dire qu'il n'existe pas de couple $(i',j') \neq (i,j)$ tel que $n = i'p_1 + j'p_2$.

On peut réécrire cela en disant que l'on veut trouver un n tel que $\exists p_1, p_2 \in \mathbb{N}$ tel que

$$n = ip_1 + jp_2, \text{ et } \forall i', j' \in \mathbb{N}, n = i'p_1 + j'p_2 \Rightarrow (i,j) = (i',j'). \tag{3.3}$$

On va choisir p_1 et p_2 premiers tels que $p_2 \geq \max(i,j)$ et $p_1 > jp_2$. On va montrer que cette définition de p_1 et p_2 satisfait la propriété 3.3.

Soit i', j' tel que $n = i'p_1 + j'p_2 = ip_1 + jp_2$. On note

$$\delta_1 = i' - i \text{ et } \delta_2 = j' - j.$$

On remarque alors que

— $\delta_1 \leq 0$, car si $\delta_1 \geq 1$ on a alors $i'p_1 \geq ip_1 + p_1 > n$, or le terme $j'p_2$ est positif, ce qui contredit l'égalité $n = i'p_1 + j'p_2$.

— $\delta_1 p_1 + \delta_2 p_2 = 0$, or p_1 et p_2 sont premiers et plus grands que i et j, ce qui implique que p_2 divise δ_1, ainsi $\delta_1 = kp_2$, avec $k \in \mathbb{Z}$.

Or si $k < 0$, on a alors $\delta_1 \leq -p_2$, ce qui veut dire que $i' \leq 0$, ce qui contredit son appartenance à $\mathbb{N}$. On a donc $i = i'$ et $j = j'$.

On étend à 3 termes pour i, j, l, en prenant p_1, p_2, p_3, p_{23} premiers tel que $p_1 > p_{23}(jp_2 + lp_3)$, et $p_2 > lp_3$, et on prend $n = ip_1 + p_{23}(jp_2 + lp_3)$. On réalise le même schéma de preuve pour montrer que $i = i'$, puis on l'utilise une deuxième fois pour prouver que $j = j'$.

Il reste ensuite le problème de connaître les coefficients des multipolynômes $[n]\Psi[\mathcal{C}]$. On peut effectuer des évaluations du polynôme pour tous les éléments d'un domaine premier, ce qui veut dire que l'on peut avoir suffisamment de points pour interpoler le coefficient devant $c_{p_1}c_{p_2}\dots c_{p_n}$, pour les nombres premiers $p_1, p_2, \dots, p_n$ du domaine premier. □

3.5.2 Constructions de base

Constructions élémentaires

La construction $\mathcal{E}$ est clairement uniquement réplicable, de série indicatrice de réplication égale à 1.

La construction $\mathcal{Z}$ est non réplicable, puisque la série indicatrice de réplication de $\mathcal{Z}$ n'est pas définie. On l'a volontairement écartée du modèle pour pouvoir prouver le théorème 3.1, mais la rajouter au modèle est une piste pour des développements futurs.

Réplication

La construction $\mathcal{S}_i$ est uniquement réplicable, de série indicatrice de réplication s_i.

3.6 Séries indicatrices de réplication sur les opérations

On va ici montrer comment les spécifications des constructions nous donnent des équations sur les séries indicatrices de réplication associées, c'est à dire comment réaliser une méthode symbolique sur les constructions.

Pour toute cette partie on va considérer les constructions uniquement réplicables Ψ et Φ, telles que :

— il existe Λ_Ψ telle que pour toute classe $\mathcal{A}$ du domaine de Ψ, si on note $\mathcal{C}_\Psi = \Psi[\mathcal{A}]$, alors

$$C_\Psi(z) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\Psi(b) \prod_i A(z^i)^{b_i},$$

— il existe Λ_Φ telle que pour toute classe $\mathcal{A}$ du domaine de Φ, si on note $\mathcal{C}_\Phi = \Phi[\mathcal{A}]$, alors

$$C_\Phi(z) = \sum_{(d_i)_{i\in\mathbb{N}}} \Lambda_\Phi(d) \prod_i A(z^i)^{d_i},$$

3.6.1 Opérations basiques sur les constructions

Union disjointe

Si l'on note $\mathcal{C}_{\Psi+\Phi} = (\Psi+\Phi)[\mathcal{A}]$ pour toute classe $\mathcal{A}$ du domaine de $\Phi+\Psi$, alors

$$C_{\Psi+\Phi}(z) = C_\Psi(z) + C_\Phi(z) = \sum_{(b_i)_{i\in\mathbb{N}}} (\Lambda_\Phi(b) + \Lambda_\Psi(b)) \prod_i A(z^i)^{b_i}.$$

Donc $\Theta = \Psi + \Phi$ est réplicable et si elle est significative, elle a pour série indicatrice de réplication :

$$\theta(s_1, s_2, s_3, \dots) = \psi(s_1, s_2, s_3, \dots) + \phi(s_1, s_2, s_3, \dots).$$

Produit cartésien

Et si l'on note $\mathcal{C}_{\Psi\times\Phi} = (\Psi\times\Phi)[\mathcal{A}]$ pour toute classe $\mathcal{A}$ du domaine de $\Phi\times\Psi$, alors

$$C_{\Psi\times\Phi}(z) = C_\Psi(z)C_\Phi(z) = \sum_{(b_i)_{i\in\mathbb{N}}} \left(\sum_{s+r=b} \Lambda_\Phi(s)\Lambda_\Psi(r) \right) \prod_i A(z^i)^{b_i},$$

où on définit l'addition sur les suites tel que $(r+s)(n) = r(n) + s(n)$.
Donc $\Theta = \Psi\times\Phi$ est réplicable et si elle est significative, elle a pour série indicatrice de réplication :

$$\theta(s_1, s_2, s_3, \dots) = \psi(s_1, s_2, s_3, \dots)\phi(s_1, s_2, s_3, \dots)$$

Produit cartésien avec nombre infini de constructions

On regarde comment le produit cartésien se comporte lorsque l'on rajoute des constructions. Soit $\Psi^{(j)}$ une famille de constructions. Alors si l'on note pour un n fixé $\mathcal{C}_{\Psi^{(1)}\times\Psi^{(2)}\times\dots\times\Psi^{(n)}} = (\Psi^{(1)}\times\Psi^{(2)}\times\dots\times\Psi^{(n)})[\mathcal{A}]$ pour toute classe $\mathcal{A}$ du domaine de $\Psi^{(1)}\times\Psi^{(2)}\times\dots\times\Psi^{(n)}$, alors

$$\begin{aligned} C_{\Psi^{(1)}\times\Psi^{(2)}\times\dots\times\Psi^{(n)}}(z) &= C_{\Psi^{(1)}}(z)\dots C_{\Psi^{(n)}}(z) \\ &= \sum_{(b_i)_{i\in\mathbb{N}}} \left(\sum_{\sum_{j\le n} s^{(j)}=b} \Lambda_{\Psi^{(1)}}(s^{(1)})\dots\Lambda_{\Psi^{(n)}}(s^{(n)}) \right) \prod_i A(z^i)^{b_i}, \end{aligned}$$

On remarque donc que si $\Psi^{(j)}$ est une famille de constructions telle que :

$$\begin{cases} \forall j, \Lambda_{\Psi^{(j)}}(1,0,0,0,\dots) = 1 \\ \forall j, \Lambda_{\Psi^{(j)}}(s) = 0 \text{ si } \exists p \text{ tel que } 0 < p < j \text{ et } s_p \neq 0 \end{cases} \tag{3.4}$$

alors la construction $\prod_{j\in\mathbb{N}} \Psi^{(j)}$ est bien définie, et réplicable.

Composition

Si l'on note $\mathcal{C}_{\Psi\circ\Phi} = (\Psi \circ \Phi)[\mathcal{A}]$ pour toute classe $\mathcal{A}$ du domaine de $\Psi \circ \Phi$, alors

$$C_{\Psi\circ\Phi}(z) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\Phi(b) \prod_i \left(\sum_{(d_i)_{i\in\mathbb{N}}} \Lambda_\Phi(d) \prod_j A(z^{ij})^{d_j} \right).$$

Donc $\Theta = \Psi \circ \Phi$ est réplicable et si elle est significative, elle a pour série indicatrice de réplication :

$\theta(s_1, s_2, s_3, \dots) = (\psi \circ \phi)(s_1, s_2, s_3, \dots) = \psi(\phi(s_1, s_2, s_3, \dots), \phi(s_2, s_4, s_6, \dots), \phi(s_3, s_6, s_9, \dots), \dots)$.

L'effet sur les séries indicatrices de réplication est la composition plethystique [1], dont l'effet sur les réplications est :

$$s_k \circ s_p = s_{k\cdot p}.$$

Un cas particulier intéressant est le cas où ψ peut s'écrire sous la forme $\psi(s_1, s_2, s_3, \dots) = \sum_i p_i s_i$, et $\theta(s_1, s_2, s_3, \dots) = \sum_i \frac{t_i}{i} g(s_i)$, où p_i et t_i sont deux fonctions arithmétiques, et g une fonction à valeurs dans $\mathbb{R}$.

Alors

$$(\psi \circ \phi)(s_1, s_2, s_3, \dots) = \sum_n \sum_{k|n} p_k \frac{n t_{\frac{n}{k}}}{k} g(s_n) = \sum_n \frac{((t \times id) \star p)(n)}{n} g(s_n),$$

où $\star$ est la convolution de Dirichlet (définie dans les préliminaires dans la section 0.4), t et p sont les fonctions arithmétiques cannoniquement dérivées des coefficients.

On peut faire un parallèle avec la multiplication des séries de Dirichlet, pour qui

$$\frac{1}{k^s} \times \frac{1}{p^s} = \frac{1}{(kp)^s},$$

et dont les produits font appel à des convolutions de Dirichlet.

Pointage

Si l'on note $\mathcal{C}_{\nabla\Psi} = \nabla(\Psi)[\mathcal{A}]$ pour toute classe $\mathcal{A}$ dans le domaine de définition de $\nabla\Psi$:

$$\mathcal{C}_{\nabla\Psi} = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\Phi(b) \sum_i i b_i A(z^i)^{b_i} \prod_{j\neq i} A(z^j)^{b_j},$$

Donc $\nabla\Psi$ est réplicable et si elle est significative, elle a pour série indicatrice de réplication :

$$\nabla(\Psi)(s_1, s_2, s_3, \dots) = \sum_{k\geq 1} \frac{k s_k \partial\Psi(s_1, s_2, s_3, \dots)}{\partial s_k}$$

Inverse

L'inverse d'une construction réplicable n'est pas nécessairement réplicable, par exemple l'inverse compositionel de la séquence ne peut pas s'écrire comme une série indicatrice de réplication.

Supposons que l'on trouve une série $\theta(s_1, s_2, s_3, \dots) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\theta(b) \prod_i s_i^{b_i}$ telle que :

$$\psi \circ \theta = \theta \circ \psi = s_1,$$

où $\circ$ correspond à la composition plethystique définie à la section 3.6.1.

Alors si l'on note $\mathcal{C}_{\Psi^{[-1]}} = \Psi^{[-1]}[\mathcal{A}]$ pour toute classe $\mathcal{A}$ dans le domaine de définition de $\Psi^{[-1]}$, et $\mathcal{F}$ une classe telle que $\mathcal{A} = \Psi[\mathcal{F}]$, alors :

$$\mathcal{C}_{\Psi^{[-1]}}(z) = F(z) = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\phi(b) \prod_i A(z^i)^{b_i}.$$

Donc $\Psi^{[-1]}$ est réplicable, et si elle est significative, de série indicatrice de réplication $\theta(s_1, s_2, \dots)$.

Ainsi l'inverse d'une construction significative est réplicable si l'on arrive à trouver une série qui vérifie les bonnes propriétés. Le théorème suivant va donner une formule automatique dans des cas simples.

Théorème 3.2. *Soit Ψ une construction telle que $\psi = \sum_{k\geq 1} \frac{f(k)}{k} g(s_k)$, où f est une fonction à valeurs dans $\mathbb{N}$, $f(1) \neq 0$, et g une fonction injective. Alors*

$$\theta(s_1, s_2, \dots) = g^{[-1]} \left(\sum_{k\geq 1} \frac{f^{\{-1\}}(k)}{k} s_k \right), \tag{3.5}$$

où $g^{[-1]}$ est l'inverse compositionnel de g, et $f^{\{-1\}}$ l'inverse de Dirichlet de f, vérifie

$$(\theta \circ \psi)(s_1, s_2, \dots) = (\psi \circ \theta)(s_1, s_2, \dots) = s_1.$$

Démonstration. On va poser $\theta = g^{[-1]} \left(\sum_{k\geq 1} \frac{f^{\{-1\}}(k)}{k} s_k \right)$ et vérifier que

$$\psi \circ \theta = \theta \circ \psi = s_1.$$

$$\begin{aligned}
\theta \circ \psi &= g^{[-1]} \left(\sum_{k\geq 1} \frac{f^{\{-1\}}(k)}{k} \sum_{p\geq 1} \frac{f(p)}{p} g(s_{kp}) \right) = g^{[-1]} \left(\sum_{n\geq 1} \sum_{p|n} \frac{f^{\{-1\}}(n/p) f(p)}{n} g(s_n) \right) \\
&= g^{[-1]} \left(\sum_{n\geq 1} \frac{\delta(n)}{n} g(s_n) \right) = g^{[-1]} \left(g(s_1) \right) = s_1
\end{aligned}$$

et

$$\begin{aligned}
\psi \circ \theta &= \sum_{k\geq 1} \frac{f(k)}{k} g \left(g^{[-1]} \left(\sum_{p\geq 1} \frac{f^{\{-1\}}(p)}{p} s_{kp} \right) \right) \qquad = \sum_{n\geq 1} \left(\sum_{p|n} \frac{f(n/p) f^{\{-1\}}(p)}{n} s_n \right) \\
&= \sum_{n\geq 1} \frac{\delta(n)}{n} s_n = s_1
\end{aligned}$$

□

Ce théorème couvre la plupart des constructions classiques.

3.6.2 Opérations avancées sur les constructions

Pointage inverse

Si l'on note $\mathcal{C}_{\nabla^{[-1]}\Psi} = \nabla^{[-1]}(\Psi)[\mathcal{A}]$ pour tout $\mathcal{A}$ dans le domaine de définition de $\nabla^{[-1]}\Psi$:

$$\mathcal{C}_{\nabla^{[-1]}\Psi} = \sum_{(b_i)_{i\in\mathbb{N}}} \Lambda_\Phi(b) \sum_{i\geq 1, b_i > 0} \frac{A(z^i)^{b_i+1}}{ib_i} \prod_{j\neq i} A(z^i)^{b_i},$$

Donc $\nabla^{[-1]}\Psi$ est réplicable et si elle est significative, elle a pour série indicatrice de réplication :

$$\left(\nabla^{[-1]}\psi \right)(s_1, s_2, \dots) = \sum_{k\geq 1} \int_{x_k=0}^{s_k} \frac{\psi_k(s_1, s_2, \dots, s_{k-1}, x_k, s_{k+1}, \dots)}{kx_k} dx_k,$$

où ψ_k représente les termes de ψ qui font apparaître s_k.

Séquence compositionnelle

La construction $\textsc{Seq}_\circ(\mathcal{S}_i)$ est uniquement réplicable, et sa série indicatrice de réplication est

$$(\mathrm{Seq}_\circ(S_i))(s_1, s_2, \dots, s_n, \dots) = \sum_{n\in\mathbb{N}} s_{i^n}.$$

3.7 Séries indicatrices de réplication de constructions classiques

3.7.1 Séquence

La séquence SEQ est une construction réplicable significative classique, de série indicatrice de réplication :

$$\mathrm{Seq}(s_1, s_2, \dots, s_n, \dots) = \frac{1}{1 - s_1}.$$

Les séquences non vides ont pour série indicatrice de réplication :

$$\mathrm{Seq}_{\geq 1}(s_1, s_2, \dots, s_n, \dots) = \frac{s_1}{1 - s_1}.$$

Puissance et factorisation

On a donc

$$\hat{P}(s_1, s_2, \dots) = \sum_{n \in \mathbb{N}} s_n.$$

Dans le cas où $\theta(s_1, s_2, \dots) = \sum_{p \geq 1} \frac{t(p)}{p} g(s_p)$, où g est une fonction à valeurs dans $\mathbb{R}$ et t une fonction arithmétique, on en déduit d'après l'équation 3 (issue des préliminaires) :

$$(\hat{P} \circ \theta)(s_1, s_2, \dots) = \sum_{n} \sum_{k|n} \frac{\frac{n}{k} t_k}{n} g(s_n) = \sum_{n \geq 1} \frac{(t \star id)(n)}{n} g(s_n),$$

Le théorème 3.2 nous permet d'avoir une formule simple pour l'inverse de $\hat{P}$. Ainsi $\hat{P}^{[-1]}$ est réplicable significative, de série indicatrice de réplication :

$$\hat{P}^{[-1]} = \sum_{p \geq 1} \mu(p) s_p,$$

où μ est la fonction de Möbius, qui est l'inverse de Dirichlet de $n \mapsto 1$.

3.7.2 Séquence compositionnelle inverse de réplications

On utilise ici le résultat de la section 3.4.4, qui nous indique que $\mathrm{SEQ}_\circ(s_i)$ est réplicable, significative, et de série indicatrice de réplication

$$(\mathrm{SEQ}_\circ(s_i))^{[-1]} = s_1 - s_i.$$

3.7.3 Cycles

Séquence primitive

On reprend l'expression vue à la section 3.4.5

$$\mathrm{SeqP} = \hat{P}^{[-1]} \circ \mathrm{Seq}_{\geq 1}.$$

On en déduit que la séquence primitive est réplicable et significative, et de série indicatrice de réplication de SeqP :

$$\mathrm{SeqP}(s_1, s_2, \dots) = \left(\sum_{p \geq 1} \mu(p) s_p \right) \circ \left(\frac{s_1}{1 - s_1} \right) = \sum_{p \geq 1} \mu(p) \frac{s_p}{1 - s_p}$$

Cycle non-périodique

On reprend l'expression vue à la section 3.4.5

$$\mathrm{PCyc} = \nabla^{[-1]}(\mathrm{SeqP}).$$

On en déduit que le cycle non-périodique est réplicable et significatif, et de série indicatrice de réplication :

$$\mathrm{PCyc}(s_1, s_2, \dots) = \sum_{p\geq 1} \frac{\mu(p)}{p} \ln\left(\frac{1}{1-s_p}\right)$$

Cycle

On reprend l'expression vue en 3.4.5

$$\mathrm{Cyc} = \hat{P} \circ \nabla^{[-1]}(\hat{P}^{[-1]} \circ \mathrm{Seq}).$$

On en déduit que le cycle est réplicable et significatif, et de série indicatrice de réplication :

$$\mathrm{Cyc}(s_1, s_2, \dots) = \left(\sum_{p\geq 1} s_p\right) \circ \left(\sum_{p\geq 1} \frac{\mu(p)}{p} \ln\left(\frac{1}{1-s_p}\right)\right) = \sum_{p\geq 1} \frac{\varphi(p)}{p} \ln\left(\frac{1}{1-s_p}\right),$$

où φ est l'indicatrice d'Euler, puisque $\varphi = \mu \star id$.

3.7.4 Constructions ensemblistes

Multiensemble

On reprend l'expression vue en 3.4.6

$$\overline{\mathrm{MSet}} = \mathrm{Seq} \circ \mathrm{PCyc}^{[-1]}.$$

On peut calculer la série indicatrice de réplication de $\mathrm{PCyc}^{[-1]}$ en utilisant le Théorème 3.2 :

$$\mathrm{PCyc}^{[-1]}(s_1, s_2, s_3, \dots) = 1 - 1/\exp\left(\sum_{k\geq 1} \frac{(\mu)^{\{-1\}}(k)}{k} s_k\right). \tag{3.6}$$

On en déduit que le multiensemble est réplicable et significatif sur le domaine de $\mathrm{PCyc}^{[-1]}$, et de série indicatrice de réplication :

$$\overline{\mathrm{MSet}}(s_1, s_2, \dots) = \exp\left(\sum_{k\geq 1} \frac{1}{k} s_k\right),$$

puisque par définition $\mu = (n \mapsto 1)^{\{-1\}}$.

On ne peut donc pas savoir à partir de la bijection utilisée si la construction MSet est réplicable, puisqu'elle ne donne aucune indication sur son comportement en dehors du domaine de $\mathrm{PCyc}^{[-1]}$. En revanche si l'on suppose que MSet est réplicable sur tout son domaine de définition, on en déduit que

$$\mathrm{MSet}(s_1, s_2, \dots) = \exp\left(\sum_{k\geq 1} \frac{1}{k} s_k\right).$$

On va montrer que MSet est réplicable sur tout son domaine de définition, en utilisant sa définition :

$$\mathrm{MSet} = \prod_{i\geq 1} \mathrm{Seq}(\mathcal{S}_i).$$

Donc MSet est réplicable, puisque la famille de constructions $\mathrm{Seq}(\mathcal{S}_i)$ vérifie les propriétés le l'équation (3.4).

k-Ensemble

On reprend l'expression vue en 3.4.6

$$\overline{k\text{-PSET}} = \text{MSET} \circ \left(\text{SEQ}_\circ(\mathcal{S}_k)^{[-1]}\right).$$

On en déduit que les k-ensembles sont réplicables et significatifs sur le domaine de $\text{SEQ}_\circ(\mathcal{S}_k)^{[-1]}$, et de série indicatrice de réplication :

$$\overline{\text{k-PSet}}(s_1, s_2, \dots) = \left(\exp\left(\sum_{n\geq 1} \frac{1}{n} s_n\right)\right) \circ (s_1 - s_k) = \exp\left(\sum_{n\geq 1} \frac{\eta_k}{n} s_n\right),$$

où $\eta_k = 1 \star (id \times (\delta_1 - \delta_k))$, soit

$$\eta_k(n) = \begin{cases} -(k-1) & \text{quand } n \equiv 0 \bmod k \\ 1 & \text{sinon} \end{cases}$$

On peut donc faire la même constat que pour les multiensembles, c'est-à-dire que si k-PSET est réplicable sur tout son domaine de définition, alors

$$\text{k-PSet}(s_1, s_2, \dots) = \exp\left(\sum_{n\geq 1} \frac{\eta_k}{n} s_n\right).$$

Il est possible de montrer qu'il sont réplicables en utilisant la même technique que pour les multiensembles.

3.7.5 Constructions miroir

Séquence miroir

On reprend l'expression vue en 3.4.7

$$\text{USEQ} = \frac{1}{2}\left(\text{Seq}(\mathcal{S}_1) + (\mathcal{E} + \mathcal{S}_1)\text{Seq}(\mathcal{S}_2)\right)$$

La séquence miroir est réplicable et significative, et de série indicatrice de réplication :

$$USeq(s_1, s_2, \dots) = \frac{1}{2}\left(\text{Cyc}(s_1, s_2, \dots) + \frac{1 + s_1}{(1 - s_2)}\right)$$

Cycle miroir

On reprend l'expression vue en 3.4.7

$$\text{UCYC} = \frac{1}{2}\left(\text{CYC} + \frac{1}{2}(\mathcal{E} + \mathcal{S}_1)^2\text{Seq}(\mathcal{S}_2)\right)$$

La séquence miroir est réplicable et significative, et de série indicatrice de réplication :

$$UCyc(s_1, s_2, \dots) = \frac{1}{2}\left(Cyc + \frac{(1 + s_1)^2}{2(1 - s_2)}\right)$$

3.8 Conclusion

On a donc défini dans ce chapitre un cadre permettant de spécifier des constructions si elles peuvent être décrites à partir d'opérations simples. On se limite donc à un sous-ensemble des constructions, qui sont les constructions décomposables en réplications. On arrive notamment à interpréter la preuve classique de la construction du cycle en utilisant des opérations sur les constructions, ainsi que les constructions ensemblistes. Le deuxième volet de résultat dans ce chapitre permet

de traduire les spécifications de nos constructions en équations sur leurs séries indicatrices de cycle, ce qui nous permet de retrouver les formules classiques des cycles et des ensembles. Il reste à étendre le cadre des constructions réplicables pour inclure comme construction de base celle correspondant à la classe atomique, afin de pouvoir définir des classes comme des points fixes de constructions réplicables.

Deuxième partie

Génération aléatoire

Chapitre 4

Génération récursive de structures étiquetées avec contraintes

Ce chapitre va s'intéresser uniquement à la génération récursive, qui est la méthode de génération aléatoire la plus naturelle dès lors que l'on sait pleinement énumérer le nombre d'objet d'une classe. Introduite dans [26], cette méthode consiste à utiliser les équations de récurrence que l'on possède sur les classes d'objets concernées, de regarder exhaustivement toutes les possibilités, calculer les probabilités associées, puis de choisir les tailles des différents sous-objets définis par l'équation de récurrence. On se contentera de décrire des techniques de génération uniforme en taille exacte, qui consistent à se fixer une taille n puis à générer uniformément un objet parmi tous les objets de taille n dans la classe. Nous n'aborderons pas dans ce chapitre la notion de génération approchée, puisqu'elle n'offre aucun gain de complexité dans le cadre de la génération récursive.

Le but est de générer uniformément des structures étiquetées avec des distributions d'étiquetages non triviales. On décomposera la génération en deux parties : la génération uniforme de la silhouette (la structure sans les étiquettes), puis l'étiquetage uniforme d'une structure. Les deux parties peuvent être utilisées indépendamment, l'étiquetage pouvant par exemple être utilisé dans le cadre de la théorie de la concurrence, où l'on dispose d'un processus fixé et l'on souhaite tirer une exécution uniformément *sur ce processus* (cf. chapitre 1).

La section 4.1 va présenter des méthodes de génération récursive, la section 4.2 va s'appesantir sur les méthodes utilisables pour générer la silhouette d'un objet étiqueté, la section 4.3 montrera comment étiqueter uniformément une silhouette, et la section 4.4 donne des exemples d'application de ces méthodes.

4.1 Génération d'objets étiquetés : silhouettes et étiquettes

Nous allons discuter dans cette partie de comment générer uniformément des objets étiquetés sur des classes munies d'un ordre partiel que doit respecter les étiquettes de ses objets. Nous allons montrer précisément comment l'appliquer dans le cas des arbres synchronisés (introduits au Chapitre 2).

Cette génération repose sur la génération récursive classique [26], avec un ajout de complexité dû à la spécification moins classique des classes impliquées, et à la nécessité de générer un étiquetage uniforme [1]. On va donc générer les structures étiquetées en deux temps : on construit d'abord la *silhouette* de la structure, c'est à dire la structure où l'on a enlevé les étiquettes, puis on étiquette uniformément la silhouette.

Il faut faire attention lorsque l'on veut générer une silhouette d'un objet étiqueté uniformément,car bien que les silhouettes des objets étiquetés soient des objets non-étiquetés, la distribution pour générer une silhouette uniformément sur les silhouettes d'objets étiquetés n'est *pas* la distribution uniforme sur les objets non-étiquetés (voir Exemple 4.1). Pour éviter toute confusion, on va appeler

1. On doit bien entendu aussi générer un étiquetage uniforme dans le cas classique, mais c'est une permutation uniforme ce qui rend la tâche simple ([26, note p. 12]).

la distribution uniforme des silhouettes d'objets étiquetés *distribution pondérée*, puisque les objets non-étiquetés n'y sont pas uniformément distribués.

Exemple 4.1. *Arbres croissants de taille* 3.

Il y a deux silhouettes d'arbres croissants généraux de taille 3 : une racine avec deux fils, et un fil croissant de taille 3, tandis que le nombre d'arbres croissants différents s'élève à trois (voir Fig 4.1).

Ainsi un tirage uniforme des arbres croissants va tirer la silhouette du fil avec probabilité 1/3, ce qui correspond à la distribution *pondérée* sur les silhouettes, alors que la distribution *uniforme* sur les arbres généraux tire un fil avec probabilité 1/2.

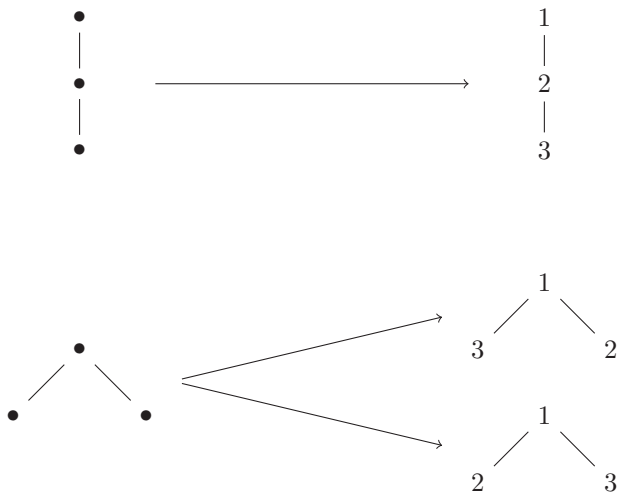


FIGURE 4.1 – Silhouettes des arbres généraux de taille 3 (à gauche) et les arbres croissants correspondants (à droite).

On va donc distinguer la génération en deux parties clairement distinctes, qui n'utiliseront pas les même techniques :

— La génération pondérée des silhouettes, qui fera usage des techniques classiques de génération récursive, avec parfois des choix de tailles fait à l'aide de séries génératrices multivariées.

— La génération uniforme d'étiquetages, qui va utiliser une pondération de la structure obtenue pour décider récursivement de l'étiquetage.

4.1.1 Choix de taille multivarié

On va développer ici la notion de génération récursive multivariée, qui nous est nécessaire pour générer finement certains objets. Nous allons commencer par regarder le cas de la génération bivariée, l'extension à la génération multivariée étant naturelle, puisque qu'on peut rajouter les variables une à une.

Le principe consiste à ajouter à une classe combinatoire un *poids*, qui à chaque objet associe un entier, ce qui permet de décrire plus précisément les objets d'une classe. Par exemple, pour un arbre, on peut s'intéresser à sa hauteur, à sa largeur, au nombre de fils de la racine, etc.

Le but de la génération bivariée consiste à rajouter une fonction de poids à la classe $\mathcal{C}$, puis à découper $\mathcal{C}$ en classes $\mathcal{C}^{(i)}$ pour lesquelles le poids en question est fixé avec pour valeur i. On tire ensuite une classe $\mathcal{C}^{(i)}$ uniformément, c'est-à-dire qu'on choisit le poids i de l'objet à tirer, puis on tire enfin uniformément un objet de $\mathcal{C}^{(i)}$. On a ainsi réduit l'espace des objets que l'on doit tirer uniformément, et on espère ainsi que la classe $\mathcal{C}^{(i)}$ sera plus facile à spécifier que $\mathcal{C}$.

Pour une classe $\mathcal{C}$, sa série génératrice ordinaire bivariée est

$$C(z,u) = \sum_{k,n\in\mathbb{N}} c_{k,n} z^n u^k,$$

où $c_{k,n}$ est le nombre d'objet de taille n et de poids k.

Alors la probabilité qu'un objet γ de taille n ait pour poids k est :

$$\mathbb{P}_n[|\gamma| = k] = \frac{c_{k,n}}{\sum_{i\in\mathbb{N}} c_{i,n}} = \frac{[u^k]C_n(u)}{C_n(1)}, \tag{4.1}$$

où $C_n(u) = [z^n]C(z,u) = \sum_{k\in\mathbb{N}} c_{k,n}u^k$.

Par la suite, pour un polynôme $P(u)$, un appellera *distribution canonique associée* la distribution discrète qui associe à k la probabilité

$$\mathbb{P}[X = k] = \frac{[u^k]P(u)}{P(1)}.$$

Ainsi pour choisir un poids uniformément il suffit de tirer le poids dans la distribution canonique associée au polynôme $[z^n]C(z,u)$.

Dans le cas où nos objets sont étiquetés, on peut là aussi choisir selon la distribution canonique de $[z^n]C(z,u)$, puisque la distribution est la même que celle de $n![z^n]C(z,u)$. En revanche si l'on étiquette séparément par rapport au poids, il faut considérer la distribution canonique de $\sum_{i\in\mathbb{N}} i!u^i[z^nv^i]C(z,v)$. On peut facilement étendre la génération à des cas multivariés quelconques, en choisissant les différents poids successivement.

Exemple 4.2. *Arbres binaires.*

On va montrer comment on peut générer uniformément des arbres binaires de taille n donnée en utilisant des techniques de choix multivarié [2] : on va attribuer un poids à tous les nœuds du sous-arbre gauche comme sur la Fig 4.2. On rappelle que la série génératrice des arbres binaires $C(z) = (1-\sqrt{1-4z})/2z$ est une solution de $C(z) = 1 + zC(z)^2$. On va ici modifier l'équation pour ajouter la variable associée au poids u :

$$C(z,u) = 1 + zC(zu)C(z)$$

Pour générer un arbre de taille n, on va donc choisir un k avec probabilité $\frac{[u^k]C(z,u)}{C(z)}$. On connaît donc la taille du sous-arbre gauche et droit, et on détermine récursivement la taille de tous les sous-arbres.

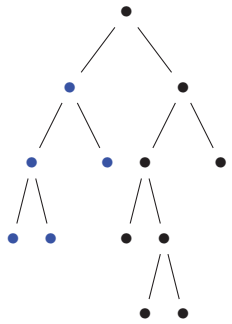


FIGURE 4.2 – Arbre binaire avec des poids (en bleu) sur les nœuds du sous-arbre gauche

4.2 Génération aléatoire de silhouettes

La génération récursive introduite dans [26] génère automatiquement les structures décomposables, c'est-à-dire pouvant être décrites à l'aide d'une grammaire utilisant les opérateurs classiques de la

2. C'est évidemment loin d'être la meilleur méthode pour générer des arbres binaires, et sert ici juste à illustrer la technique.

méthode symbolique. Nous allons montrer comment étendre ces techniques à des constructions additionnelles. On va ici décrire une méthode générique pour générer uniformément un objet tiré d'une classe construite à partir d'une construction admissible[3], définie dans les préliminaires 0.2.2.

Le schéma de la génération sera le suivant :

— on décide pour chaque classe prise en argument de la taille prise par l'objet correspondant,

— on génère indépendamment les objets de chacune de ces classes avec la taille désirée à l'aide de leur générateur (que l'on suppose connu),

— on recombine les différents objets générés.

Le problème de ce schéma est qu'il suppose que chaque classe prise en entrée n'est associée qu'à un seul objet. On souhaite donc distinguer les différentes instances d'une même classe utilisées dans une construction donnée. On veut par exemple distinguer le premier et le deuxième élément du produit $\mathcal{X} \mapsto \mathcal{X} \times \mathcal{X}$, car les tailles des deux instances de la classe $\mathcal{X}$ sont a priori différentes.

On va donc définir la notion de construction linéaire, qui désigne une construction pour laquelle chaque classe impliquée possède au plus une instance dans la construction finale. Ainsi pour engendrer un objet d'une construction non linéaire, on ajoutera une étape préliminaire qui consiste à linéariser les constructeurs utilisés. Dans certains cas cela peut être fait facilement, par exemple $\mathcal{X} \mapsto \mathcal{X} \times \mathcal{X}$ qui se linéarise en $(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}) \mapsto \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$, d'autres demanderont de faire appel à des bits d'aléa.

On va caractériser les constructions qui ne produisent pas différentes instances provenant de la même classe :

Définition 4.1. *On dit que Ψ est* linéaire *si son opérateur $\psi(u_1, u_2, u_3, \ldots, u_k)$ est un polynôme de degré strictement inférieur à 2 en les variables $u_1, u_2, \ldots, u_{k-1}$ et u_k.*

Exemple 4.3. *La construction $\Psi(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}) = \mathcal{X}^{(1)} + \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$ est* linéaire.
En effet $\psi(u_1, u_2) = u_1 + u_1 u_2$ ne contient pas de monôme avec une variable de degré 2 ou plus.

Exemple 4.4. *La construction $\Psi(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}) = \mathcal{X}^{(1)} \times \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$ n'est pas* linéaire.
En effet $\psi(u_1, u_2) = u_1^2 u_2$ contient un terme non nul avec une variable de degré 2 en u_1, et n'est donc pas linéaire.

4.2.1 Génération de constructeurs admissibles

On va dans un premier temps montrer que l'on sait générer des constructions admissibles et linéaires, puis on va montrer comment linéariser une construction admissible non linéaire.

On va supposer par la suite que l'on se donne une construction Ψ admissible et linéaire, des classes $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)}$ que l'on sait générer, et que l'on souhaite générer uniformément un élément de taille n de $\mathcal{C} = \Psi(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)})$.

Puisque Ψ est construction admissible et linéaire, générer un élément de $\mathcal{C}$ uniformément revient à décider d'une répartition des tailles des éléments provenant de $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)}$, d'engendrer aléatoirement et indépendamment des éléments de $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)}$ selon les tailles choisies, puis de recomposer les objets tirés. La recomposition est une étape qui va dépendre de la classe étudiée, et qui peut s'avérer complexe.

On va donc introduire des poids $\mathcal{U}^{(i)}$ déterminant chacun respectivement le poids de la classe $\mathcal{X}^{(i)}$, et étudier $\mathcal{K} = \Psi(\mathcal{U}^{(1)}\mathcal{X}^{(1)}, \mathcal{U}^{(2)}\mathcal{X}^{(2)}, \ldots, \mathcal{U}^{(k)}\mathcal{X}^{(k)})$, dont la fonction génératrice multivariée est $K(z, u_1, u_2, \ldots, u_k) = \psi(X_1(u_1 z), X_2(u_2 z), \ldots, X_k(u_k z))$, où $X_1, X_2, \ldots, X_k$ sont les fonctions génératrices associées à $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)}$. On peut ensuite choisir uniformément une répartition de taille uniforme pour notre tirage, en tirant la taille de p_1 l'instance de la classe $\mathcal{X}^{(1)}$ selon la distribution du polynôme $\frac{[z^n]K(z,u_1,1,1,\ldots)}{[z^n]C(z)}$, puis la taille p_2 l'instance de la classe $\mathcal{X}^{(2)}$ selon la distribution du polynôme $\frac{[u_1^{p_1} z^n]K(z,u_1,u_2,1,\ldots)}{[z^n]C(z)}$, et ainsi de suite jusqu'à u_k.

On génère ensuite indépendamment des éléments de $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(k)}$ selon les tailles que l'on vient de tirer. La dernière partie, la recomposition, consiste en deux parties : recomposition des silhouettes et recomposition des étiquetages. La recomposition des silhouettes dépend entièrement

3. On suppose que l'on sait générer indépendamment les classes de l'ensemble de départ de la construction, à l'aide d'un générateur de notre choix.

de la construction, et sera dans les exemples abordés dans la Section 4.4 triviale. La recomposition des étiquettes va dépendre fortement de Ψ ; on montrera dans la partie 4.4 comment la faire dans certains cas particuliers.

Dans le cas d'opérateurs non linéaires mais dont les fonctions génératrices sont des polynômes, on sépare les instances indépendantes, par exemple en remplaçant $\mathcal{X}^n$ par $\mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \cdots \times \mathcal{X}^{(n)}$. Dans le cas des opérateurs dont les fonctions génératrices sont des séries multivariées, il faut d'abord choisir le nombre de fois que chaque classe va être utilisé, que l'on va choisir selon la distribution canonique de l'équation (4.1).

On génère $\mathcal{C} = \Psi(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(k)})$ de la façon suivante :

— On calcule la série génératrice ordinaire multivariée $C(z, u_1, \ldots, u_k) = \psi(X_1(u_1 z), \ldots, X_k(u_k z))$.

— On choisit les tailles $p_1, p_2, \ldots, p_n$ correspondant aux classes $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(k)}$.

— On génère indépendamment et uniformément des objets de taille p_i pour les classes $\mathcal{X}^{(i)}$.

— On recompose les différents objets générés avec Ψ.

4.3 Génération d'étiquetages

On va donner des méthodes nous permettant d'étiqueter des silhouettes déjà connues.

4.3.1 Étiquetages croissants

La méthode qui va être décrite ici est une application directe de la méthode récursive.

On dispose ici d'une silhouette de taille n et $C(z)$ la fonction génératrice associée[4] , et l'on souhaite lui donner un étiquetage qui respecte les contraintes d'ordre données par la silhouette[5]. On va réaliser l'étiquetage récursivement étiquette par étiquette, en enlevant un nœud à chaque fois. On distingue deux cas :

— Si c'est un arbre croissant, on élève la racine, et on lui donne l'étiquette suivante (on commence à 0 en incrémentant).

— Si c'est une forêt croissante $\mathcal{L}$ de série génératrice $L(z)$, on calcule pour chaque racine r_i de la forêt la série génératrice $L^{(i)}(z)$ correspondant à la forêt $\mathcal{L}$ où l'on remplace l'arbre issu de r_i par la forêt $\mathcal{L}^{(i)}$ constitué des sous-arbres de r_i. On choisit ensuite le nœud r_i avec probabilité $\frac{L^{(i)}(1)}{L(1)}$, qui obtient l'étiquette suivante, puis on le retire de la structure, on le remplace par $\mathcal{L}^{(i)}$ et on étiquette la structure obtenue.

4.3.2 Étiquetages croissants partiels

On cherche ici à tirer un profil à un niveau l, c'est à dire un étiquetage de $l \leq n$ nœuds de la silhouette, uniforme *parmi tous les étiquetages du même niveau*. Ce n'est en général *pas* équivalent à faire un étiquetage croissant et à l'arrêter au bout de l étapes, puisque la distribution sera biaisée par ce qu'il peut arriver sur les niveaux supérieurs. Pour le réaliser, la solution consiste à rajouter une variable u sur $C(z)$, tel que $[u^l z^n]C(z, u)$ compte le nombre d'étiquetages croissants s'arrêtant au niveau l. On peut ensuite utiliser la méthode d'étiquetage de 4.3.1, en décrémentant l à chaque étape. On verra en 4.4.2 comment le faire efficacement dans le cas des arbres croissants.

4.4 Exemples de génération : arbres étiquetés avec contraintes d'ordre

On va ici illustrer les méthodes vues précédemment sur des structures d'arbres étiquetés avec contraintes d'ordres. On présente cela en deux temps : d'abord la génération des silhouettes selon la distribution pondérée, puis celle des étiquettes.

4. Qui ici se résume à un terme en z^n.
5. dans le cas des arbres que nous considérons, c'est que l'étiquette du père est plus petite que celle de ses fils.

4.4.1 Génération aléatoire pondérée de silhouettes

On décrit dans cette section comment utiliser les principes précédemment vus pour générer des silhouettes selon la distribution pondérée.

Arbres croissants

Les arbres croissants sont décomposables, on peut donc utiliser les méthodes de génération récursive de [26]. On rappelle que la spécification des arbres croissants s'écrit :

$$\mathcal{G} = \mathcal{Z}^{\square} \star \textsc{Seq}(\mathcal{G}).$$

On le réécrit pour le décomposer en opérations plus élémentaires, en introduisant les forêts croissants $\mathcal{F}$, qui correspondent aux séquences d'arbres croissants :

$$\begin{cases} \mathcal{G} = \mathcal{Z}^{\square} \star \mathcal{F} \\ \mathcal{F} = \mathcal{G} \times \mathcal{F} + \mathcal{E} \end{cases}$$

On calcule ensuite les coefficients g_k de $\mathcal{G}$ ainsi que les coefficients f_k de $\mathcal{F}$ jusqu'à la taille souhaitée, et on génère récursivement :

— Un arbre croissant de $\mathcal{G}$ de taille n s'écrit comme une racine de taille 1, et une forêt croissante de taille $n-1$.

— Une forêt croissante de taille $n > 0$ est un arbre croissant de taille k, avec comme frères les arbres d'une forêt de taille $n-k$. On tire un arbre de taille k avec probabilité

$$\mathbb{P}[\text{Arbre de taille } k] = \frac{g_k f_{n-k}}{\sum\limits_{p \leq n} g_p f_{n-p}}.$$

On calcule les coefficients en utilisant la solution du système :

$$G(z) = 1 - \sqrt{1-2z}.$$

Arbres synchronisés

On s'intéresse maintenant au cas des arbres synchronisés. La difficulté réside ici dans la présence du produit contraint généralisé (présenté dans la partie 2).

On rappelle la spécification des arbres synchronisés $\mathcal{S}$:

$$\begin{aligned} \mathcal{S} &= \mathcal{T}^{\square} \star \mathcal{S} + \mathcal{P} \\ \mathcal{P} &= \sum_{p,q \in \mathbb{N}} \star^{\square}_{\leq_{p,q}} (\mathcal{R}, \mathcal{X}_1, \ldots, \mathcal{X}_p, \mathcal{Y}_1, \ldots, \mathcal{Y}_q, \mathcal{H}) \\ \mathcal{X}_i &= \mathcal{Y}_i = \mathcal{H} = \mathcal{T} = \mathcal{Z}^{\square} \star (\mathcal{F}^2) \\ \mathcal{R} &= \mathcal{Z}^{\square} \star (\mathcal{F}^3) \end{aligned}$$

Au vu de la spécification des arbres synchronisés, on va devoir greffer des forêts croissantes à la structure principale de l'arbre synchronisé (tronc, branche gauche et branche droite). Une forêt croissante étant spécifiée comme une séquence d'arbres croissant, on peut en générer facilement en utilisant la méthode de génération d'arbre croissant vu au paragraphe précédent. On procède donc en trois étapes principales :

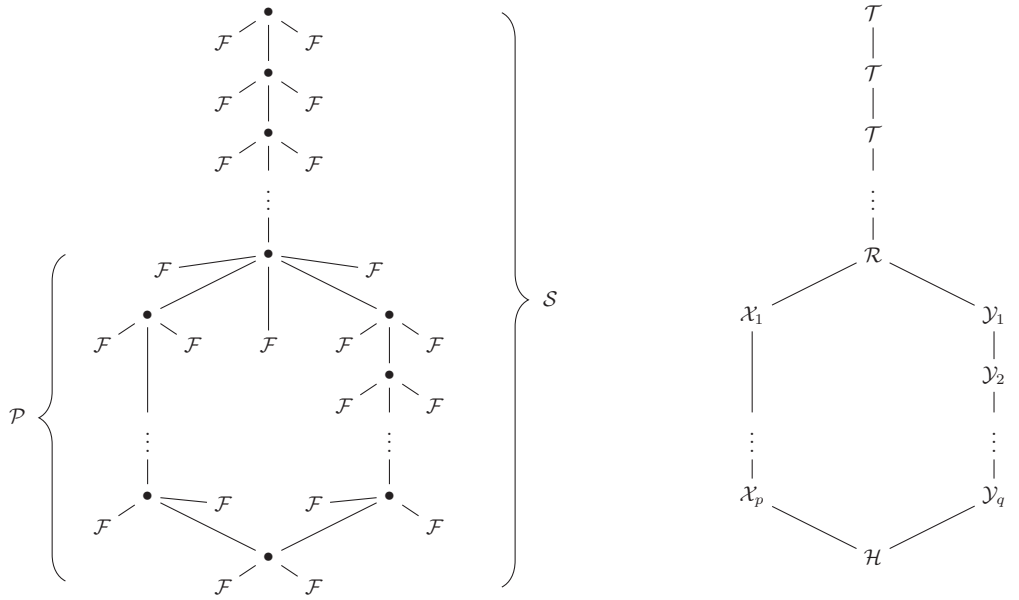


FIGURE 4.3 – Spécification combinatoire des arbres synchronisés

— **Choix des tailles du tronc, de la branche droite et de la branche gauche.** Les premiers éléments que l'on va déterminer sont le nombre de nœuds du tronc, puis les nombres de nœuds sur chaque branche (gauche et droite) de $\mathcal{R}$ à $\mathcal{H}$. Pour cela, on va tout d'abord mettre un poids sur les nœuds avant la séparation, ce qui nous donne :

$$\begin{aligned}
S_1(z,u) &= \int_{v=0}^{z} u\frac{\partial T}{\partial v} S_1(v,u)dv + P(z) \\
&= \int_{t=0}^{z} \left(\frac{\sqrt{1-2t}}{\sqrt{1-2z}}\right)^u P'(t) \\
&= -\frac{((u-3)z+1)\sqrt{-2\,z+1}\,(-2\,z+1)^{\frac{1}{2}\,u} - 4\,z^2 + 4\,z - 1}{4\,(u^2-4\,u+3)z^2(-2\,z+1)^{\frac{1}{2}\,u} - 4\,(u^2-4\,u+3)z(-2\,z+1)^{\frac{1}{2}\,u} + (u^2-4\,u+3)(-2\,z+1)^{\frac{1}{2}\,u}}
\end{aligned}$$

Pour générer un objet de taille n, on tire le nombre de nœuds entre la racine et la séparation selon $[z^n]S_1(z,u)$, ce qui nous permet de modifier la spécification pour en prendre compte. Maintenant on va s'intéresser au nombre de nœuds sur les branches gauche et droite (que l'on peut voir sur la Figure 4.3). On va utiliser deux nouvelles variables x et y pour les compter :

$$\begin{aligned}
P(z,x,y) &= \sum_{p,q\in\mathbb{N}} \int_{r=0}^{z} \int_{x_1=0}^{r} \dots \int_{y_q=0}^{y_{q-1}} \int_{h=0}^{\min(x_p,y_q)} R'(r)xX_1'(x_1)\dots xX_p'(x_p)yY_1'(y_1)\dots H'(h)dr\dots \\
&= \int_{r=0}^{z}\int_{h=0}^{r} \frac{1}{(1-2r)\sqrt{1-2r}}\frac{1}{1-2h}\left(\frac{\sqrt{1-2h}}{\sqrt{1-2r}}\right)^{2(x+y)} dr\ dh \\
&= \frac{\left((x+y)(-2\,z+1)^{\frac{1}{2}\,x+\frac{1}{2}\,y+\frac{1}{2}} - (x+y+1)(-2\,z+1)^{\frac{1}{2}\,x+\frac{1}{2}\,y} + 1\right)(-2\,z+1)^{-\frac{1}{2}\,x-\frac{1}{2}\,y-\frac{1}{2}}}{x^2+(2\,x+1)y+y^2+x}.
\end{aligned}$$

On calcule ensuite $S(z,x,y)$ à partir de $P(z,x,y)$, dont la relation dépend maintenant du nombre que l'on a tiré dans l'étape précédente. On peut donc tirer la taille k de la branche droite selon $[z^n]S(z,x,1)$ et la taille de la branche gauche selon $[x^k z^n]S(z,x,y)$.

— **Choix du reste des tailles.** On va calculer la série $S(z, u_1, \dots)$ qui possède une variable pour chaque classe utilisée dans sa spécification. Pour P, cela donne en introduisant les variables $a, b_1, b_2, \dots, b_p, c_1, \dots, c_q, f$:

$$P(z, a, b_1, \dots) = \int_{r=0}^{z} \dots \int_{h=0}^{\min(x_p, y_q)} aR'(ar)b_1X_1'(b_1x_1)\dots b_pX_p'(b_px_p)c_1Y_1'(c_1y_1)\dots fH'(fh)dr\ dx_1\dots$$

On peut calculer $S(z, u_1, u_2, \dots, a, b_1, \dots)$ à partir de P, en rajoutant les variables u_i pour compter les tailles des forêts entre la racine et la séparation. On tire ensuite uniformément toutes les tailles à utiliser, puis tous les objets correspondants. La recomposition des silhouettes est simple, puisque les relations de parenté sont clairement établies entre les différentes classes.

4.4.2 Génération aléatoire uniforme d'étiquetages

On essaye ici de décrire des méthodes de génération aléatoire d'étiquetage uniforme sur une silhouette donnée. Les arbres croissants ont un algorithme très efficace, en raison de la structure très contrainte du poset sur lesquelles ils reposent (il peut être décrit par un arbre), qui permet notamment d'avoir une relative indépendance entre les sous-arbres d'un même nœud, ce qui s'exprime par la formule des équerres [34, p. 60]. La génération d'un profil uniforme utilise dans le détail la série génératrice qui correspond à la silhouette, puis se ramène à étiqueter un arbre croissant. La génération des étiquettes d'un arbre synchronisé profite de sa faible distance à un étiquetage croissant d'un arbre pour se ramener au problème de l'étiquetage d'arbres croissants.

Étiquetage croissant d'arbres

Dans [6] est décrit un algorithme qui tourne avec une complexité de $O(n \log n)$ avec n la taille de l'arbre de processus initial. L'algorithme repose sur la formule des équerres [34, P. 60] et l'implémentation utilise un générateur de multiensemble dynamique.
L'algorithme 1 en est une version simplifiée. Pour un arbre A on note $p(A)$ le poids de l'arbre, qui correspond au nombre de nœud dans l'arbre. Pour un ensemble d'arbres E, $Tirer(E)$ tire un sous-arbre uniformément sur le nombre total de nœuds, c'est à dire qu'un arbre A a une probabilité $p(A)/S$ d'être tiré, où S est la somme des poids des arbres de E.

Données : Arbre A
Résultat : Liste ordonnée des nœuds de A correspondant à un étiquetage croissant uniforme.
début
  $E \leftarrow \{A\}$;
  $R \leftarrow [\,]$; `/* liste ordonnée des nœuds de l'arbre */`
  tant que $E \neq \emptyset$ **faire**
    $S \leftarrow tirer(E)$; `/* on tire l'arbre suivant */`
    $R \leftarrow R + [racine(S)]$; `/* on rajoute la racine à la liste */`
    $E \leftarrow E \setminus \{S\}$; `/* on enlève l'arbre tiré à l'ensemble de travail */`
    $E \leftarrow E \cup Sous-arbres(S)$; `/* on le remplace par ses sous-arbres */`
  fin
  retourner R
fin

Algorithme 1 : Génération d'étiquetage d'arbre croissant.

Arbres croissants partiellement étiquetés uniforme

Soit une silhouette de la la forme $\mathcal{S} = \mathcal{Z}^{\square} \star (\mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \dots \times \mathcal{S}^{(k)})$. On transforme cette spécification afin de capturer toutes les manières de couper l'arbre de manière croissante (voir la définition 1.3 du chapitre 1). On introduit une classe atomique $\mathcal{U}$ qui va mesurer la taille des coupes possibles. On construit la classe $\overset{\circ}{\mathcal{S}}$ des coupes de $\mathcal{S}$ en décidant à chaque nœud si l'on coupe. On peut écrire cette spécification de la manière suivante :

$$\overset{\circ}{\mathcal{S}} = \mathcal{U}^{\square} \star (\overset{\circ}{\mathcal{S}}{}^{(1)} \times \overset{\circ}{\mathcal{S}}{}^{(2)} \times \dots \times \overset{\circ}{\mathcal{S}}{}^{(k)}) + \mathcal{E},$$

où les $\overset{\circ}{\mathcal{S}}^{(i)}$ sont définis par récurrence de la même manière. On en déduit une série génératrice exponentielle $\overset{\circ}{S}(u)$, où u compte le nombre de coupes admissibles sur cette silhouette.

Pour tirer uniformément un profil, va utiliser la spécification précédente en rajoutant une nouvelle variable atomique par nœud (on multiplie la classe atomique $\mathcal{U}$ par une classe atomique $\mathcal{X}^{(i)}$ correspondant au nœud que l'on spécifie). On obtient ainsi une fonction génératrice de la forme $S(u, x_1, x_2, \dots, x_n)$. Si on veut tirer uniformément un profil de taille k, on choisit uniformément un terme de taille k, puis on tire l'étiquetage normalement sur la coupe en utilisant par exemple l'algorithme 1.

Exemple 4.5. *Prenons la silhouette définie par la spécification suivante :*

$$P = (\mathcal{Z}^{\square} \star \mathcal{Z}) \times \mathcal{Z} \times \mathcal{Z} \times (\mathcal{Z}^{\square} \star (\mathcal{Z} \times \mathcal{Z})).$$

On lui associe la spécification suivante :

$$\mathcal{C}_P = \Big((\mathcal{U}_a^{\square} \star (\mathcal{U}_b + \mathcal{E}) + \mathcal{E})\Big) \times (\mathcal{U}_c + \mathcal{E}) \times (\mathcal{U}_d + \mathcal{E}) \times \Big(\mathcal{U}_e^{\square} \star ((\mathcal{U}_f + \mathcal{E}) \times (\mathcal{U}_g + \mathcal{E})) + \mathcal{E}\Big).$$

Qui se traduit en équation, avec les paramètres x_i :

$$C_P(u, x_1, x_b, \dots) = \left(\int_{t=0}^{u} x_a(1 + x_b t)dt + 1\right) \cdot \left[(1 + x_c u)(1 + x_d u) \cdot \left(\int_0^u x_e(1 + x_f t)(1 + x_g t)dt + 1\right)\right].$$

Ce qui donne la formule suivante :

$$\begin{aligned}
C_P(u, x_1, x_b, \dots) = & \frac{1}{6} x_a x_b x_c x_d x_e x_f x_g u^7 + \\
& \frac{1}{12}((3x_a x_b x_c x_d x_e x_f + (3x_a x_b x_c x_d x_e + 2(x_a x_b x_c + (x_a x_b + 2x_a x_c)x_d)x_e x_f)x_g))u^6 + \\
& \frac{1}{12}((6x_a x_b x_c x_d x_e + 3(x_a x_b x_c + (x_a x_b + 2x_a x_c)x_d)x_e x_f + (2(x_a x_b + 2x_a x_c + 2(x_a + x_c)x_d)x_e x_f + \\
& \quad 3(x_a x_b x_c + (x_a x_b + 2x_a x_c)x_d)x_e)x_g))u^5 + \\
& \frac{1}{12}((3(x_a x_b + 2x_a x_c + 2(x_a + x_c)x_d)x_e x_f + 6(x_a x_b x_c + (x_a x_b + 2x_a x_c)x_d)x_e + (4(x_a + x_c + x_d)x_e x_f + \\
& \quad 3(x_a x_b + 2x_a x_c + 2(x_a + x_c)x_d)x_e)x_g))u^4 + \\
& \frac{1}{6}((3(x_a + x_c + x_d)x_e x_f + 3(x_a x_b + 2x_a x_c + 2(x_a + x_c)x_d)x_e + (3(x_a + x_c + x_d)x_e + 2x_e x_f)x_g))u^3 + \\
& \frac{1}{2}((x_a x_b + 2(x_a + x_c + x_d)x_e + x_e x_f + x_e x_g))u^2 + \\
& (x_a + x_e)u + 1.
\end{aligned}$$

Si l'on veut tirer un profil de taille 4 à partir de cet arbre, il suffit donc de tirer uniformément un des termes devant u^4 (on connaît le poids d'un terme en remplaçant toutes les variables par 1). On tire ensuite un profil correspondant à ce terme uniformément en utilisant l'algorithme d'étiquetage des arbres croissants. Par exemple on tire l'étiquetage correspondant au terme $x_a x_d x_e x_g$ avec probabilité $\frac{6/12}{78/12} = \frac{1}{13}$, il faut tirer un étiquetage croissant du profil $P_{adeg} = (\mathcal{U}^{\square} \star \mathcal{E}) \times \mathcal{E} \times \mathcal{U} \times (\mathcal{U}^{\square} \star (\mathcal{E} \times \mathcal{U}))]$.

Arbres synchronisés.

On va ici profiter de la faible distance qu'a un arbre synchronisé avec un arbre croissant. On va le décomposer en deux arbres, que l'on va étiqueter séparément : l'arbre avec des étiquettes plus petites que l'étiquette synchronisée, et l'arbre avec des étiquettes plus grandes. On va pour cela utiliser la même technique que pour la génération du profil. Pour chaque forêt insérée dans la structure de l'arbre croissant (sauf celles insérées en dessous de la synchronisation), on calcule son profil, et on l'insère dans la structure d'arbre synchronisé (avec une variable par forêt). On peut ensuite l'utiliser pour choisir uniformément l'étiquette du nœud synchronisé, puis on tire un profil correspondant. On rassemble enfin les nœuds n'ayant pas d'étiquettes, ce qui forme un arbre croissant avec le nœud synchronisé comme racine, et on l'étiquette uniformément en utilisant l'algorithme 1.

4.5 Conclusion

Ce chapitre étend le cadre classique de génération récursive pour montrer comment étiqueter efficacement des structures avec des contraintes d'ordres non triviales, comme par exemple celles étudiées au premier et deuxième chapitres. On montre comment décomposer la générations en deux parties : la génération de la silhouette et celle des étiquettes. La génération des silhouettes se fait naturellement en utilisant les principes classiques de la génération récursive. La génération des étiquettes est habituellement négligée dans les schémas de génération classiques, puisque les objets produits ont un étiquetage sans contrainte, et l'étiquetage se réduit ainsi à tirer une permutation aléatoire. On montre que dans le cas où la contrainte sur les étiquetages peut se décrire à l'aide d'un arbre, on peut générer l'étiquetage efficacement.

Chapitre 5

Analyse quantitative de différentes versions de la génération de Boltzmann

Dans les deux chapitres qui vont suivre, on va quitter le monde de la génération récursive pour s'intéresser à la génération de Boltzmann. La méthode de Boltzmann est une méthode de génération aléatoire relativement récente qui permet de générer des objets combinatoires uniformément.

La principale différence par rapport à la méthode récursive est que l'on tire dans une *distribution* de taille au lieu de tirer des objets d'une taille fixée. Cela permet d'avoir une méthode simple et efficace, qui limite les pré-calculs, mais se paye si on désire avoir des objets dans un certain intervalle. En effet on doit pour cela rejeter les objets de mauvaise taille, ce qui gâche des tirages d'aléa et coûte du temps de calcul de manière générale.

Un choix habile de distribution doit alors se faire, pour en prendre une pratique à calculer et qui minimise les pertes dues au rejet. Dans le cadre des générateurs de Boltzmann, la distribution choisie est imposée et correspond à la distribution de Boltzmann. Bien que n'étant pas la distribution qui minimise le rejet, sa distribution uniforme pour une taille fixée, elle est dérivée simplement des fonctions génératrices et possède aussi l'avantage d'être paramétrée, ce qui lui donne une plus grande flexibilité. Il ne reste donc plus qu'à choisir le paramétrage, afin que la distribution se comporte le mieux possible *pour la taille que l'on désire avoir*. Une méthode d'optimisation possible consiste à changer de classe d'objet à générer, ce qui implique une distribution de Boltzmann différente, de générer un objet dans cette nouvelle classe, puis de le transformer en un objet de la première classe. Il faut bien entendu s'assurer que la transformation va préserver l'uniformité.

Le chapitre qui suit est consacré à comparer finement deux méthodes de génération de Boltzmann, la première utilisant une méthode de transformation de classe pour diminuer le rejet, l'autre s'appuyant sur une évaluation rapide du paramétrage.

5.1 Contexte des générateurs de Boltzmann

Dans cette partie, on va rappeler les définitions classiques dans le contexte des structures combinatoires et des générateurs de Boltzmann. Nous ne nous intéresserons dans cette partie uniquement à des classes combinatoires non étiquetées (associées au fonctions génératrices ordinaires). La modification à faire pour l'adapter au modèle des structures étiquetées est directe, puisque la modification apportée aux séries génératrices (le terme en $n!$) est compensé dans la définition de la distribution de Boltzmann exponentielle, que qui rend les calculs identiques dans les deux cas.

Pour tout ce chapitre, on utilisera comme *exemple référent* la classe des arbres binaires (définie en 0.2).

5.1.1 Générateur de Boltzmann

On rappelle (cf. 0.3.2) la définition de la distribution de Boltzmann :

Définition 5.1. *Soit $\mathcal{C}$ une classe combinatoire non étiquetée, de fonction génératrice $C(z)$. Son modèle de Boltzmann associé, dépendant d'un paramètre réel $0 < z < \rho$ ou $z = \rho$ dans certains cas, est la distribution sur les objets de $\mathcal{C}$, telle que la probabilité d'un objet γ est*

$$\mathbb{P}_z(\gamma) = \frac{z^{|\gamma|}}{C(z)}, \text{ où } C(z) \text{ est la fonction génératrice de } \mathcal{C} \text{ évaluée en } z.$$

Un *générateur de Boltzmann* est un constructeur qui génère des objets selon leur distribution de Boltzmann. Puisque la taille de l'objet généré est aléatoire, pour tirer un objet de taille n, les générations sont répétées jusqu'à obtenir un objet de la bonne taille. Pour obtenir une génération plus efficace [20], on s'accorde généralement une marge d'erreur sur la taille des objets. Par exemple, on acceptera les objets dont la taille est dans les bornes $n(1-\varepsilon)$ et $n(1+\varepsilon)$, où ε est la marge d'erreur.

Nous allons restreindre l'étude aux classes combinatoires dont les fonctions génératrices sont Δ – singulières. Cette contrainte sur les classes combinatoires est assez technique, néanmoins elle couvre une très large gamme de classes, car elle est associé aux objets combinatoires décomposables, ainsi qu'on peut le voir dans [26].

Définition 5.2. *Une fonction $C(z)$ analytique en 0 et avec un rayon d'analyticité $\rho > 0$ est dite* Δ-singulière *si elle vérifie les deux conditions suivantes :*

(i) La fonction admet ρ comme unique singularité en $|z| = \rho$, et est relevable sur un domaine

$$\Delta(r, \theta) = \{z \mid z \neq \rho, |z| < r, arg(z - \rho) \notin (-\theta, \theta)\},$$

pour un $r > \rho$ et un θ vérifiant $0 < \theta < \frac{\pi}{2}$. L'ensemble $\Delta(r, \theta)$ est appelée un Δ-domaine.

(ii) Pour z convergeant vers ρ dans le Δ-domaine, $C(z)$ vérifie un développement de la forme :

$$C(z) =_{z\to\rho} P(z) + c_0(1 - z/\rho)^{-\alpha}) + o((1 - z/\rho)^{-\alpha}, \alpha \in \mathbb{R}\backslash\{0, -1, -2, ...\},$$

où $P(z)$ est un polynôme ; c_0 une constante et $-\alpha$ est appelé l'exposant singulier *de $C(z)$.*

En utilisant le théorème de transfert de Flajolet-Odlysko (cf. [24, chapitre VI] pour les détails), les coefficients des fonctions Δ-singulières vérifient le comportement asymptotique suivant :

$$C_n := [z^n]C(z) \sim \frac{c_0}{\Gamma(\alpha)}\rho^{-n}n^{\alpha-1}. \tag{5.1}$$

Cet équivalent asymptotique est l'élément qui sera crucial pour les preuves, les autres propriétés des fonctions Δ-singulières n'étant pas utilisées [1].

Notons que pour $-\alpha > 0$, la queue de la distribution de taille, c'est-à-dire la partie de la distribution éloignée de la moyenne, est importante, et les distributions ayant cette propriété sont généralement appelées *distributions piquées*, tandis que les distributions pour $-\alpha < 0$ sont appelées *distributions plates*. La classe des arbres binaires fait partie des structures à distribution piquée, puisque dans ce cas $-\alpha = 1/2 > 0$.

Pour générer des objets selon leur distribution de Boltzmann, Duchon *et al.* [20] ont présenté trois générateurs différents. Le premier, que nous appellerons *générateur de Boltzmann classique*, se contente de choisir son paramètre z_n de manière à ce qu'il soit le meilleur possible, c'est-à-dire pour maximiser ses chances d'avoir un objet de la taille voulue n. Le deuxième générateur, appelé *générateur de Boltzmann pointé*, s'utilise sur des distributions piquées (provenant de structures arborescentes par exemple). Pour changer la distribution de piquée en plate, on peut marquer un atome de la structure, puis générer des structures marquées. La distribution des structures ainsi générées est uniforme sur les objets marqués pour une taille donnée. On peut alors se rendre compte qu'en enlevant la marque on obtient un objet pris dans une distribution uniforme sur les objets non marqués à taille donnée. Le générateur de Boltzmann pointé consiste donc générer des objets appartenant à la classe pointée en utilisant un générateur de Boltzmann classique, puis d'enlever la

1. Ainsi si l'on trouve une fonction qui vérifie cet équivalent sans être Δ-singulière, les théorèmes s'adaptent de façon assez directe.

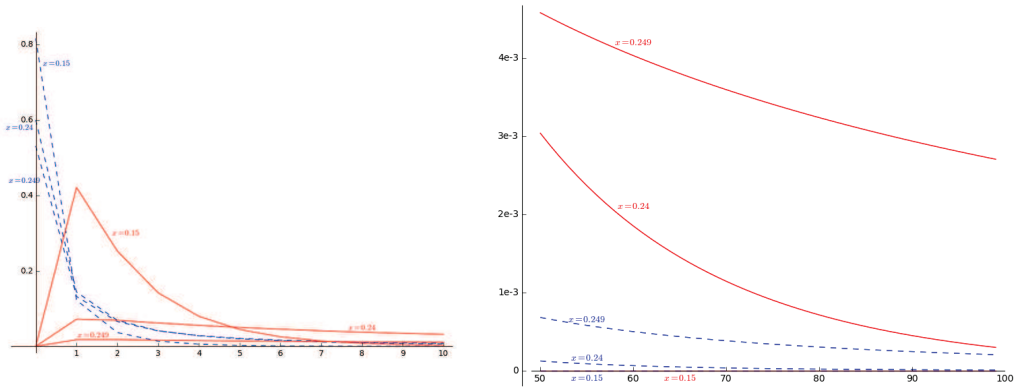


FIGURE 5.1 – Distributions de Boltzmann avec différents paramètres z (deux distributions piquées en traits bleus, deux distributions plates en lignes rouges).

marque. Cette méthode est évidemment uniforme sur les objets de même taille, même si la distribution utilisée n'est pas celle de Boltzmann pour notre objet (donnée dans la définiton 5.1). Finalement, le troisième générateur, appelé *générateur de Boltzmann singulier*, s'applique sur des structures à distribution piquée, pour lequel on utilise la singularité comme paramètre. C'est une valeur de paramètre non optimale dans le cas général [2], et on mise ici sur le fait que sur les distributions piquées la valeur de paramètre optimale ne soit que marginalement meilleure que celle égale à la singularité.

5.2 Complexité des générateurs de Boltzmann

On cherche ici à déterminer une mesure pertinente de la complexité d'un générateur de Boltzmann. Le précalcul peut être dur, mais ne dépend que de la taille des objets à générer et le type d'objets, et non pas du nombre d'objets à générer, puisqu'il s'agit de calculer et stocker un nombre constant de paramètres calculés avec une précision arbitraire [3] On peut notamment distinguer deux cas principaux : celui où la valeur de la singularité dominante de la fonction est simple à calculer dans quel cas le coût est négligeable, et celui où calculer la singularité dominante se calcule difficilement, et peut se retrouver équivalent à celui de calculer les termes de la série jusqu'à la taille désirée.

On ne comptera pas donc pas le précalcul dans la mesure de la complexité, mais c'est évidemment à prendre en compte au moment où l'on compare les différentes méthodes, qui n'auront pas les mêmes précalculs. La complexité en nombre de bit aléatoires, en temps et en espace de la génération d'un objet à l'aide d'un générateur de Boltzmann croît dans la plupart des cas linéairement avec la taille des objets générés. (cf. [20]), on se contentera donc de compter la taille totale générée par notre générateur. Il y a donc deux parts qui contribuent à la complexité de la génération : la partie qui correspond à l'objet que l'algorithme donne en sortie (par définition de taille approximativement n), et la partie qui correspond aux objets rejetés. La première partie étant essentiellement constante (entre $n(1-\epsilon)$ et $n(1+\epsilon)$) entre toutes les générations, on ne va pas la compter et se concentrer sur la somme des tailles des objets rejetés, qui sera notre mesure de complexité.

Lorsque l'on considère cette mesure de complexité, une amélioration évidente du générateur consiste à ne pas générer d'objets trop grands, sachant qu'on va les rejeter. L'algorithme de génération étant récursif, on peut avoir pendant la génération un minimum de la taille finale de l'objet généré. En gardant trace de la taille de l'objet que l'on est en train de générer, on peut le rejeter avant la fin de la génération si il dépasse une taille choisie. Cette méthode est appelée *rejet anticipé*.

Le résultat ci-dessous est un lemme prouvé dans [20], qui nous sera utile pour ce chapitre. Il donne la complexité du générateur de Boltzmann pour des objets de taille approchée avec et sans rejet anticipé.

2. Sinon cette méthode se confondrait avec la première.
3. On considère que cela prend une place constante et donc avec une précision fixée, car si on a besoin du n-ième bit qu'avec probabilité environ 2^{-n}, on peut faire un calcul à la volée.

Lemme 5.1. *([20]) Soit $C(z)$ la fonction génératrice de la classe $\mathcal{C}$, et soit $C^{<n_1}$, $C^{>n_2}$ et $C^{[n_1,n_2]}$ les fonctions génératrices pour les sous-classes d'objets de $\mathcal{C}$ de taille, respectivement, strictement plus petite que n_1, strictement plus grande que n_2, et entre n_1 et n_2. La taille cumulée, T_n, des objets générés et ensuite rejetés car n'étant pas dans la fenêtre de tolérance $[n_1, n_2]$ et le paramètre z vérifie*[4] *:*

$$\mathbb{E}(T_n) = \frac{zC'^{<n_1}(z) + n_2 C^{>n_2}(z)}{C^{[n_1,n_2]}(z)} \qquad \text{et}$$

$$\mathbb{E}(T_n^2) = \frac{z^2 C''^{<n_1}(z) + n_2(n_2-1)C^{>n_2}(z)}{C^{[n_1,n_2]}(z)} + 2\mathbb{E}(T_n)^2 + \mathbb{E}(T_n).$$

Sans rejet anticipé, on obtient :

$$\mathbb{E}(T_n) = \frac{zC'^{<n_1}(z) + zC'^{>n_2}(z)}{C^{[n_1,n_2]}(z)} \qquad \text{et}$$

$$\mathbb{E}(T_n^2) = \frac{z^2 C''^{<n_1}(z) + z^2 C''^{>n_2}(z)}{C^{[n_1,n_2]}(z)} + 2\mathbb{E}(T_n)^2 + \mathbb{E}(T_n).$$

Démonstration. La preuve ci-dessous est donnée dans [20].
La fonction génératrice de probabilité du générateur[5] de Boltzmann avec une fenêtre de tolérance $[n_1, n_2]$ est

$$F(u,z) = \sum_k \mathbb{P}(T_n(z) = k)u^k.$$

Un appel au générateur va se décomposer en une succession d'appels infructueux, suivi d'un appel fructueux qui arrive avec probabilité $F_{acc}(z)$, et qui ne compte pas dans la taille de T_n. Si on note $A(z)$ la variable aléatoire donnant la taille d'un appel, et $F_{nacc}(u,z)$ la fonction génératrice de probabilité des appels infructueux

$$F_{nacc}(u,z) = \sum_{k\in\mathbb{N}\setminus\{n_1,\ldots,n_2\}} \mathbb{P}\left(A(z) = k\right) u^k,$$

la remarque précédente implique que :

$$F(u,z) = (1 - F_{nacc}(u,z))^{-1} F_{acc}(z).$$

La probabilité $F_{acc}(z)$ est calculable simplement d'après la définition de la distribution de Boltzmann :

$$F_{acc}(z) = \frac{\sum_{n=n_1}^{n_2} C_n z^n}{\sum_{n=1}^{\infty} C_n z^n} = \frac{C^{[n_1,n_2]}(z)}{C(z)}.$$

$F_{nacc}(u,z)$ se décompose en deux parties :

— Les appels plus petits que n_1 s'écrivent simplement :

$$\frac{\sum_{n=1}^{n_1} C_n z^n u^n}{\sum_{n=1}^{\infty} C_n z^n} = \frac{C^{<n_1}(uz)}{C(z)}.$$

— Pour les appels plus grand que n_2, cela dépend de la présence de rejet anticipé ou non.

- Dans le cas du rejet anticipé, la taille est constante égale à n_2, qui est comptée par la variable u, cette partie est égale à :

$$\frac{u^{n_2} \sum_{n=n_2}^{\infty} C_n z^n}{\sum_{n=1}^{n_1} C_n z^n} = \frac{u^{n_2} C^{>n_2}(z)}{C(z)}.$$

4. T_n dépend de z, n_1 et n_2, mais ils ne sont pas notés comme des paramètres pour ne pas surcharger les notations.
5. C'est ici une fonction génératrice en fonction de u, la variable z est un paramètre qui est fixé.

- Dans le cas sans rejet anticipé, la taille est variable, comme pour $n < n_1$, cette partie est égale à :

$$\frac{\sum_{n=n_2}^{\infty} C_n z^n u^n}{\sum_{n=1}^{n_1} C_n z^n} = \frac{C^{>n_2}(uz)}{C(z)}.$$

D'où

$$F(u,z) = \left(1 - \frac{1}{C(z)}(C^{<n_1}(uz) + u^{n_2}C^{>n_2}(z))\right)^{-1} \frac{C^{[n_1,n_2]}(z)}{C(z)}$$

dans le cas avec rejet anticipé, et

$$F(u,z) = \left(1 - \frac{1}{C(z)}(C^{<n_1}(uz) + C^{>n_2}(uz))\right)^{-1} \frac{C^{[n_1,n_2]}(z)}{C(z)}$$

dans le cas sans rejet anticipé.
L'espérance des coûts et le second moment sont donnés respectivement par

$$\mathbb{E}(T_n) = \frac{u\partial}{\partial u} F(u,z)_{|u=1} \qquad \text{et}$$
$$\mathbb{E}(T_n^2) = \left(\frac{u\partial}{\partial u} + \frac{u^2\partial^2}{\partial u^2}\right) F(u,z)_{|u=1}.$$

On obtient donc :

$$\mathbb{E}(T_n) = \left(\left(zC'^{<n_1}(uz) + n_2 u^{n_2-1}C^{>n_2}(z)\right) \frac{C^{[n_1,n_2]}(z)}{\left(C(z) - C^{<n_1}(uz) - u^{n_2}C^{>n_2}(z)\right)^2}\right)_{|u=1}$$
$$\mathbb{E}(T_n^2) = \left(\left(z^2C''^{<n_1}(uz) + n_2(n_2-1)u^{n_2-2}C^{>n_2}(z)\right) \frac{C^{[n_1,n_2]}(z)}{\left(C(z) - C^{<n_1}(uz) - u^{n_2}C^{>n_2}(z)\right)^2}\right)_{|u=1}$$
$$+ 2\mathbb{E}(T_n)^2 + \mathbb{E}(T_n)$$

dans le cas avec rejet, et :

$$\mathbb{E}(T_n) = \left(\left(zC'^{<n_1}(uz) + z^2C^{>n_2}(z)\right) \frac{C^{[n_1,n_2]}(z)}{\left(C(z) - C^{<n_1}(uz) - C^{>n_2}(uz)\right)^2}\right)_{|u=1},$$
$$\mathbb{E}(T_n^2) = \left(\left(z^2C''^{<n_1}(uz) + z^2C^{>n_2}(z)\right) \frac{C^{[n_1,n_2]}(z)}{\left(C(z) - C^{<n_1}(uz) - C^{>n_2}(uz)\right)^2}\right)_{|u=1} + 2\mathbb{E}(T_n)^2 + \mathbb{E}(T_n)$$

dans le cas sans rejet. On remarque que $C(z) = C^{<n_1}(z) + C^{[n_1,n_2]}(z) + C^{>n_2}(z)$, ce qui donne les résultats annoncés. □

Ce lemme permet de calculer l'espérance et la variance du coût d'un générateur de Boltzmann avec rejet anticipé connaissant les valeurs de $zC'^{<n_1}(z)$, $n_2C^{>n_2}(z)$, $C^{[n_1,n_2]}(z)$ et $z^2C''^{<n_1}(z)$. Dans le cas où $z < \rho$, ces valeurs existent, et on peut les calculer explicitement dans le cas où l'on connaît $C(z)$. Nous concluons cette section en rappelant quelques notations associées à chaque générateur, qui ont été introduites dans [20]. Notons $\mu C(z,n,\varepsilon)$ pour désigner un générateur de Boltzmann pour la classe combinatoire $\mathcal{C}$ sans rejet anticipé, et notons $\nu C(z,n,\varepsilon)$ pour le générateur de Boltzmann associé avec rejet anticipé.

Nous allons maintenant passer au calcul de complexité des générateurs. Pour faciliter l'étude nous séparons les structures combinatoires selon la valeur de leur exposant singulier.

5.3 Objets à distribution plate

Dans cette section, nous considérons des structures combinatoires telles que leur exposant singulier vérifie $-\alpha < 0$, ce qui correspond à des objets dont la distribution de Boltzmann est plate.

Exemple 5.1. *Arbres binaires marqués.*

On va construire un exemple simple d'objet ayant une distribution de Boltzmann plate : la classe $\mathcal{B}^\bullet$ les arbres binaires avec un nœud marqué. En effet, cette classe peut se spécifier de la manière suivante :

$$\mathcal{B}^\bullet = \mathcal{E} + \mathcal{Z} \times \mathcal{B}^\bullet \times \mathcal{B} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}^\bullet,$$

où $\mathcal{B}$ est la classe des arbres binaires classiques. Ainsi la fonction génératrice ordinaire de la classe $\mathcal{B}^\bullet$ est :

$$B^\bullet(z) = \frac{1}{\sqrt{1-4z}},$$

Qui a un exposant singulier $-\alpha < 0$.

Choix du paramètre pour une distribution plate. La valeur optimale pour le paramètre z correspond à la valeur qui maximise la probabilité de tirer un objet dans l'intervalle $[n_1, n_2]$ désiré. Trouver cette valeur optimale n'est en général pas trivial, puisqu'elle consiste à calculer la probabilité de tomber dans un intervalle précis pour une distribution, Une manière plus simple (et du coup largement utilisée) consiste à choisir un paramètre approprié z_n afin que l'espérance de la taille des objets générés soit égale ou proche de n. Cette valeur n'est pas optimale, mais dans la plupart des cas les distributions sont centrées sur la moyenne ce qui rend cette approximation valide. Elle a l'avantage dans le cas des distribution plates d'être assez facilement approximable.

Nous allons donc réaliser cette approximation en prenant z_n proche de la solution de $\mathbb{E}_{z_n}(N) = z_n$. Dans le cas $-\alpha < 0$, on a $z_n \sim \rho(1 - \frac{\alpha}{n})$. Nous allons donc choisir arbitrairement comme valeur de paramètre

$$z_n = \rho\left(1 - \frac{\alpha}{n}\right). \tag{5.2}$$

Cette approximation a l'avantage d'être très facile à calculer (on a juste besoin de la valeur de l'exposant singulier et de la singularité), et pratique pour l'analyse.

Maintenant que l'on s'est choisi une valeur pour le paramètre, on peut calculer l'efficacité des générateurs de Boltzmann sur ce type de structures.

5.3.1 Générateur de Boltzmann classique sans rejet anticipé

On va d'abord regarder le cas sans rejet, qui est clairement moins efficace dans notre mesure de la complexité, mais qui pourrait être intéressant si on prend en compte d'autre coûts. Le rejet nécessite en effet de savoir la taille de l'objet à tout instant de sa construction, ce qui limite le caractère parallélisable de la génération de Boltzmann. Le théorème suivant est une version généralisée d'un théorème démontré dans l'article [20].

Théorème 5.1. *Soit $\mathcal{C}$ une classe combinatoire dont la fonction génératrice est Δ-singulière d'exposant $-\alpha < 0$. Alors la taille cumulée T_n des objets générés et rejetés par un générateur de Boltzmann classique approché sans rejet anticipé, $\mu C(z_n, n, \varepsilon)$, satisfait, quand n tends vers l'infini :*

$$\mathbb{E}(T_n) \sim n\kappa(\varepsilon, \alpha), \qquad \textit{et} \qquad \mathbb{E}(T_n^2) \sim 2\mathbb{E}(T_n)^2 + n^2\kappa'(\varepsilon, \alpha),$$

où

$$\kappa(\varepsilon,\alpha) = \frac{\displaystyle\int_{w=0}^{1-\varepsilon} w^{\alpha}e^{-\alpha w}dw + \int_{w=1+\varepsilon}^{\infty} w^{\alpha}e^{-\alpha w}dw}{\displaystyle\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1}e^{-\alpha w}dw}$$

et

$$\kappa'(\varepsilon,\alpha) = \frac{\displaystyle\int_{w=0}^{1-\varepsilon} w^{\alpha+1}e^{-\alpha w}dw + \int_{w=1+\varepsilon}^{\infty} w^{\alpha+1}e^{-\alpha w}dw}{\displaystyle\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1}e^{-\alpha w}dw}.$$

Démonstration. On a besoin pour appliquer le Lemme 5.1 de calculer les quantités suivantes : $z_nC'^{<n(1-\varepsilon)}(z_n)$, $z_nC'^{>n(1+\varepsilon)}(z_n)$ et $C^{[n(1-\varepsilon),n(1+\varepsilon)]}(z_n)$, $z_n^2C''^{<n(1-\varepsilon)}(z_n)$ et $z_n^2C''^{>n(1+\varepsilon)}(z_n)$.

On va montrer comment calculer un équivalent de $z_nC'^{<n(1-\varepsilon)}(z_n)$, les autres expressions étant trouvées d'une manière similaire. On utilise l'approximation de l'équation (5.1) avec la valeur choisie pour z_n définie à l'équation (5.2) :

$$\begin{aligned} z_nC'^{<n(1-\varepsilon)}(z_n) &= \frac{c_0}{\Gamma(\alpha)}\sum_{k=0}^{\lceil n(1-\varepsilon)\rceil} e^{k\ln(1-\alpha/n)}k^{\alpha}(1+o_{k\to\infty}(1)) \\ &= \frac{c_0}{\Gamma(\alpha)}\sum_{k=0}^{\lceil n(1-\varepsilon)\rceil} e^{-\frac{\alpha k}{n}}k^{\alpha}(1+o_{k\to\infty}(1)). \end{aligned}$$

Puisque la somme $\sum_{k=0}^{\lceil n(1-\varepsilon)\rceil} e^{-\frac{\alpha k}{n}}k^{\alpha}$ tend vers l'infini quand n tend vers l'infini, on peut tirer le terme d'erreur hors de la somme :

$$z_nC'^{<n(1-\varepsilon)}(z_n) \underset{n\to\infty}{=} \frac{c_0}{\Gamma(\alpha)}\left(\sum_{k=0}^{\lceil n(1-\varepsilon)\rceil} e^{-\frac{\alpha k}{n}}k^{\alpha}\right)(1+o_{n\to\infty}(1)).$$

On utilise ensuite la formule sommatoire d'Euler-MacLaurin :

$$\frac{c_0}{\Gamma(\alpha)}\left(\sum_{k=0}^{\lceil n(1-\varepsilon)\rceil} e^{-\frac{\alpha k}{n}}k^{\alpha}\right)(1+o(1)) = \frac{c_0}{\Gamma(\alpha)}\left(\int_{t=1}^{\lceil n(1-\varepsilon)\rceil} t^{\alpha}e^{-\alpha t}dt\right)(1+o(1)).$$

Ainsi

$$z_nC'^{<n(1-\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0n^{\alpha+1}}{\Gamma(\alpha)}\int_{w=0}^{1-\varepsilon} w^{\alpha}e^{-\alpha w}dw.$$

On peut utiliser la formule d'Euler-MacLaurin sur les coefficients de $C(z)$ et le même type d'analyse pour avoir les estimations suivantes :

$$C^{>n(1+\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0n^{\alpha+1}}{\Gamma(\alpha)}\int_{w=1+\varepsilon}^{\infty} w^{\alpha}e^{-\alpha w}dw,$$

$$C^{[n(1-\varepsilon),n(1+\varepsilon)]}(z_n) \underset{n\to\infty}{\sim} \frac{c_0n^{\alpha}}{\Gamma(\alpha)}\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1}e^{-\alpha w}dw,$$

$$z_n^2 C''^{<n(1-\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+2}}{\Gamma(\alpha)} \int_{w=0}^{1-\varepsilon} w^{\alpha+1} e^{-\alpha w} dw,$$

et

$$z_n^2 C''^{>n(1+\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+2}}{\Gamma(\alpha)} \int_{w=1+\varepsilon}^{\infty} w^{\alpha+1} e^{-\alpha w} dw.$$

Il ne reste plus qu'à appliquer le Lemme 5.1 pour calculer le premier et le second moment de la mesure de complexité. □

Si l'on fixe une classe combinatoire $\mathcal{C}$ (et un ϵ), toutes ces intégrales peuvent être calculées explicitement indépendamment de n, et la comparaison des différentes méthodes devient directe (voir l'exemple dans la sous-section 5.5).

5.3.2 Générateur de Boltzmann classique avec rejet anticipé

Ici à chaque étape de construction d'un objet, on regarde la taille de l'objet en construction, et on la compare à la taille maximale acceptée. Si on l'a déjà dépassé avant la fin de la construction, on arrête la construction, on rejette ce que l'on a fait et on construit un nouvel objet.

Théorème 5.2. *Soit $\mathcal{C}$ une classe combinatoire dont la fonction génératrice est Δ-singulière d'exposant $-\alpha < 0$. Alors la taille cumulée T_n des objets générés et rejetés par un générateur de Boltzmann approché avec rejet anticipé, $\nu C(z_n, n, \varepsilon)$, satisfait, quand n tends vers l'infini :*

$$\mathbb{E}(T_n) \underset{n\to\infty}{\sim} n\kappa_r(\varepsilon,\alpha), \qquad \text{et} \qquad \mathbb{E}(T_n^2) \underset{n\to\infty}{\sim} 2\mathbb{E}(T_n)^2 + n^2\bar{\kappa}_r(\varepsilon,\alpha),$$

où

$$\kappa_r(\varepsilon,\alpha) = \frac{\displaystyle\int_{w=0}^{1-\varepsilon} w^{\alpha} e^{-\alpha w} dw + \int_{w=1+\varepsilon}^{\infty} (1+\varepsilon)\cdot w^{\alpha-1} e^{-\alpha w} dw}{\displaystyle\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1} e^{-\alpha w} dw},$$

et

$$\bar{\kappa}_r(\varepsilon,\alpha) = \frac{\displaystyle\int_{w=0}^{1-\varepsilon} w^{\alpha+1} e^{-\alpha w} dw + \int_{w=1+\varepsilon}^{\infty} (1+\varepsilon)\cdot w^{\alpha-1} e^{-\alpha w} dw}{\displaystyle\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1} e^{-\alpha w} dw}.$$

Le schéma de la preuve est identique à celui du Théorème 5.1.

Démonstration. Notre but est d'utiliser le Lemme 5.1. On a donc besoin des estimations pour $z_n C'^{<n(1-\varepsilon)}(z_n)$, $n(1+\varepsilon)C^{>n(1+\varepsilon)}(z_n)$, $C^{[n(1-\varepsilon),n(1+\varepsilon)]}(z_n)$ et $z_n^2 C''^{<n(1-\varepsilon)}(z_n)$. On utilise la formule sommatoire d'Euler-MacLaurin sur les coefficients de $C(z)$ pour obtenir ces estimations :

$$z_n C'^{<n(1-\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+1}}{\Gamma(\alpha)} \int_{w=0}^{1-\varepsilon} w^{\alpha} e^{-\alpha w} dw,$$

$$n(1+\varepsilon)C^{>n(1+\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+1}}{\Gamma(\alpha)} \int_{w=1+\varepsilon}^{\infty} w^{\alpha-1} e^{-\alpha w} dw,$$

$$C^{[n(1-\varepsilon),n(1+\varepsilon)]}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha}}{\Gamma(\alpha)} \int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha-1} e^{-\alpha w} dw,$$

et

$$z_n^2 C''^{<n(1-\varepsilon)}(z_n) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+2}}{\Gamma(\alpha)} \int_{w=1+\varepsilon}^{\infty} w^{\alpha+1} e^{-\alpha w} dw.$$

□

5.4 Objets avec distribution piquée

Dans cette section, on considère des structures combinatoires, dont l'exposant singulier $-\alpha$ est positif. Ces objets présentent une distribution piquée. Les structures d'arbres non marquées sont des exemples de cette sorte d'objets avec $-\alpha$ étant un nombre rationnel positif, plus petit que 1.

Choix du paramètre pour une distribution piquée. Dans le cas où $-\alpha > 0$, on peut toujours calculer une approximation du paramètre z_n solution de $\mathbb{E}_{z_n}(N) = z_n$, ainsi qu'on le fait dans le paragraphe 5.3. En revanche l'approximation 5.2 n'est plus valide, et même la solution exacte offre un gain en complexité négligeable par rapport à une génération singulière.

On va donc changer de méthode dans ce cas, en utilisant soit un générateur de Boltzmann pointé, soit un générateur de Boltzmann singulier.

5.4.1 Générateur de Boltzmann pointé

L'idée va être de changer la distribution afin de se retrouver dans le cas des distribution plates, et se reporter aux résultats de la section précédente. On fait cela en générant des objets pointés. Un objet pointé est une structure avec un élément marqué. En générant un objet pointé avec un générateur de Boltzmann puis en retirant la marque, on réalise une génération de Boltzmann pointée. On peut remarquer que le générateur génère des objets d'un ensemble beaucoup plus large que celui considéré, puis projette l'objet construit sur l'ensemble considéré. Puisque pour un objet de départ de taille donnée il y a toujours le même nombre d'objet pointé correspondant, la méthode ne modifie pas la distribution des objets pour une taille donnée (qui reste donc uniforme), L'opération de pointage a pour effet de briser la symétrie de la génération et de transformer des distributions piquées en distributions plates.

Pour une classe combinatoire donnée $\mathcal{C}$, on définit la classe pointée comme :

$$\mathcal{C}^\bullet \underset{n\to\infty}{\sim} \{(\gamma, i) \mid \gamma \in \mathcal{C}, i \in \{1, \ldots, |\gamma|\}\}.$$

On en déduit que $|\mathcal{C}_n^\bullet| = n|\mathcal{C}_n|$, et la fonction génératrice associée est $C^\bullet(z) = zC'(z)$. En effet, si $\mathcal{C}$ a comme exposant singulier $-\alpha$, alors $\mathcal{C}^\bullet$ a comme exposant singulier $-\alpha - 1$.

Supposons maintenant que $\mathcal{C}$ est une classe combinatoire d'exposant singulier $-\alpha > 0$. En marquant $\lceil \alpha \rceil$ atomes de la structure considérée on obtient une distribution plate sur les structures multiplement marquées. Donc les Théorèmes 5.1 et 5.2 peuvent s'appliquer sur ces structures multiplement marquées. Lorsqu'un objet est généré, on efface toutes ses marques puis on obtient un objet de $\mathcal{C}$ uniformément au hasard parmi les objets de la même taille. La classe des arbres binaires marqués (5.1) est un exemple de classe marquée.

On peut noter que bien que la différenciation de certaines spécifications puisse donner des spécifications beaucoup plus larges, cela n'interfère pas avec notre mesure de complexité.

5.4.2 Générateur de Boltzmann singulier

L'autre méthode pour gérer le cas lorsque $0 < -\alpha < 1$ consiste à éviter le calcul du paramètre en choisissant la singularité en tant que paramètre, ce qui est raisonnable puisque la distribution de la taille a une grande queue.

La taille moyenne d'un objet généré selon cette distribution est ici infinie, mais en utilisant le rejet anticipé on s'assure un temps de génération fini.

Théorème 5.3. *Soit $\mathcal{C}$ une classe combinatoire dont la fonction génératrice est Δ-singulière avec un exposant $0 < -\alpha < 1$. Alors la taille cumulée T_n des objets générés puis rejetés par un générateur de Boltzmann approché avec rejet anticipé, $\nu C(\rho, n, \varepsilon)$, satisfait :*

$$\mathbb{E}(T_n) \underset{n\to\infty}{\sim} n\kappa_s(\varepsilon,\alpha), \qquad \text{et} \qquad \mathbb{E}(T_n^2) \underset{n\to\infty}{\sim} 2\mathbb{E}(T_n)^2 + n^2\bar{\kappa}_s(\varepsilon,\alpha),$$

où

$$\kappa_s(\varepsilon,\alpha) = \frac{-\alpha\cdot\left(\frac{(1-\varepsilon)^{\alpha+1}}{\alpha+1} + \frac{(1+\varepsilon)^{\alpha+1})}{-\alpha}\right)}{(1-\varepsilon)^\alpha - (1+\varepsilon)^\alpha},$$

et

$$\bar{\kappa}_s(\varepsilon,\alpha) = \frac{-\alpha\cdot\left(\frac{(1-\varepsilon)^{\alpha+2}}{\alpha+2} + \frac{(1+\varepsilon)^{\alpha+1})}{-\alpha}\right)}{(1-\varepsilon)^\alpha - (1+\varepsilon)^\alpha}.$$

La preuve du résultat est une conséquence du Lemme 5.1, comme pour les deux générateurs précédents. Par contre, puisque le paramètre a une expression très simple (c'est la singularité dominante), les équivalents peuvent être explicitement calculés.

Démonstration. Pour pouvoir utiliser le Lemme 5.1, on a besoin des estimations suivantes, obtenues par des sommations de Euler-MacLaurin sur les coefficients de $C(z)$.

$$\rho C'^{<n(1-\varepsilon)}(\rho) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+1}(1-\varepsilon)^{\alpha+1}}{(\alpha+1)\Gamma(\alpha)}, \qquad C^{>n(1+\varepsilon)}(\rho) \underset{n\to\infty}{\sim} -\frac{c_0 n^\alpha(1+\varepsilon)^\alpha}{\alpha\Gamma(\alpha)},$$

$$C^{[n(1-\varepsilon),n(1+\varepsilon)]}(\rho) \underset{n\to\infty}{\sim} \frac{c_0 n^\alpha}{\Gamma(\alpha)}\frac{(1+\varepsilon)^\alpha - (1-\varepsilon)^\alpha}{\alpha}$$

et

$$\rho^2 C''^{<n(1-\varepsilon)}(\rho) \underset{n\to\infty}{\sim} \frac{c_0 n^{\alpha+2}(1-\varepsilon)^{\alpha+2}}{(\alpha+2)\Gamma(\alpha)}.$$

□

5.5 Comparaison des différents méthodes pour les distributions piquées

Le but de cette section consiste à comparer les différentes méthodes par rapport à nos mesures de complexité définies plus tôt. L'approche la plus facile (déjà présente dans [20]) consiste à laisser ε tendre vers 0, puis de comparer le comportement des complexités.

5.5.1 Une classe d'objet générale avec $0 < -\alpha < 1$

On est intéressé par des objets provenant de classes combinatoires dont l'exposant singulier $-\alpha$ appartient à $]0,1[$. En pointant un élément, on peut utiliser le générateur de Boltzmann classique (Théorèmes 5.1 et 5.2) pour l'exposant $-\alpha - 1$, ou on peut directement utiliser un générateur de Boltzmann singulier. On regroupe les résultats dans la Figure 5.2 ci-dessous.

À la lumière de cette étude, on observe que les différentes optimisations pour la génération de Boltzmann présentées dans l'article initial [20] sont efficaces, et ne diffèrent que d'un facteur multiplicatif constant.

Si l'on s'intéresse à la génération exacte (où on rejette tant que la taille n'est pas exactement la bonne), les résultats de la Figure 5.2 sont encore valides pour $\varepsilon = \frac{1}{2n}$. Ce n'est pas une conséquence directe des théorèmes, mais peut être prouvé sans effort en suivant un plan de preuve identique. On obtient ainsi des générateurs de complexité quadratique.

	Complexité moyenne exacte
Générateur de B. pointé sans rejet anticipé	$n \cdot \dfrac{\int_{w=0}^{1-\varepsilon} w^{\alpha+1} e^{(-\alpha-1)w} dw + \int_{w=1+\varepsilon}^{\infty} w^{\alpha+1} e^{(-\alpha-1)w} dw}{\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha} e^{(-\alpha-1)w} dw}$
Générateur de B. pointé avec rejet anticipé	$n \cdot \dfrac{\int_{w=0}^{1-\varepsilon} w^{\alpha+1} e^{(-\alpha-1)w} dw + \int_{w=1+\varepsilon}^{\infty} (1+\varepsilon)\ w^{\alpha} e^{(-\alpha-1)w} dw}{\int_{w=1-\varepsilon}^{1+\varepsilon} w^{\alpha} e^{(-\alpha-1)w} dw}$
Générateur de B. singulier	$-\alpha n \left(\frac{(1-\varepsilon)^{\alpha+1}}{\alpha+1} + \frac{(1+\varepsilon)^{\alpha+1})}{-\alpha} \right) \left((1-\varepsilon)^{\alpha} - (1+\varepsilon)^{\alpha} \right)^{-1}$

	Complexité moyenne approchée ($\varepsilon \to 0$ et $\frac{1}{\varepsilon} = o(n)$)
Générateur de B. pointé sans rejet anticipé	$\frac{n}{2\varepsilon} \left(\left(\frac{e}{\alpha+1} \right)^{\alpha+1} \cdot \Gamma(\alpha+1) \right) + o\left(\frac{n}{\varepsilon}\right)$
Générateur de B. pointé avec rejet anticipé	$\frac{n}{2\varepsilon} \left(\left(\frac{e}{\alpha+1} \right)^{\alpha+1} \cdot \Gamma(\alpha+1) - \frac{1}{\alpha+1} \right) + o\left(\frac{n}{\varepsilon}\right)$
Générateur de B. singulier	$\frac{n}{2\varepsilon} \cdot \frac{1}{-\alpha(\alpha+1)} + o\left(\frac{n}{\varepsilon}\right)$

FIGURE 5.2 – Équivalent de la taille cumulée des objets générés puis rejetés, quand $0 < -\alpha < 1$.

5.5.2 Grammaires apériodiques et fortement connectées : $-\alpha = 1/2$

Les complexitées approchés pour les générateurs de Boltzmann avaient déjà été calculées [20] dans le cas de grammaires apériodiques et fortement connectées, comme par exemple dans le cas des arbres, mais sans comparaison des méthodes.

En prenant $-\alpha = 1/2$ dans les formules précédentes quand ε tends vers 0 (avec la contrainte $\frac{1}{\varepsilon} = o(n)$), des calculs directs donnent les résultats suivants :

— Dans le cas des générateurs de Boltzmann pointés sans rejet anticipé, on obtient une complexité moyenne équivalente à :

$$\left(\sqrt{\frac{\pi e}{2}}\right) \cdot \frac{n}{\varepsilon}, \qquad \text{avec } \sqrt{\frac{\pi e}{2}} \approx 2.066.$$

— Dans cas du générateur de Boltzmann pointés avec rejet anticipé, on obtient une complexité moyenne équivalente à :

$$\left(\sqrt{\frac{\pi e}{2}} - 1\right) \cdot \frac{n}{\varepsilon}, \qquad \text{avec } \sqrt{\frac{\pi e}{2}} - 1 \approx 1.066.$$

— Dans le cas de générateur de Boltzmann singulier, on obtient une complexité moyenne équivalente à :

$$2 \cdot \frac{n}{\varepsilon}.$$

On peut finalement remarquer que le pointage sans rejet anticipé est à peu près équivalent à la génération singulière, mais que la génération pointée avec rejet anticipé est presque deux fois plus efficace [6] !

6. Il faut se souvenir que notre complexité ne prend pas en compte le terme linéaire qui provient de l'objet généré, ce qui mitige le facteur 2 annoncé.

5.5.3 Comparaison expérimentale des différents générateurs

Prenons un exemple de grammaire apériodique et fortement connectée, les arbres binaires où la taille correspond au nombre de nœuds internes. On rappelle que sa fonction génératrice ordinaire correspond à la série des nombres de Catalan et est :

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

Son exposant singulier est donc $-\alpha = 1/2$, et la distribution correspondante est piquée.

Dans la Figure 5.1, la distribution piquée en lignes hachées bleues est exactement celle des arbres binaires complets avec paramètre de Boltzmann $z \in \{0.15, 0.24, 0.249\}$.

Notre but est d'observer expérimentalement si la mesure de complexité des générateurs se comporte de façon suffisamment proche comme celle asymptotique pour des petits arbres (quelques milliers de nœuds).

On peut donner quelques propriétés sur la distribution en taille. Tout d'abord avec le paramètre $z = 0.249$, assez proche de la singularité dominante pour $\mathcal{B}$, L'arbre de taille 0 (juste une feuille) a une probabilité approximativement de 0.532 d'être tiré. En tirant dans $\mathcal{B}^\bullet$ au lieu de $\mathcal{B}$, avec le même paramètre on obtient des arbres de taille moyenne de 74 au lieu de 1, avec une probabilité proche de 0.501. En prenant un paramètre plus proche de la singularité, la distribution piquée ne change pas beaucoup, tandis que pour les distribution plates on obtient des arbres de plus en plus gros. Pour $z = 0.249999$ on génère des arbres de taille entre 1 et 3134, et pour $z = 0.2499999$ on obtient avec probabilité 1/2 des arbres de taille dans $[1, 570351]$.

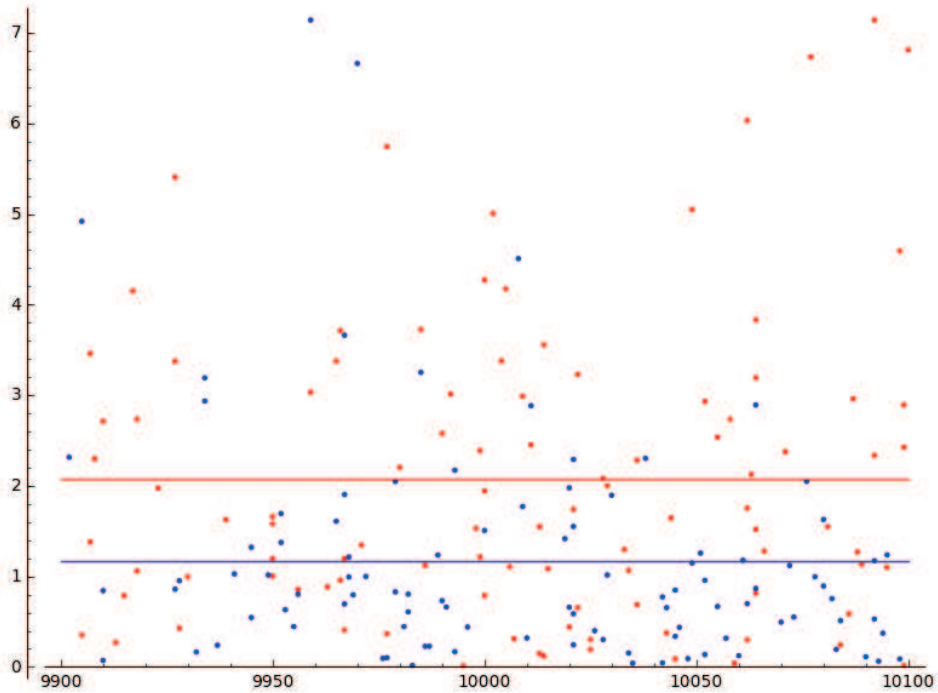


FIGURE 5.3 – Graphe d'arbres de taille $(10,000 \pm 1\%)$ et leur complexité normalisée correspondante.

Pour les différentes expériences, on a généré une centaine d'arbres binaires de taille $10,000 \pm 1\%$, $100,000 \pm 10\%$ et $100,000 \pm 1\%$. Le ratio n/ε est le même pour les deux premiers ensembles d'arbres, et plus petit pour le suivant.

Dans les Figures 5.3, 5.4 et 5.5, In Figures 5.3, 5.4 et 5.5, on représente la complexité normalisée (i.e., la complexité multipliée par n/ε) par rapport à la taille des arbres générés.

Les arbres représentés avec des points rouges ont été tirés avec un générateur de Boltzmann singulier. Dans le premier graphe (taille des arbres d'à peu près 10.000), la complexité normalisée

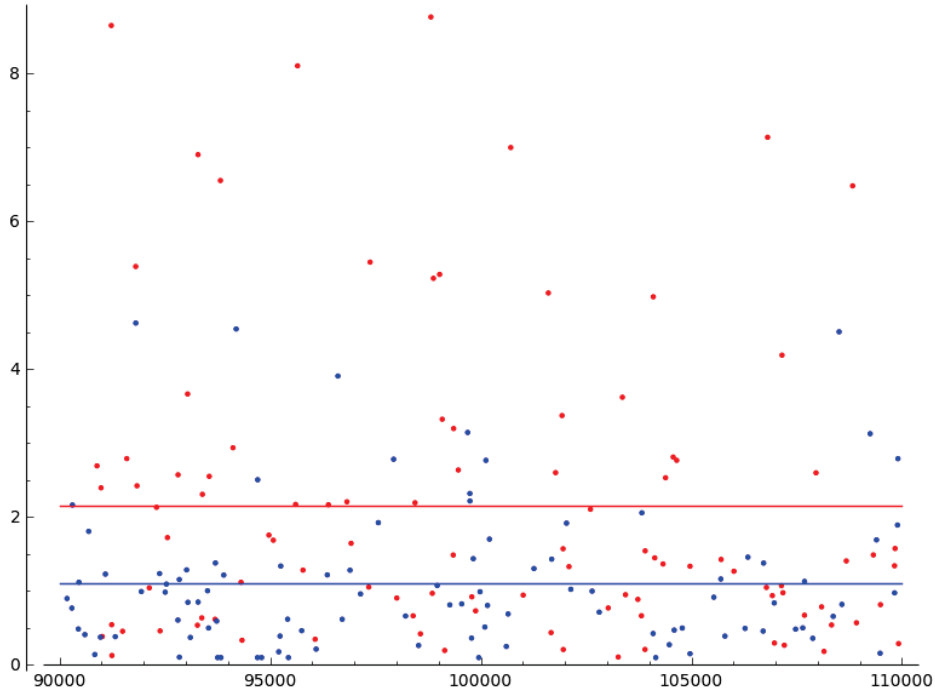


FIGURE 5.4 – Graphe d'arbres de taille $(100,000 \pm 10\%)$ et leur complexité normalisée correspondante.

moyenne (représentée par la ligne rouge) est approximativement égale à 2.1072. Pour les arbres de taille environ $100,000$ la valeur normalisée moyenne est approximativement 2.1464 (pour $\varepsilon = 0.1$) et 2.1330 (pour $\varepsilon = 0.01$).

Pour la génération pointée avec rejet anticipé (croix bleues et ligne bleue en pointillés pour la valeur moyenne), les valeurs normalisées sont respectivement 1.1710, 1.1055 et 1.0673.

En étudiant les graphes des Figures 5.3, 5.4 et 5.5, on remarque que la différence de complexité d'une génération à une autre peut être assez grande. Par contre en générant quelques centaines d'arbres les valeurs moyennes sont "proches" des valeurs théoriques.

5.6 Conclusion

Nous avons montré dans ce chapitre les différences entre deux méthodes classiques de génération uniforme de structures, le pointage et la méthode de la singularité. La conclusion est que la complexité des deux méthodes ne diffère que d'un facteur constant, la plus efficace pour la complexité comptabilisée étant de manière prévisible la méthode du pointage. Ce facteur dépend du type de structure, mais il est compris entre 1.3 et 10 pour des structures raisonnables ($0 < -\alpha < 0.94$). On en conclut donc que la méthode du pointage est à préférer si l'on désire la meilleure complexité, mais que la méthode de la singularité a sa place si d'autres facteurs rendent la rend plus attractive (comme par exemple une évaluation plus simple des paramètres).

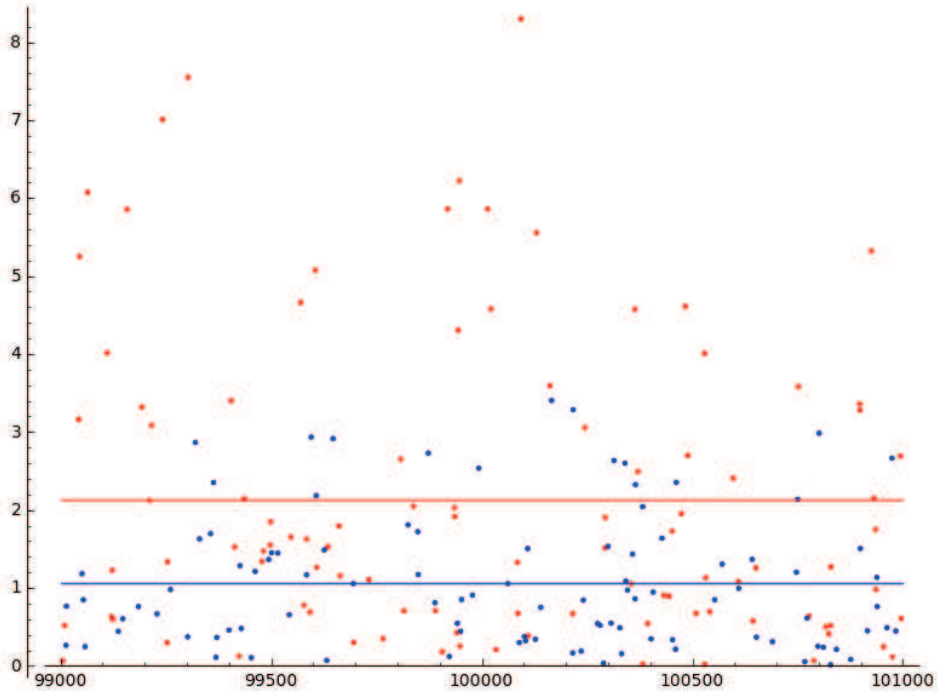


FIGURE 5.5 – Graphe d'arbres de taille $(100,000 \pm 1\%)$ et leur complexité normalisée correspondante.

Chapitre 6

Génération de Boltzmann mortelle

Ce chapitre décrit une extension des générateurs de Boltzmann où l'on se munit d'un rejet additionnel (*la mort*) qui permet de générer uniformément des objets en utilisant des approximations. L'objectif est donc d'avoir une méthode efficace pour calculer les oracles de la génération de Boltzmann (les valeurs des paramètres), sans avoir à calculer des valeurs réelles (symboliques) et se contenter de valeurs rationnelles. On s'évite aussi des implémentations potentiellement lourdes pour conserver l'uniformité de la génération, puisque dans le cas de valeurs réelles ont doit parfois calculer de nouvelles décimales de nos valeurs d'oracles à la volée pendent la génération.

Le modèle de Boltzmann On rappelle (cf. 0.3.2) la définition de la distribution de Boltzmann : Avec un générateur de Boltzmann pour une classe combinatoire non étiquetée $\mathcal{C}$, pour laquelle il y a c_n éléments de taille n, la probabilité de tirer un objet $\gamma \in \mathcal{C}$ est :

$$\mathbb{P}_z[\gamma] = \frac{z^{|\gamma|}}{C(z)} \qquad \text{avec} \qquad C(z) := \sum_{n=1}^{\infty} c_n z^n \ = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}, \tag{6.1}$$

où $|\gamma|$ est la taille de l'objet γ et x est un paramètre de contrôle à choisir. Donc la probabilité de tirer un objet de taille n est

$$\mathbb{P}_z[|\gamma| = n] = \frac{c_n z^n}{C(z)} \qquad\qquad \mathbb{P}_s[\gamma \mid \ |\gamma| = n] = \frac{1}{c_n},$$

tandis que la probabilité de tirer un objet sachant sa taille donnée est uniforme.

Le nom de la méthode évoque le modèle de Boltzmann de physique statistique, qui assigne à chaque état possible d'un système une probabilité $\mathrm{e}^{-\beta E}/Z$, où E est l'énergie de l'état, $\beta = 1/T$ est une constante, et Z est une constante normalisante.

Bien que la distribution des tailles des objets soit une distribution très générique déjà connue par les probabilistes en tant que *distribution de séries entières*[1] d'après Nelson [33], la terminologie est souvent créditée à Noack [44] en 1950.

Évaluer des fonctions génératrices proches de la singularité. Les générateurs de Boltzmann dépendent fortement de la détermination de l'évaluation d'une fonction génératrice sur un paramètre qui approche la singularité, et ce en complexité arithmétique linéaire. Le problème de savoir exactement comment évaluer ses fonctions était laissé ouvert dans l'article originel, et était sans réponse convaincante jusqu'à l'article de Pivoteau *et al.* [46, 47]. Ils ont introduit une variante de l'approximation de Newton sur des systèmes combinatoires, qui possède une convergence quadratique, ce qui suffit pour toute application pratique.

1. Les distributions de Poisson, géométrique, et log-séries sont toutes des cas particuliers de cette distribution, un fait qui est utilisé par Flajolet *et al.* [23] qui a créé l'algorithme de Von Neumann/Flajolet pour simuler des distributions de séries entières en utilisant uniquement des bits aléatoires.

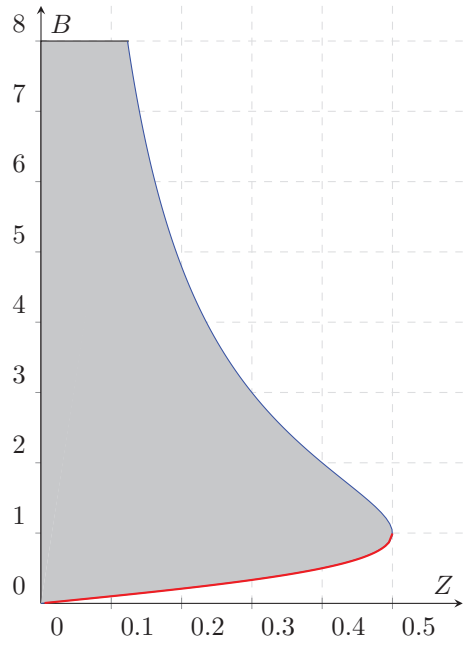


FIGURE 6.1 – Figure associée avec la spécification des arbres binaires où tous les nœuds sont comptés, $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B}^2$. La courbe rouge en bas décrit $B(z) = z + zB(z)^2$, où toutes les coordonnés $(z, B(z))$ sont généralement considérées pour la génération de Boltzmann. La région grisée est la région vérifiant $b \geqslant z + zb^2$, et dans laquelle se trouve les coordonnées (z, b) que nous allons utiliser dans notre modèle modifié.

La contribution : un cadre pratique et extensible

L'idée ici est de faire appel au concept de *rejet*, classique en génération aléatoire (voir par exemple le chapitre de Devroye sur la méthode de rejet [17, §2]). Plutôt que d'évaluer des fonctions génératrices directement, on choisit un point avoisinant qui est plus facile à calculer.

Cela s'illustre simplement sur la Figure 6.1. Les générateurs de Boltzmann classiques se limitent aux coordonnées qui se situent sur la courbe rouge en bas de la région grisée, notre modèle modifié nous permet de choisir n'importe que point dans cette région. Bien sûr, cela coûte un prix additionnel de rejet, mais nous allons montrer que ce rejet est constant et que pour des choix de coordonnées raisonnables il devient négligeable en pratique.

En ne se restreignant pas à une courbe fixée, on se donne plus de latitude pour choisir les points. Par exemple, les nombres rationnels engendrent des probabilités qui sont plus faciles à simuler exactement et efficacement [37], et sont plus pratiques à manipuler exactement. Puisque les nombres rationnels sont denses dans les nombres réels, on va pouvoir sans problème se limiter uniquement à des nombres rationnels dans nos générateurs, en utilisant une approximation de suites de Farey, et cette liberté d'action nous est apportée par le rejet.

Intuitivement, cette méthode correspond à ajouter des objets que l'on va automatiquement jeter dans nos spécifications pour retourner à la courbe et à une génération de Boltzmann classique.

6.1 Générateurs analytiques

Dans cette section nous donnons les principales définitions pour nos générateurs aléatoires analytiques, puis nous montrons les algorithmes associés avec les constructions basiques.

6.1.1 Définitions principales

Définition 6.1. *Soit $\mathcal{A}$ une classe combinatoire définie symboliquement qui peut être traduite, suivant la méthode symbolique, en une équation fonctionnelle sur $A(z)$, la fonction génératrice associée*

avec $\mathcal{A}$,

$$\mathcal{A} = \Phi(\mathcal{Z}, \mathcal{A}, \boldsymbol{\mathcal{X}}) \qquad \Rightarrow \qquad A(z) = \phi(z, A(z), \boldsymbol{X}(z)),$$

où à la fois Φ *et* ϕ *peuvent possiblement impliquer d'autres classes/fonctions génératrices qui sont notés en gras (et chaque symbole/fonction génératrice du vecteur étant eux-même définis par leurs propres équations). Une paire de coordonnées* (z, a) *est dite* analytiquement valide *pour une classe combinatoire* $\mathcal{A}$ *si et seulement si elle vérifie l'inégalité*

$$a \geqslant \phi(z, a, \boldsymbol{X}(z)).$$

En général, par soucis de clarté de praticité, on omettra le vecteur dans les notations, et tous les symboles liées additionnels seront implicites. Si l'on reprend l'exemple de la Figure 6.1, les coordonnées analytiquement valides sont les points dans la région grisée.

Définition 6.2. *Un* générateur analytique *pour une classe combinatoire non étiquetée* $\mathcal{A}$ *est un algorithme qui tire un objet* $\alpha \in \mathcal{A}$ *de taille* $|\alpha|$ *avec probabilité*

$$\mathbb{P}_{(z,a)}[\alpha] = \frac{z^{|\alpha|}}{a} \qquad \textit{et échoue avec probabilité} \qquad \mathbb{P}_{(z,a)}[\dagger] = 1 - \frac{A(z)}{a}$$

où $A(z)$ *est la fonction génératrice ordinaire associée avec la classe* $\mathcal{A}$*, et les coordonnées analytiques valides* (z, a) *sont appelées les* paramètres de contrôle. *On note* $\Gamma_{(z,a)}[\mathcal{A}]$ *ce générateur analytique.*

Les générateurs de Boltzmann classiques utilisant déjà, un concept de rejet pour contrôler la taille de la sortie et la contraindre dans un intervalle de tolérance, on choisit de nommer le rejet associé aux générateurs analytiques *mort*, pour éviter toute ambiguïté. Un générateur de Boltzmann analytique va donc générer un objet à l'aide d'un générateur analytique, et si l'objet meurt, en relance un autre jusqu'à avoir un objet qui survit. Il est donc intéressant d'étudier en moyenne combien de générateurs analytiques on doit lancer avant de réussir à générer un objet.

Théorème 6.1. *Soit* $\mathcal{A}$ *une classe combinatoire et* $A(z)$ *sa fonction génératrice, et soit* (z, a) *des coordonnées analytiquement valides pour* $\mathcal{A}$*. La proportion des objets pour qui la génération conduit à la* mort *ne dépends pas de la taille de l'objet sorti à la fin, et est égale à* $1 - A(z)/a$.

Démonstration. Cela se déduit de la définition du modèle des générateurs analytiques, où la probabilité de mort d'un tirage est constante—dans le sens où la probabilité ne dépends pas de la taille de l'objet qui était construit quand l'objet généré est mort—et est égale à

$$\mathbb{P}_{(z,a)}[\dagger] = 1 - \frac{A(z)}{a},$$

et donc un objet de taille aléatoire est tiré avec la probabilité complémentaire. Le nombre d'objets morts avant qu'un objet qui survive soit tiré est alors géométriquement distribué avec $p = 1 - A(z)/a$. On a alors :

$$\begin{aligned}\mathbb{E}_{(z,a)}[\#\dagger] &= \sum_{k=0}^{\infty} k \left(1 - \frac{A(z)}{a}\right)^k \frac{A(z)}{a} = \frac{A(z)}{a} \frac{\left(1 - \frac{A(z)}{a}\right)}{\left(\frac{A(z)}{a}\right)^2} \\ &= \frac{a}{A(z)} \left(1 - \frac{A(z)}{a}\right) = \frac{a}{A(z)} - 1\end{aligned}$$

et puisqu'il y a un dernier objet généré (celui qui ne meurt pas), la proportion du nombre d'objets mort générée est $\mathbb{E}_{(z,a)}[\#\dagger] / (\mathbb{E}_{(z,a)}[\#\dagger] + 1)$. □

Il faut noter que la valeur minimale pour a est $a = A(z)$, et que pour ce choix le générateur analytique a une mortalité nulle ; en effet l'inégalité peut naturellement se voir comme une égalité impliquant une variable δ, $a = \phi(z, a, \boldsymbol{x}) + \delta$, où δ est la proportion d'échecs. L'idée est que si on passe du temps de calcul nécessaire pour calculer la fonction génératrice, on est récompensé par une absence de morts.

Exemple 6.1. *On va maintenant illustrer ces définitions en regardant la classe $\mathcal{B}$ des arbres binaires, dans laquelle chaque nœud à la fois interne et externe*[2] *compte dans la taille de l'arbre. Ces arbres peuvent être spécifiés symboliquement comme soit une feuille ($\mathcal{Z}$) soit comme un nœud où sont attachés deux sous-arbres ($\mathcal{Z} \times \mathcal{B}^2$),*

$$\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B}^2 \qquad \text{et} \qquad B(z) = z + zB(z)^2. \tag{6.2}$$

L'équation fonctionnelle correspond à une inégalité ,

$$b \geqslant z + zb^2.$$

Les coordonnées analytiquement valides pour $\mathcal{B}$ sont tous les points appartenant à la région grisée dans la Figure 6.1. On peut remarquer de plus que la probabilité de mourir dépend directement de l'écartement vertical[3] *du point à la courbe rouge.*

6.1.2 Constructions

Dans cette partie, on donne les constructions de base utilisés par les générateurs analytiques. Les notations suivent celles de l'article original [20], qui sont étendues pour inclure une probabilité de mort. La notation

$$\Gamma\mathcal{A} : [p_1] \cdot \mathrm{Ber}(p_2) \Rightarrow X \mid Y$$

signifie que l'on meurt avec probabilité $1-p_1$, puis on tire une variable de Bernoulli U avec paramètre p_2, si $U = 1$ alors on retourne X, sinon on retourne Y. Si à la place d'une distribution de Bernoulli on a une distribution discrète K, cela veut dire que l'on retourne un tuplet de K appels indépendants au générateur.

Soit $\mathcal{A}$, $\mathcal{B}$ et $\mathcal{C}$ des classes combinatoires. On rappelle que l'on note $\mathbb{P}[\alpha]$ la probabilité de tirer un objet α ; lorsque l'on veut que la classe dans lequel il est tiré explicite, on note $\mathbb{P}[\alpha \in \mathcal{A}]$

Union disjointe. Soit $\mathcal{A} = \mathcal{B} + \mathcal{C}$ et $a \geqslant b + c$. On commence par tuer l'objet avec probabilité $1 - (b + c)/a$, puis on fait une génération de Boltzmann normale.

$$\Gamma\mathcal{A} : \left[\frac{b+c}{a}\right] \cdot \mathrm{Ber}\left(\frac{b}{b+c}\right) \Rightarrow \Gamma\mathcal{B} \ \mid \ \Gamma\mathcal{C} \tag{6.3}$$

Démonstration. On doit montrer que le générateur $\Gamma\mathcal{A}$ retourne des objets $\alpha \in \mathcal{A}$ avec la bonne probabilité $\mathbb{P}_{(z,a)}[\alpha] = z^{|\alpha|}/a$ (c'est à dire la probabilité de tirer un objet de $\mathcal{A}$ suivant la définition 6.2), en supposant récursivement que les générateurs $\Gamma\mathcal{B}$ et $\Gamma\mathcal{C}$ sont corrects. Supposons que $\alpha \in \mathcal{B}$, alors :

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b+c}{a}\left(\frac{b}{b+c}\mathbb{P}_{(z,b)}[\alpha \in \mathcal{B}]\right)$$

C'est à dire que la probabilité de tirer un objet de $\mathcal{A}$ est d'abord la probabilité de ne pas mourir, $(b + c)/a$, puis la probabilité de tirer un objet en utilisant un générateur de la classe $\mathcal{B}$ avec la probabilité appropriée. Par hypothèse ces deux générateurs retournent des objets avec la bonne probabilité, donc on peut calculer la probabilité de tirer un objet qui vient de $\mathcal{B}$ et qui, puisque l'union est *disjointe*, ne peut venir de $\mathcal{C}$:

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{B}] = \frac{b+c}{a}\left(\frac{b}{b+c}\frac{z^{|\alpha|}}{b}\right) = \frac{z^{|\alpha|}}{a}.$$

On trouve par symétrie le même résultat sur les objets provenant de $\mathcal{C}$. Notons que dans cette preuve, et la suivante, on ne prouve pas explicitement la probabilité de mourir puisque c'est une conséquence directe de la probabilité de tirer un objet, en prenant le complémentaire de la somme des probabilités de tirer chaque objet. □

2. C'est une légère variation des arbres binaires vus dans les autres chapitres, qui ne comptaient pas les nœuds externes.

3. La probabilité de mourir est d'ailleurs simplement le rapport entre la distance à la courbe et l'ordonnée du point.

Produit cartésien. Soit $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, et $a \geqslant b \cdot c$.

$$\Gamma\mathcal{A} : \left[\frac{b \cdot c}{a}\right] \Rightarrow (\Gamma\mathcal{B} \ ; \ \Gamma\mathcal{C}) \tag{6.4}$$

Démonstration. La preuve suit le même modèle que pour les précédentes constructions ; soit $\alpha = (\beta, \gamma)$,

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b \cdot c}{a} \mathbb{P}_{(z,b)}[\beta \in \mathcal{B}]\, \mathbb{P}_{(z,c)}[\gamma \in \mathcal{C}]$$

et puisque les générateurs pour $\Gamma\mathcal{B}$ et $\Gamma\mathcal{C}$ sont récursivement supposés corrects :

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{b \cdot c}{a} \frac{z^{|\beta|}}{b} \frac{z^{|\gamma|}}{c} = \frac{z^{|\beta|+|\gamma|}}{a} = \frac{z^{|\alpha|}}{a}.$$

□

Séquence. Soit $\mathcal{A} = \mathrm{Seq}(\mathcal{B})$ et $a \geqslant 1/(1-b)$.

$$\Gamma\mathcal{A} : \left[\frac{(1-b)^{-1}}{a}\right] \cdot \mathrm{Geo}(b) \Rightarrow (\Gamma\mathcal{B}, \ldots) \tag{6.5}$$

Démonstration. On suit le même modèle que précédemment. Soit $\alpha = (\beta_1, \ldots, \beta_k)$.

$$\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] = \frac{(1-b)^{-1}}{a} \mathbb{P}[\mathrm{Geo}(b) = k] \prod_{i=1}^{k} \mathbb{P}_{(z,b)}[\beta_i \in \mathcal{B}]$$

et par hypothèse :

$$\begin{aligned}
\mathbb{P}_{(z,a)}[\alpha \in \mathcal{A}] &= \frac{(1-b)^{-1}}{a} b^k (1-b) \prod_{i=1}^{k} \frac{z^{|\beta_i|}}{b} \\
&= \frac{(1-b)^{-1}}{a} b^k (1-b) \frac{z^{|\beta_1|+\ldots+|\beta_k|}}{b^k} \\
&= \frac{z^{|\beta_1|+\ldots+|\beta_k|}}{a} = \frac{z^{|\alpha|}}{a}.
\end{aligned}$$

□

6.1.3 Illustration du taux de "mortalité" avec les arbres de Cayley

On va donner ici un exemple un peu plus intéressant que les arbres binaires planaires, on va définir les objets étiquetés, et utiliser l'opérateur Set, qui est présenté dans le cas classique dans l'article[20], en utilisant une loi de poisson notée $\mathrm{Pois}(t)$. Cela va nous servir à montrer comment s'éloigner de la courbe influe sur le taux de mortalité, c'est-à-dire le rejet qui doit être fait pour compenser l'approximation.

Considérons l'exemple de la classe $\mathcal{T}$ des arbres de Cayley (étiquetés, non-planaires et d'arité non-bornée), spécifiée par l'équation :

$$\mathcal{T} = \mathcal{Z} \star \mathrm{Set}\, \mathcal{T}.$$

Sa fonction génératrice exponentielle, $T(z) = ze^{T(z)}$, est proche de la fonction W de Lambert (utilisée pour la proposition 1.2.1), qui est implicitement définie.

Avec un générateur analytique, le point de départ est le système d'équations fonctionnelles donné par la méthode symbolique. On remplace ensuite chaque fonction par une variable libre, et on obtient l'inégalité $t \geq z \cdot \exp(t)$ et l'algorithme :

$$\Gamma\mathcal{T}(z,t) : \left[\frac{z \exp(t)}{t}\right] \cdot \mathrm{Pois}(t) \Rightarrow \Box(\Gamma\mathcal{T}(z,t), \ldots, \Gamma\mathcal{T}(z,t))$$

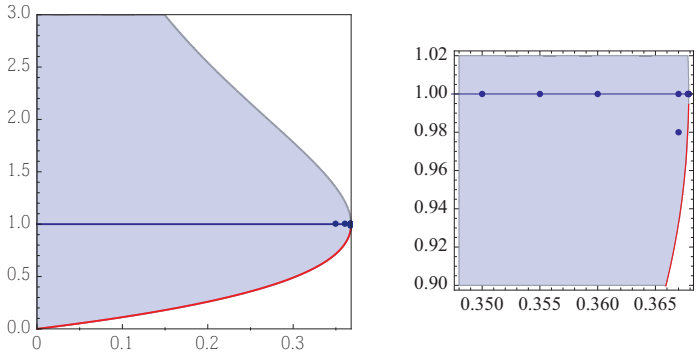


FIGURE 6.2 – C'est un graphe de la région définie par l'inégalité $t \geq z \cdot e^t$, avec z sur l'axe des abscisses et t sur l'axe des ordonnées. La frontière basse de la région, en rouge et en gras, est la courbe de la fonction génératrice $T(z)$.

				$t = 1$				$t = 0.98$
z =	0.35	0.36	0.367	0.3678	0.36787	0.367879	e^{-1}	0.367
Mort (observée)	28.8%	19.2%	6.4%	1.7%	0.4%	0.3%	0%	3.5%
Mort (théorique)	28.3%	19.4%	6.8%	2.1%	0.7%	0.2%	0%	3.9%
Taille moyenne	6.6	9.9	28.8	127.	177.3	2716.7	4944.3	35.9
Taille maximale	235	131	1493	17 799	26 531	826 167	2 518 975	1563

TABLE 6.1 – Ce tableau contient le résultat de 1000 appels à des générateurs analytiques pour des arbres de Cayley, avec diverses valeurs de z (la paramètre de contrôle) et t (l'approximation la fonction génératrice).

Après le rejet initial (que l'on appelle mort) avec probabilité $z \cdot \exp(t)/t$, pour prendre en compte l'approximation que l'on a fait de la fonction génératrice, on tire une variable de Poisson de paramètre t, pour indiquer le nombre d'enfants à générer. On voit assez directement que cet algorithme est correct pour toute paire (z, t) qui satisfait l'inégalité précédente. On remarque aussi que la probabilité de rejet peut facilement être simulée exactement en utilisant les techniques décrites par Flajolet *et al.* [23].

Plus remarquablement, une expérience résumée dans le tableau 6.1 montre que l'impact de l'approximation est modeste. Pour diverses paires (z, t), le tableau montre le résultat de 1000 appels au générateur : il indique la proportion des générations qui échoue prématurément ; et prend note de la taille moyenne et maximum parmi les arbres tirés. Le cas où $z = \exp(-1)$ et $t = 1 = T(z)$ est spécial : en premier parce que c'est le seul cas où t est exactement égal à la fonction génératrice évalué (ainsi la mortalité est de 0% et le générateur analytique est un générateur de Boltzmann classique) ; et en deuxième parce que puisqu'on l'évalue en sa singularité, il s'agit d'un *générateur de Boltzmann singulier* (pour lequel l'espérance de la taille est non bornée). Tous les autres points, ainsi qu'on le voit sur la Figure 6.2, sont plus ou moins distants de la courbe de $T(z)$, avec un taux de mortalité beaucoup plus élevé. Mais même à une distance importante de la courbe, le taux de mortalité reste tolérable.

La *mort* est la proportion d'arbres qui doivent être tués à cause du choix d'approximer de la fonction génératrice au lieu de l'évaluer. Ainsi pour la paire de valeur $z = e^{-1}$ et $t = 1 = T(z)$, pour laquelle on utilise la valeur exacte de $T(z)$, nos générateurs sont exactement équivalents aux générateurs de Boltzmann, ainsi le taux de mortalité est 0%. Le taux de mort pour des approximations plutôt larges reste acceptable. On peut aussi remarquer dans ce tableau que l'on a deux paramètres indépendants : le taux de mortalité, qui peut se lire sur la Figure 6.2 comme étant la proportion de bleu qui sépare le point de l'axe des abscisses, et la valeur du paramètre z, qui définit à lui seul la taille moyenne des arbres obtenus. Ainsi les deux points à partager la valeur $z = 0.637$ ont la même taille moyenne et taille maximale (ce qui se traduit dans le tableau par des valeurs similaires), et seul leur taux de mortalité change.

6.2 Arbres simplement générés

On veut maintenant illustrer comment s'intéresser à une région (et une inéquation) peut rendre la recherche d'une paire de valeurs optimales pour le générateur plus simple.

Les arbres simplement générés ont été introduits par Meir et Moon [39] en tant que classes d'arbres définis par la spécification suivante.

$$\mathcal{Y} = \mathcal{Z} \times \Phi(\mathcal{Y}) \tag{6.6}$$

où Φ est un polynôme défini par

$$\Phi(w) = \sum_{\omega \in \Omega} w^\omega \qquad \textit{ou} \qquad \Phi(w) = \sum_{\omega \in \Omega} \frac{w^\omega}{\omega!} \tag{6.7}$$

respectivement, selon si la classe est étiquetée ou non étiquetée, et où $\Omega \subseteq \mathbb{N}$ est le multiensemble des degrés autorisés (par exemple, pour les arbres binaires, $\Omega = \{0, 2\}$). Meir and Moon ont identifié que les arbres définis de cette manière partageaient un grand nombre de propriétés communes, comme une longueur de cheminement moyenne de l'ordre de $n\sqrt{n}$ ou une hauteur moyenne de $\sqrt{n}$.

6.2.1 Approches existantes

Générer aléatoirement un élément de cette classe d'arbre n'est pas très compliqué : il y a plusieurs méthodes pour le faire, avec différentes propriétés d'optimalité (temps, nombre de bits aléatoires, etc.). On n'introduit ici donc pas des générateurs plus efficaces, mais cet exemple devrait montrer que la calibration peut être parfois plus pratique avec des générateurs analytiques.

Il se trouve que les arbres simplement générés ont une singularité de branchement. Cela signifie que leurs fonctions génératrices peuvent être évaluées à la singularité, et aussi que la distribution de la taille des objets produits par un générateur de Boltzmann est piquée, c'est-à-dire très concentrée sur des objets de petite taille. La solution est traditionnellement de faire de la génération singulière, et donc de prendre z étant égal à, ou proche de, la singularité, et générer des objets de taille moyenne non bornée, et rejeter ceux qui sont trop gros.

À part pour les cas simples (tels que les arbres binaires, pour qui la singularité est connue pour être $1/4$, la singularité n'est *pas* connue, donc elle doit être déterminée avec des méthodes souvent algorithmiques. On peut faire une dichotomie, suivant l'implémentation de Darrasse [16] : l'oracle de Pivoteau *et al.* [47] converge lorsqu'il est dans le rayon de convergence, il est du coup possible de déterminer lorsque l'on est allé au delà de la singularité. Cela demande un nombre logarithmique d'appels à l'oracle pour avoir une approximation de la singularité.

Notre méthode permet d'éviter d'avoir à calculer exactement cette singularité.

6.2.2 Générateurs analytiques : Maximiser un polynôme

De la spécification dans l'équation (6.6), on obtient la condition pour la validité analytique de la paire (z, y),

$$y \geqslant z \cdot \Phi(y). \tag{6.8}$$

Il est maintenant plus facile de ne pas regarder la fonction génératrice $Y(z)$, mais à la place la fonction $y \mapsto y/\Phi(y)$, qui est une fonction rationnelle. Cette fonction admet un point maximal dans l'intervalle unité, qui est le point singulier de $Y(z)$. On illustre cela sur la Figure 6.3 qui montre le graphe d'un arbre simplement généré (dans ce cas, la classe $\mathcal{U}$ des arbres binaires-unaires, i.e., $\Omega = \{0, 1, 2\}$). Sur la gauche, l'axe des abscisses est z, le paramètre de contrôle et l'axe des ordonnées est $U(z) = z\Phi(U(z))$. Sur la droite, on trace la fonction $u \mapsto u/\Phi(u)$. Le problème de trouver la singularité (à gauche) à été réduit au problème plus concret de maximiser une fonction (à droite).

Calculer ce point maximal est un problème beaucoup plus facile, qui ne requiert pas d'évaluations de la fonction génératrice (excepté peut-être pour une première tentative) ; cela peut être résolu par différentiation, par itération de Newton, ou avec un algorithme spécifiquement optimisé de la littérature, comme par exemple l'algorithme de Brent [12].

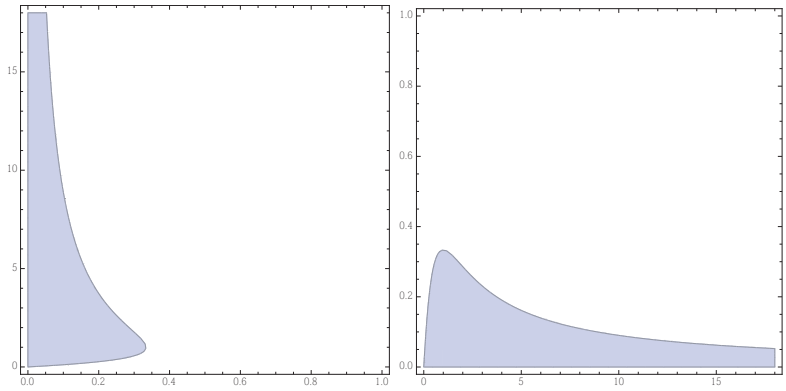


FIGURE 6.3 –
Graphe d'un arbre simplement généré

6.3 Constructions additionnelles

Bien que l'on ait montré comment construire les générateurs analytiques seulement pour les constructeurs élémentaires, les constructions peuvent être plus complexes. En particulier, les opérateurs fonctionnels tel que le pointage (différentiation) et la substitution (composition) peuvent être utilisés.

La substitution se révèle assez compliquée, mais nous allons comment on peut utiliser cette approche à partir de substitutions élémentaires. On présentera ensuite comment construire les paires non-ordonnées, MSet_2, et montrer une application avec la génération aléatoire d'arbres d'Otter.

6.3.1 Construction de réplications

On va montrer comment générer une réplication élémentaire. On pourra ensuite utiliser les opérations usuelles (union, produit, séquence,...) pour générer des constructions plus complexes.

Soit $\mathcal{B}$ une classe combinatoire, et $\mathcal{A} = \mathcal{S}_k[\mathcal{B}]$ la classe contenant les séquences de k éléments identiques de $\mathcal{B}$. Les fonctions génératrices correspondantes $A(z)$ et $B(z)$ vérifient l'équation fonctionnelle :

$$A(z) = B(z^k).$$

Supposant qu'il y a un générateur analytique pour $\mathcal{B}$, on peut construire un générateur analytique pour $\mathcal{A}$. Soit (z, b) une coordonnée analytique valides pour $\mathcal{B}$, et soit (z, a) des coordonnées analytiquement valides pour $\mathcal{A}$, c'est à dire

$$a \geq b.$$

En utilisant les notations précédentes,

$$\Gamma_{(z,a)}[\mathcal{A}] : \left[\frac{b}{a}\right] \Rightarrow \Gamma_{(z^k,b)}[\mathcal{B}] \text{ et en prendre } k \text{ copies.}$$

En d'autres termes, après avoir fait le test de mort, on fait un appel à $\Gamma_{(z^k,b)}[\mathcal{B}]$, et on en prend k copies.

Prouver la validité de cet algorithme revient à montrer que le générateur analytique $\Gamma_{(z,a)}[\mathcal{A}]$ retourne un objet $\alpha \in \mathcal{A}$ avec probabilité $z^{|\alpha|}/a$. Le k-uplet $\alpha = (\beta, \beta, \ldots, \beta)$ contient k objets identiques. On a alors, en supposant encore que le générateur analytique pour $\mathcal{B}$ est correct,

$$\mathbb{P}_{z,a}[\alpha \ \in \mathcal{A}] = \frac{b}{a} \cdot \frac{(z^k)^{|\beta|}}{b} = \frac{z^{k|\beta|}}{a} = \frac{z^{|\alpha|}}{a}.$$

6.3.2 Construction des paires non ordonnées

Soit $\mathcal{B}$ une classe combinatoire, et $\mathcal{A} = \mathrm{MSET}_2\,\mathcal{B}$ la classe contenant les paires non-ordonnées des éléments de $\mathcal{B}$. Les fonctions génératrices correspondantes $A(z)$ et $B(z)$ vérifient l'équation

fonctionnelle :

$$A(z) = \frac{B(z)^2 + B(z^2)}{2}.$$

supposant qu'il y a un générateur analytique pour $\mathcal{B}$, on peut construire un générateur analytique pour $\mathcal{A}$. Soit (z, b) et $(z^2, \bar{b})$ des coordonnés analytiques valides pour $\mathcal{B}$ (on remarque que la variable z doit être la même dans les deux paires), et soit (z, a) des coordonnées analytiquement valides pour $\mathcal{A}$, c'est-à-dire

$$a \geqslant \frac{b^2 + \bar{b}}{2}.$$

Supposant qu'il y a un générateur analytique pour $\mathcal{B}$, on peut construire un générateur analytique pour $\mathcal{A}$. Soit (z, b) et $(z^2, \bar{b})$ analytiquement valides pour $\mathcal{B}$ (la variable z doit être la même dans les deux coordonnées), et soit (z, a) analytiquement valide pour $\mathcal{A}$, c'est à dire :

$$a \geqslant \frac{b^2 + \bar{b}}{2}.$$

En utilisant les notations précédentes,

$$\Gamma_{(z,a)}[\mathcal{A}] : \left[\frac{b^2 + \bar{b}}{2a}\right] \cdot \mathrm{Ber}\left(\frac{b^2}{b^2 + \bar{b}}\right) \Rightarrow \left\{\Gamma_{(z,b)}[\mathcal{B}] \ ; \ \Gamma_{(z,b)}[\mathcal{B}]\right\} \quad | \quad \Gamma_{(z^2,\bar{b})}[\mathcal{B}] \text{ et dupliquer.}$$

En d'autres termes, après avoir fait le test de mort, on choisit avec la probabilité appropriée si on crée une paire d'éléments correspondant à des appels indépendants à $\Gamma_{(z,b)}[\mathcal{B}]$, ou si on fait une appel à $\Gamma_{(z^2,\bar{b})}[\mathcal{B}]$, et on duplique le résultat obtenu et on obtient une paire d'objets identiques. On le construit aisément à l'aide des réplications élémentaires et des constructions de base.

6.3.3 Arbres d'Otter

On vient de voir la classe $\mathcal{B}$ des arbres binaires. Ces arbres binaires sont *planaires*, dans le sens où les fils d'un nœud interne sont distingués ; il y a un nœud gauche et un nœud droit. On considère maintenant la classe $\mathcal{V}$ des *arbres d'Otter*, qui sont des arbres binaires de recherche *non planaires*, en utilisant notre opérateur MSet[2] :

$$\mathcal{V} = \mathcal{Z} + \mathrm{MSet}_2\mathcal{V}. \tag{6.9}$$

La fonction génératrice $V(z)$ pour les arbres d'Otter satisfait l'équation fonctionnelle

$$V(z) = z + \frac{V(z)^2 + V(z^2)}{2}.$$

On remarque que pour cette classe on ne compte que les nœuds externes. Cette classe combinatoire n'a pas de forme close pour sa fonction génératrice : les précédents générateurs de Boltzmann pour les arbres d'Otter ont déjà informellement utilisés des approximations [21, §5] ; Pivoteau [45] a utilisé le fait que $V(z) = 1 - \sqrt{1 - 2z - V(z^2)}$ pour en tirer une approximation. En pratique ces approximations font des simulations correctes, mais théoriquement elles peuvent introduire un biais. Avec les générateurs analytiques, les biais sont corrigés par la mort de certaines générations ; cela nous donne aussi plus de flexibilité pour choisir les approximations.

Construire l'inéquation. Pour notre générateur analytique, on a besoin des valeurs $v_{[i]}$, correspondant à $V(z^{2^i})$, qui sont récursivement définis par le système d'inéquations :

$$\forall i \in \mathbb{N}^+, \quad v_{[i]} \geqslant z^{2^i} + \frac{{v_{[i]}}^2 + v_{[i+1]}}{2}. \tag{6.10}$$

Puisque ce système est infini, on va décider d'une valeur de palier i limite après laquelle les équations seront approximées ; et on calculera une bonne approximation pour les termes restants.

Pour être capable de trouver des solutions, on a besoin d'un intervalle initial pour z, qui n'a pas besoin d'être précis. Pour ce faire $0 \leqslant z \leqslant 1$ suffit (même si il est assez facile de trouver que

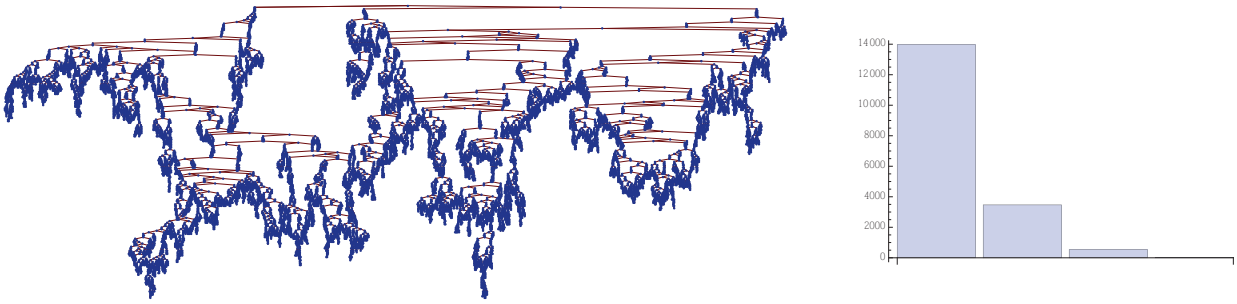


FIGURE 6.4 – Un arbre d'Otter de taille $n = 17\,979$.

$1/4 < z < 1/2$). La partie constante de cette inéquation récursive est z^{2^i}, il est donc raisonnable de poser $v_{[i]} = Kz^{2^i}$, que l'on peut ensuite injecter dans notre inéquation. En divisant des deux cotés par z^{2^i} et en factorisant, on obtient

$$K \geqslant 1 + \frac{Kz^{2^i}}{2}(K+1)\,. \tag{6.11}$$

Choix des paramètres. Il nous reste donc seulement deux paramètres à choisir. D'abord on doit trouver une constante K satisfaisant l'inéquation (6.11) ; K peut être aussi petit que l'on veut, tant que $K > 1$. On doit aussi choisir une valeur de palier i_0 qui va déterminer la valeur de i où l'on va cesser de calculer de bonnes approximations de $V(z^{2^i})$.

Une fois que l'on a choisi une valeur limite de palier i_0, et une constante K, qui approcheront les termes $v_{[i]}$ pour $i > i_0$, on peut calculer les termes initiaux de manière exacte. Cela est fait en résolvant de manière exacte les équations quadratiques,

$$\frac{1}{2}{v_{[i]}}^2 - v_{[i]} + \left(z^{2^i} + \frac{1}{2}v_{[i+1]}\right) = 0$$

En allant en arrière de $i_0 - 1$ à 0, et en utilisant pour les termes restants $o_{[i]} = Kz^{2^i}$.

Les approximations prises vont avoir un impact sur le taux de mort, et on peut le réduire en prenant les mesures suivantes : on peut choisir un plus grand palier. i_0 ; on peut choisir un z plus proche de la singularité ou on peut utiliser plus qu'une constante pour la partie où on rejette $z^{2^{i+1}}$ pour approximer les termes après le palier.

Cela fait un générateur efficace pour les arbres d'Otter, dont nous avons dessiné un très grand en Figure 6.4. C'est un arbre d'Otter de taille $n = 17\,979$ (la taille visée était $20\,000 \pm 10\%$), généré en $13s$ sur un ordinateur portable standard (Macbook Air 2012). L'histogramme indique le degré de symétrie des feuilles ; la première barre indique le nombre de feuilles non dupliquées, puis dupliquées une fois, puis quatre fois, puis huit fois.

6.3.4 L'opérateur de pointage (différentiation)

Une fonction génératrice dérivée est une fonction génératrice comme les autres, il n'y a donc pas besoin de traitement spécial, excepté qu'on doit la considérer comme une nouvelle fonction génératrice, avec une nouvelle variable. On va faire un exemple avec les arbres binaires planaires pointés.

Soit $\mathcal{B} = \mathcal{Z} + \mathcal{Z}\mathcal{B}^2$ la classe des arbres binaires planaires où tous les nœuds sont comptés. Alors, suivant la définition traditionnelle de l'opérateur de *marquage* Θ, on a $\Theta\mathcal{B} = \mathcal{Z} + \mathcal{Z}\mathcal{B}^2 + \mathcal{Z}(\Theta\mathcal{B})\mathcal{B}2 + \mathcal{Z}\mathcal{B}(\Theta\mathcal{B})$. Les inégalités associées sont

$$\begin{aligned} \bar{b} &\geqslant z + zb^2 + 2zb\bar{b} \\ b &\geqslant z + zb^2. \end{aligned}$$

On peut ensuite faire un générateur pour $\Theta\mathcal{B}$ en utilisant les constructions précédemment décrites.

6.4 Conclusion

On a montré dans ce chapitre comment on peut étendre le modèle de génération de Boltzmann, pour gérer efficacement les problèmes d'approximations de nombres flottants en conservant l'uniformité de la génération. La simplicité de cette extension réside dans le fait que l'on se contente d'agrandir l'espace des valeurs de paramètre que l'on s'autorise à prendre, et que l'on paye le prix après coup en ajoutant du rejet. Cette méthode est donc applicable dans tous les cas rencontrés pour la génération de Boltzmann classique. Cela permet notamment de pouvoir faire de la génération de Boltzmann en utilisant uniquement des valeurs rationnelles en limitant les probabilités de rejet dues aux approximations (puisque les rationnels sont denses dans les réels), ce qui simplifie grandement les problèmes d'évaluation de nombres flottants. On a montré comment on peut utiliser ce cadre pour générer simplement les arbres d'Otter, en utilisant juste une courte analyse préalable des équations induites par la spécification de cette classe.

Conclusion et perspectives

Chapitre 7

Conclusion et perspectives

A travers cette thèse, j'ai mis l'accent sur les opérateurs de la méthode symbolique et leurs usages. Dans la première première partie j'ai apporté un nouvel opérateur (le produit avec contrainte d'ordre partiel), ainsi qu'une méthode pour spécifier des classes d'opérateurs, et j'ai utilisé les spécifications pour pourvoir analyser des structures. La deuxième partie présente des résultats sur les méthodes de génération aléatoire, avec une extension des opérateurs utilisables avec la génération récursive, une étude comparative de deux techniques pour la génération de Boltzmann ainsi qu'une extension de la génération de Boltzmann.

Le premier chapitre montre des résultats quantitatifs sur des modèles d'arbres croissants, qui modélisent des objets issus de la théorie de la concurrence. C'est une extension d'un résultat précédent, où l'on a modifié la nature d'un des opérateurs, qui passe d'arité quelconque à binaire. Le résultat est que la différence de profil entre les deux modèles (binaire et d'arité quelconque) est minime, puisque l'équivalent asymptotique de la taille est porté par une factorielle dans les deux cas, et seul le facteur exponentiel change. On obtient aussi un résultat sur les profils dans le cas où les entrées des opérateurs binaires commutent, et l'on montre que le gain de taille effectuée n'est que d'un facteur exponentiel, et ne compense pas le terme factoriel.

Le deuxième chapitre introduit un nouvel opérateur sur les classes étiquetée, qui permet de faire des produits avec des contraintes d'ordres complexes, qui généralise le produit avec contrainte d'ordre binaire traditionnel. L'opérateur résultant se traduit sur les séries génératrices en une intégrale multiple, où l'ordre désiré se retrouve dans le domaine d'intégration. On montre que l'on peut spécifier des objets complexes, comme des arbres croissants où deux nœuds se partagent une étiquette, en utilisant ce produit. On en tire une formule close directement utilisable par les théorèmes classiques de la combinatoire analytique. On peut aussi spécifier des objets plus complexes (par exemple des arbres croissants avec deux partages d'étiquettes), mais la modélisation et les calculs deviennent plus difficiles.

Le troisième chapitre donne un cadre naturel sur les constructions pour la composition et l'inversion compositionnelle. On le réalise en limitant l'ensemble des constructions auquel on s'intéresse, qui sont les constructions décomposables en réplications. Cela permet d'interpréter la preuve classique de la construction du cycle en utilisant des opérations sur les constructions, de déduire la décomposition du cycle en réplications élémentaires, et ainsi de retrouver les formules classiques du cycle. Cette méthode s'avère aussi efficace pour spécifier les constructions d'ensembles et de multiensemble. Il reste à étendre le cadre des constructions réplicables pour inclure comme construction de base celle correspondant à la classe atomique, afin de pouvoir définir des classes comme des points fixes de constructions réplicables.

Le quatrième chapitre étend le cadre classique de génération récursive pour montrer comment étiqueter efficacement des structures avec des contraintes d'ordres non triviales, comme par exemple celles étudiées au premier et deuxième chapitres.

Le cinquième chapitre étudie les différences entre deux méthodes classiques de génération uniforme de structures, le pointage et la méthode de la singularité. La conclusion est que les deux méthodes ne diffèrent que d'un facteur constant, la plus efficace pour la complexité comptabilisée étant de manière prévisible la méthode du pointage. Ce facteur dépend du type de structure, mais il est compris entre 1.3 et 10 pour des structures raisonnables ($0 < -\alpha < 0.94$). On en conclut donc

que la méthode du pointage est à préférer si l'on désire la meilleure complexité, mais que la méthode de la singularité a sa place si d'autres facteurs rendent la rend plus attractive (comme par exemple une évaluation plus simple des paramètres).

Le sixième chapitre introduit une extension du modèle de génération de Boltzmann, où l'on rajoute un rejet supplémentaire, qui va être utilisé pour corriger des approximations réalisées lors du calcul des constantes nécessaires pour la génération de Boltzmann. On montre comment on peut profiter de la liberté offerte de pouvoir approximer la singularité et les valeurs des oracles sans pour autant affecter significativement la complexité de la génération. De plus cette approche est parfaitement compatible avec toutes les techniques utilisables sur la génération de Boltzmann classique, comme par exemple la méthode de calcul de l'oracle de Pivoteau *et al.* On montre notamment comment les arbres d'Otter, complexes à générer uniformément avec la génération de Boltzmann classique, sont générés simplement avec cette méthode.

Bibliographie

[1] F. Bergeron. Une combinatoire du pléthysme. *Journal of Combinatorial Theory, Series A*, 46(2) :291 – 305, 1987.

[2] F. Bergeron, P. Flajolet, and B. Salvy. Varieties of increasing trees. In *CAAP*, pages 24–48, 1992.

[3] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*, volume 67. Cambridge University Press, 1998.

[4] O. Bodini, A. Genitrini, and F. Peschanski. Enumeration and random generation of concurrent computations. In *23rd International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, (AofA)*, pages 83–96, 2012.

[5] O. Bodini, A. Genitrini, and F. Peschanski. Enumeration and random generation of concurrent computations. In *arXiv/1407.1873*, page Under Submission, 2014.

[6] O. Bodini, A. Genitrini, and F. Peschanski. A Quantitative Study of Pure Parallel Processes. *Electronic Journal of Combinatorics*, 23(1) :P1.11, 39 pages, (electronic), 2016.

[7] O. Bodini, A. Genitrini, F. Peschanski, and N. Rolin. Associativity for binary parallel processes : A quantitative study. In *Algorithms and Discrete Applied Mathematics - First International Conference, CALDAM 2015, Kanpur, India, February 8-10, 2015. Proceedings*, pages 217–228, 2015.

[8] O. Bodini, A. Genitrini, and N. Rolin. Pointed versus singular boltzmann samplers. In *Proceedings of Gascom 2014*, 2014.

[9] O. Bodini, A. Genitrini, and N. Rolin. Extended boxed product and application to synchronized trees. In *Proceedings of Gascom 2016*, 2016.

[10] O. Bodini, J. Lumbroso, and N. Rolin. Analytic samplers and the combinatorial rejection method. In *Proceedings of the Twelfth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2015, San Diego, CA, USA, January 4, 2015*, pages 40–50, 2015.

[11] M. Bodirsky, É. Fusy, M. Kang, and S. Vigerske. Boltzmann samplers, pólya theory, and cycle pointing. *SIAM Journal on Computing.*, 0(3) : 721-769, 2011.

[12] R. P. Brent. Algorithms for minimization without derivatives. *Courier Dover Publications*, 1973.

[13] G. Brightwell and P. Winkler. Counting linear extensions is #P-complete. In *STOC*, pages 175–181, 1991.

[14] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2) :244–263, 1986.

[15] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W Function. In *Advances in Computational Mathematics*, volume 5, pages 329–359, 1996.

[16] A. Darrasse. *Structures arborescentes complexes : analyse combinatoire, génération aléatoire et applications*. PhD thesis, Paris 6, 2010.

[17] L. Devroye. *Non-Uniform Random Variate Generation*. Springer Verlag, 1986.

[18] M. Drmota. *Random trees*. Springer, Vienna-New York, 2009.

[19] G. Duchamp, F. Hivert, J.-C. Novelli, and J.-Y. Thibon. Noncommutative symmetric functions vii : Free quasi-symmetric functions revisited. *Annals of Combinatorics*, 15 :655–673, 2011.

[20] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5) :577–625, 2004. Special issue on Analysis of Algorithms.

[21] P. Flajolet, É. Fusy, and C. Pivoteau. Boltzmann sampling of unlabelled structures. In D. A. *et al.*, editor, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*, pages 201–211. SIAM Press, 2007. Proceedings of the New Orleans Conference.

[22] P. Flajolet and A. M. Odlyzko. Singularity analysis of generating functions. *In SIAM J. Discrete Math.*, 3 :216–240, 1990.

[23] P. Flajolet, M. Pelletier, and M. Soria. On Buffon machines and numbers. In D. Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 172–183. SIAM, 2011.

[24] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2009. 824 pages (ISBN-13 : 9780521898065) ; also available electronically from the authors' home pages.

[25] P. Flajolet and M. Soria. The cycle construction. *SIAM Journal on Discrete Mathematics*, 4(1) :58–60, 1991.

[26] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A Calculus for the Random Generation of Labelled Combinatorial Structures. *Theoretical Computer Science*, 132(1-2) :1–35, 1994.

[27] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7 :167–203, 1999.

[28] S. Gerhold. On Some Non-Holonomic Sequences. *Elec. J. Comb.*, 11(1) :1–8, 2004.

[29] D. H. Greene. *Labelled Formal Languages and Their Uses.* PhD thesis, Stanford, CA, USA, 1983. AAI8320712.

[30] P. Henrici. *Applied and computational complex analysis. Volume 2.* Pure and applied mathematics. John Wiley and Sons, New-York, London, Sydney, 1977.

[31] G. R. Heubach S., Chinn P. Patterns arising from tiling rectangles with 1x1 and 2x2 squares. *Congressus Numerantium 150*, pages 173–192, 2001.

[32] H. Hwang, A. Panholzer, N. Rolin, T. Tsai, and W. Chen. Probabilistic analysis of the (1+1)-evolutionary algorithm. *CoRR*, abs/1409.4955, 2014.

[33] N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 3rd edition, 2005.

[34] D. E. Knuth. *The art of computer programming, volume 3 : (2nd ed.) sorting and searching.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[35] M.-L. Lackner and M. Wallner. An invitation to analytic combinatorics and lattice path counting. In *Lecture note of the 2015 ALEA in Europe Young Researchers' Workshop*, 2015.

[36] L. Lipshitz. The diagonal of a D-finite power series is D-finite. *J. Algebra*, 113(2) :373–378, 1988.

[37] J. Lumbroso. Optimal discrete uniform generation from coin flips, and applications. *CoRR*, abs/1304.1916, 2013.

[38] R. Mathar. Tilings of rectangular regions by rectangular tiles : Counts derived from transfer matrices. *arXiv :1406.7788 [math.CO]*, 2014.

[39] A. Meir and J. W. Moon. On the altitude of nodes in random trees. 30 :997–1015, 1978.

[40] R. Milner. *A Calculus of Communicating Systems.* Springer Verlag, 1980.

[41] M.Lothaire. *Combinatorics on Words.* Vol. 17 of Encyclopedia of Mathematics and its Applications, 1983.

[42] S. P. S. R. K. S. Neil J. Calkin, Kevin James and M. Yancey. Counting kings : as easy as $\lambda_1, \lambda_2, \lambda_3$.... *Congr. Numer.183*, 2006.

[43] A. Nijenhuis and H. S. Wilf. *Combinatorial algorithms.* Academic Press, 1978.

[44] A. Noack. A Class of Random Variables with Discrete Distributions. *The Annals of Mathematical Statistics*, 21(1) :127–132, March 1950.

[45] C. Pivoteau. génération aléatoire et modeles de boltzmann. Master's thesis, 2009.

[46] C. Pivoteau, B. Salvy, and M. Soria. Boltzmann oracle for combinatorial systems. In *Proceedings of the Fifth Colloquium on Mathematics and Computer Science. Blaubeuren, Germany*, pages 475–488, 2008.

[47] C. Pivoteau, B. Salvy, and M. Soria. Algorithms for combinatorial structures : Well-founded systems and Newton iterations. *Journal of Combinatorial Theory, Series A*, 119(8) :1711–1773, 2012.

[48] N. Rolin and A. Ugolnikova. Tilings by 1×1 and 2×2. *RAIRO - Theor. Inf. and Applic.*, 50(1) :105–116, 2016.

[49] G.-C. Rota. On the foundations of combinatorial theory i : Theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 2 (4)*, pages 340–368, 1964.

[50] H. S. Tiling an m-by-n area with squares of size up to k-by-k with $m \leq 5$. *Congressus Numerantium 140*, pages 43–64, 1999.

[51] N. J. A. Sloane. The on-line encyclopedia of integer sequences. *Electr. J. Comb.*, 1, 1994.

[52] R. P. Stanley. *Enumerative Combinatorics*. Wadsworth & Brooks/Cole, 1986.

[53] R. P. Stanley. Two poset polytopes. *Discrete & Comp. Geometry*, 1 :9–23, 1986.

Annexe A

Pavages 1 et 2

Introduction

In the present work, we look at tilings of a $k \times n$ board ($n, k \in \mathbb{N}$) by 1×1 (small) and 2×2 (big) squares with no holes or overlaping. The goal is to understand how the average proportion of small squares in all possible tilings of a $k \times n$ rectangle by small and big squares changes when $k, n \to +\infty$. Another question is to find the number of small squares in tilings of a $k \times n$ rectangle that maximises the number of tilings of a $k \times n$ rectangle. A simpler problem that we study here is to consider that k is fixed and $n \to +\infty$.

There has been some work done on the subject. When $k = 2$, tilings of a $2 \times n$ rectangle by 1×1 and 2×2 squares correspond to the Fibonacci sequence. For $k = 3$, one can easily show that the number of ways to cover a $3 \times n$ rectangle with 1×1 and 2×2 squares is equal to $\frac{1}{3}(-1)^n + \frac{1}{3}2^{n+1}$.

Some results were obtained by Heubach S. [50, 31]. Namely, explicit formulas for the number of tilings for k up to 5 by using introduced basic blocks and methods of analytic combinatorics for finding poles of generating functions and asymptotics. Bigger cases, however, seem to pose problems, mainly because it becomes difficult due to the number of basic blocks.

This abstract consists of four main sections, introduction and conclusion. In Section A.1 we define a set of Bivariate Generating Functions ($BGFs$) associated with tilings of a $k \times l$ rectangle (supposing that $n = lk,\ l \in \mathbb{N}$), present formulas for small cases and calculate distribution of small squares in tilings for $k \leq 10$. In Section A.2 we introduce an automaton construction that represents $BGFs$ and their relations. We extract some properties on its structure, present a simplification algorithm that allows to compute $BGFs$ more easily. In section A.3 we present another point of view on this problem related to the matrix representation. In section A.4 we introduce the problem for a cylindrical case. In conlusion, we show combinatorical results on the proportions of small squares for $k \leq 10$ for the plain case and for and $k \leq 8$ for the cylindrical case, and mention some open question.

A.1 Settings, definitions

A.1.1 Bivariate generating function

In order to study the general case, we introduce $BGFs$. For the sake of simplicity we shall define them for the case $k = 4$ and then generalize the definition. Let

$$Q_{0000}(z, u) = \sum_{n,p} A^4_{n,p} z^n u^p$$

be a BGF where the coefficient $A^4_{n,p}$ of $z^n u^p$ is the number of tilings of a $4 \times \frac{n}{4}$ rectangle with exactly p small squares, supposing that n is a multiple of 4. We want to underline that the rectangle is of area n. This choice is due to the simpler way of defining equations on $BGFs$.

Let $Q_{1000}(z, u)$ be a BGF with the coefficient of $z^n u^p$ being the number of tilings of the initial rectangle with a 1×1 square cut off from the upper left corner and $Q_{2200}(z, u)$ a BGF with the

FIGURE A.1 – $4 \times \frac{n}{4}$ board with cut off corners

coefficient of $z^n u^p$ corresponding to the number of tilings of the initial rectangle with a 2×2 square cut off from the upper left corner (illustrations are shown in Figure A.1).

From this point on, we will write $BGFs$ without arguments, always meaning that they are z, u. A relation on Q_{0000}, Q_{1000} and Q_{2200} can be expressed in the following way :

$$Q_{0000} = zuQ_{1000} + z^4 Q_{2200}.$$

Indeed, in order to obtain Q_{0000}, we can either cut off a small square or a big one from the upper left corner of the initial $4 \times \frac{n}{4}$ rectangle. The remaining areas will correspond either to Q_{1000} or to Q_{2200}. And because we cut off squares we need to multiply Q_{1000} by zu (z corresponds to the area occupied by a small square, u – to the one small square) and Q_{2200} by z^4 respectively.

In the same way we can introduce Q_{1100} and Q_{1220} and we have the relation

$$Q_{1000} = zuQ_{1100} + z^4 Q_{1220}.$$

At each step we change indexes of $Q_{i_1 i_2 i_3 i_4}$ by going from left to right in the following way : we permit changing either one 0 to 1 or 00 to 22, which means changing the left one or two columns of the board that was obtained at the previous step by cutting off either a 1×1 or a 2×2 square from the upper left corner of the board. By this rule one can never obtain Q_{1010} or Q_{1022}, for example.

As soon as we get to $Q_{i_1 i_2 i_3 i_4}$ with all indexes being different from zero, we can use a *tetris* rule to reduce the indexes of $Q_{i_1 i_2 i_3 i_4}$ by one layer with "no charge". With "no charge" here means that, given that our strip is infinite, $Q_{1122} = Q_{0011}$, $Q_{1111} = Q_{0000}$ and so on.

Using this technique one obtains a finite set of $BGFs$ $Q_{i_1 i_2 i_3 i_4}$ and a system of functional equations on them. For $k \geq 5$, the principle of constructing a set of $Q_{i_1 \ldots i_k}$ and a system of functional equations is the same.

A.1.2 Entropy

In this section , for an easier representation, let us denote as $A_l(k)$ the number of configurations of a $k \times l$ rectangle that can be easily obtained from the previous subsection. We define the entropy η of our system as follows :

Definition 2.

$$\eta = \lim_{l,k \to \infty} A_l(k)^{\frac{1}{kl}}.$$

Proposition A.1.1. *For $n, k, l \in \mathbb{N}$*

$$A_l(n + k - 1) \leq A_l(n) A_l(k) \leq A_l(n + k).$$

Proof. The right inequality follows from the fact that when we stick two rectangles of heights k and n together we have a perfect boundary, so the number of configurations is less than in the rectangle of size $k + n$ where there can be big squares on the boundary.

To prove the left inequality let us take a paving of a rectangle of height $k + n - 1$, and look at what happens at height k. On this level we have two types of big squares, those that come from the $(k - 1)$th level and those that come from the $(k + 1)$th level. This means that we can cut the rectangle in two parts, with each part keeping their big squares from the border (kth level), and filling the missing spaces with small squares. The result consists of 2 valid pavings of sizes k and n respectively, which can be used to cannonically rebuild the initial rectangle of height $k + n - 1$. □

Corollaire A.1.1.

$$A_l(p) \leq A_l(p-1)\frac{\phi^l}{\sqrt{5}}, \tag{A.1}$$

and

$$A_l(p) \leq \frac{\phi^{lp}}{\sqrt{5}}, \tag{A.2}$$

where $\phi = \frac{\sqrt{5}+1}{2}$.

Proof. Take $p = k+1, n = 2$ in the left inequality in Proposition 1 and remember that $A_l(2)$ is the $l-th$ Fibinacci number to get (1). Then apply (1) $p-2$ times to get (2). □

Corollaire A.1.2.

$$A_l(n) \geq \frac{\phi^{l\lceil \frac{n}{2} \rceil}}{\sqrt{5}}.$$

Proof. It suffices to take $k = 2$ in the right inequality in Proposition 1, apply it n times and use the fact that $A_l(2)$ is the $l-th$ Fibonacci number. □

Corollaire A.1.3. *Let*

$$A(n) = \lim_{l\to\infty} A_l(n),$$

then for $n = kp$

$$A(k)^p \leq A(n) \leq A(k+1)^p \tag{A.3}$$

and

$$A(k)^{\frac{1}{k}} \leq \eta \leq A(k+1)^{\frac{1}{k}}. \tag{A.4}$$

Proof. From (3) we get

$$A(k)^{\frac{1}{k}} \leq A(n)^{\frac{1}{n}} \leq A(k+1)^{\frac{1}{k}}.$$

If we let p tend to ∞ with k being fixed we get (4). □

For k up till 9 we have bounds (see the table in the end of Subsection A.1.3) on $A(kp)$ that become tighter with the increase of k. The further we could go with the calculation, the tighter bounds could be obtained.

A.1.3 Combinatorical results

Using traditional combinatorical tools (see, e.g., [24]) we can find formulas for our $BGFs$ and extract some properties. We can solve a system of equations and find $Q_{0\ldots0}(z,u)$ for small k. It starts getting complex for $k \geq 10$ given that the size of the associated matrix grows exponentially.

For example, for $k = 4$

$$Q_{0000}(z,u) = \frac{1-z^4}{1-z^4-z^4u^4-2z^8u^4-z^8+z^{12}u^4+z^{12}}$$

$$Q_{0000}(z,1) = \frac{1-z^4}{1-2z^4-3z^8+2z^{12}}$$

The coefficients A_n^k of $Q_{0000}(z,1)$ correspond to tilings of a $4\times\frac{n}{4}$ rectangle . They satisfy satisfy the recurrence equation : $a_n = 2a_{n-1}+3a_{n-2}-2a_{n-3}$ with $a_0 = a_1 = 1, a_2 = 5$ [A054854] [51].

For $k = 5$

$$Q_{00000} = \frac{u^2z^{10}+uz^5-1}{1-u^5z^5-uz^5-3u^6z^{10}-4u^2z^{10}-u^7z^{15}+3u^3z^{15}-3u^4z^{20}}.$$

The standart technique to calculate the expected value of a certain random variable is to differentiate generating functions and then use the singularity analysis [24]. We are interested in calculating the average proportion of space occupied by small squares. Imagine that we wanted to calculate the

k	z_0	%
3	0.7937	55.555
4	0.7721	46.954
5	0.7701	49.507
6	0.7642	47.241
7	0.7621	47.759
8	0.7596	47.029
9	0.7586	47.055
10	0.7656	46.764

FIGURE A.2 – k is the height of the region, z_0 – dominant singularity of the corresponding BGF, % – average percentage of space occupied by small squares.

expected value of the number of small squares, then we could have used the standart technique that is well detailed in, for example, [35] and look at

$$\frac{\partial_u Q_{0\ldots0}(z,1)}{Q_{0\ldots0}(z,1)}.$$

Since we need the proportion of space occupied by small squares, we need to extract the area n from the denominator. We can do this by differentiating $Q_{0\ldots0}(z,u)$ by z and multiplying it by z. It gives us the following expression :

$$\frac{\partial_u Q_{0\ldots0}(z,1)}{z\partial_z Q_{0\ldots0}(z,1)}$$

And the singularity analysis will directly give us the average proportions of space occupied by small squares in rectangles. Therefore we calculate the singularities of $Q_{0\ldots0}(z,u)$ that are the closest to zero. The list of singularities
for $k = 2,\ldots,9$ is the folowing : $0.7861, 0.7937, 0.7721, 0.7701, 0.7642, 0.7621, 0.7961, 0.7842$. Naming z_0 the dominant singularity (closest to zero) we get

$$\left.\frac{\partial_u Q_{0\ldots0}(z,u)}{z\partial_z Q_{0\ldots0}(z,u)}\right|_{(1,z_0)}.$$

Average proportions of space occupied by small squares for $k \leq 10$ are shown in Figure A.2.

A.2 Automaton representation

For each k let us introduce an automaton. Each $Q_{i_1\ldots i_k}$ with $i_j \in \{0,1,2\}$ for $j = 1,\ldots k$ is associated with a state $q = i_1\ldots i_k$ and each functional equation involving $BGFs$ can be translated into an automaton transition. For example, the relation

$$Q_{0000} = zuQ_{1000} + z^4 Q_{2200}.$$

is represented in the following way : an arrow marked by zu goes from the state 1000 to the state 0000, an arrow marked by z^4 goes from the state 2200 to the state 0000. When the tetris rule is applied, we will mark the corresponding arrows by *a star*.

Commentary : The unusual way, one might say, of directing arrows can be explained by the fact that adding 1 or 22 to the indexes of $BGFs$ corresponds to cutting off corners of the initial rectangle.

For $k = 4$ the set of states consists of the states : 0000, 1000, 2200, 1100, 1220, 2210, 2222, 1110, 1122, 1221, 1111, 0011, 0110, 1011, 2211 and an illustration of the automaton is shown in Figure A.3.

We shall refer to the state that consists of all $0s$ as *initial*. Calculation of the paths in the automaton that start and end at the initial state will allow us to find formulas for $Q_{0\ldots0}$. Our objective is to decrease the computational complexity by reducing the number of states, which basically means reducing the number of functional equations in the system.

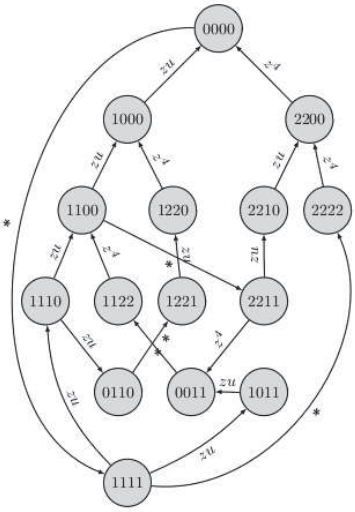


FIGURE A.3 – Automaton for $k = 4$

A.2.1 Essential, non-essential and additional states

Definition 3. *A state q of an automaton is called essential if there are at least two arrows coming in and out of q and at least one of the arrows coming out is marked by a star. It is called non-essential otherwise.*

Let E_k be the set of all essential states for each $k \geq 4$. One can see from Figure 2 that $E_4 = \{1100\}$ and $|E_4| = 1$. Let us describe the structure of E_k and find $|E_k|$.

Proposition A.2.1. *A state $q = i_1...i_k$ of an automaton is essential if and only if q has the following propertries :*

1. *q consists only of $0s$ and $1s$;*
2. *All $1s$ come in consecutive pairs in q ;*
3. *$i_1 = i_2 = 1$;*
4. *There are at least two $0s$ and the leftmost 0 in q comes in a pair with another 0.*

Proof.

$\Rightarrow$ Let us prove that if q does not have at least one of the four properties, then q is non-essential. If there exists $j \in \{1, ..., k\}$ such that $i_j = 2$ or a block of consecutive $1s$ whose length is odd in q, then no state can be reduced to q by the use of the tetris rule, so no arrow marked by a star comes out of it. If q starts with a zero, then at most one arrow comes out of q (to the state that is reduced to q by the tetris rule). If the leftmost zero in q is isolated, then only one arrow comes in q (from the state with the leftmost 0 being replaced by 1).

$\Leftarrow$ Consider that q has these four properties. Given that the leftmost 0 is not isolated, there are two arrows coming in q. And it is clear that there are two arrows that come out of q – one to the state $q' = i_1 + 1 \dots i_k + 1$ that is reduced to q by the tetris rule and one to a state with a 1 on the left from the leftmost 0 being replaced by a 0.

□

Corollaire A.2.1. *With $k \to \infty$*

$$|E_k| \sim \frac{\phi^{k-3}}{\sqrt{5}}.$$

Proof. Looking at essential states of length k is the same as looking at states of length $k-3$ that are obtained from essential states by deleting the first two $1s$ and gluing together the first two consecutive $0s$ in each essential state. The obtaiend states are in bijection with tilings of a strip $1 \times k-3$ by blocks of size 1×1 and 1×2 . □

Proposition A.2.2. *For $k \geq 4$ the number of essential states $|E_k|$ in the automaton is represented by the following formula :*

$$|E_k| = \sum_{i=1}^{\lfloor \frac{k-2}{2} \rfloor} \sum_{j=1}^{i} \binom{k-i-j-2}{i-j}.$$

Proof. For every i that corresponds to the number of pairs of 11 we calculate the number of possible essential states. For every $i = 1, \ldots, \lfloor \frac{k-2}{2} \rfloor$ we have a sum on $j = 1, \ldots i$ that corresponds to the number of pairs of 11 that precede the first pair of 00. For $j = 1, \ldots i$ the following sum can be obtained :

$$\binom{k-2-2-(i-1)}{i-1} + \binom{k-4-2-(i-2)}{i-2} + \ldots +$$

$$+\binom{k-2i-2-(i-i)}{i-i} = \sum_{j=1}^{i} \binom{k-i-j-2}{i-j}.$$

□

Definition 4. *A state is called additional if it belongs to a cycle that does not contain any essential or initial states.*

Note that only non-essential states can be additional. The interest of looking at additional states is, merely, because in order to properly reduce an automaton and get the explicit formulas for our generating functions, we need to pay attention to all the cycles in the automaton including the cycles that do not pass through the initial state. If not, we might lose some terms in the resulting formulas.

Our objective is to choose additional states in such a way, so that there won't be any cycles left in the automaton that don't include either the initial state, essential states or the chosen additional states. The idea is to minimize the number of additional states that have to be added to the initial and essential states in order to properly reduce the automaton. We are not going to calculate the number of all additional states. Nor will we minimize this number. Rather, we will define a subset of the set of additional states and try to justify this choice by proving that it provides us with the wanted structure.

Let us take a set that consists of states that have the same structure as the essential states but with an odd block of $1s$ of size at least 3 on the left from the leftmost 0. We denote this set by A_k. It follows from Proposition 2 that

$$|A_k| = \sum_{i=1}^{\lfloor \frac{k-2}{2} \rfloor} \sum_{j=1}^{i} \binom{k-2j-3-i}{i-j} = |E_{k-1}|.$$

Proposition A.2.3. *Each state from A_k is additional.*

Proof. We need to show that every $q \in A_k$ belongs to at least one cycle that doesn't contain an essential state or the initial state. For $i = 1$ $q_1 = 1110\ldots0$ and the cycle is schematically shown in Figure 3 with $f(z,u)$ and $g(z,u)$ being transition functions between states. There are as many cycles as there are possible ways to get from the state $00011\ldots\ldots$ to the state $111220\ldots0$. It is not difficult to see that there are no essential states between those two states. For other i the structure of cycles that contain q_i is analogous. □

Corollaire A.2.2. *There are states in A_k for $k \geq 6$ that belong to more than cycle with no essential or initial states contained in it.*

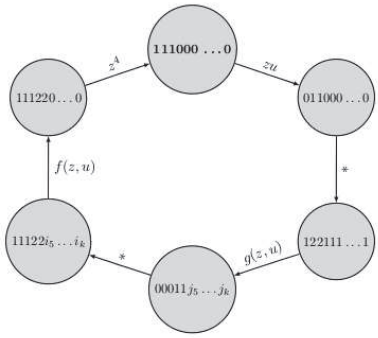


FIGURE A.4 – Cycle with the state $1110\ldots0$ for $k \geq 5$

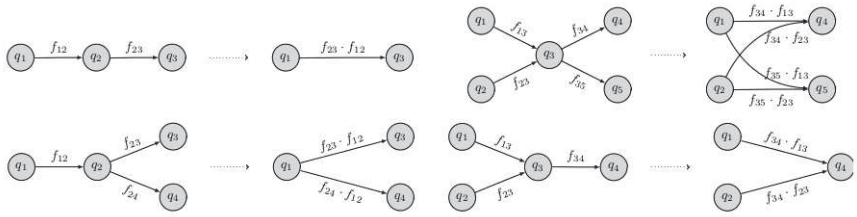


FIGURE A.5 – Rules of reduction for an automaton

Now the question is, if we mark the initial state and all the states from E_k and A_k in the automaton, does it ensure that there are no cycles left that don't contain the marked states ? Proving that will justify our choice for keeping these particular states.

Proposition A.2.4. *Let $q = i_1 \ldots i_k$ be a state that doesn't belong to $A_k \cup E_k \cup \{0\ldots0\}$. Then a cycle (or cycles) that q belongs to, contains states from $A_k \cup E_k \cup \{0\ldots0\}$.*

Proof. Let us point out that it is sufficient to prove the statement only for the states that consist of 0 and 1. So let $q = i_1 \ldots i_k$ with $i_j \in \{0,1\}$, $j = 1,\ldots k$ (Figure A.4 might help to visualize the cycles.) If the leftmost zero in q is isolated, then there is only one arrow coming in this state from a state from $A_k \cup E_k$. If the leftmost zero is not isolated, let $i_1 = \ldots = i_l = 1$, $i_{l+1} = i_{l+2} = 0$, $0 \leq l \leq k-2$. If $l = 0$, then $q = 00...011i_{n+3}...i_k$ where $i_1,\ldots,i_n = 0$, $n \geq 2$. In this case q belongs to a cycle with the state 11...100...0 belonging to $A_k \cup E_k$. The situation is similar if $l = 1$ apart from the case when $i_j = 0$ for all $k+3 \leq j \leq k$. Then $q = 10\ldots0$ and belongs to a cycle that contains the initial state. For $l \geq 2$ since all $1s$ after the $l+2$ coordinate in q come in pair, q belongs to a cycle (cycles) with a state from $A_k \cup E_k$ where the block with leftmost $0s$ is filled with $1s$. □

Remark : We shall further refer to the states from A_k as **additional***.

A.2.2 Simplified automata

We can simplify an automaton by keeping only the initial, essential and additional* states and reducing all other states. The rules of reduction are shown in Figure A.5. We denote by f_{ij} a transition between states q_i and q_j which is represented by an arrow going from q_i to q_j.

In the case $k = 4$, there is one essential state 1100 and no additional states. A reduced automaton for the case $k = 4$ is shown in Figure A.6.

For a $5 \times n$ rectangle there is one essential state 11000 and one additional* state 11100. A reduced automaton is shown in Figure A.7. For $k = 6, 7$ reduced automata are schematically shown in Figure A.8 .

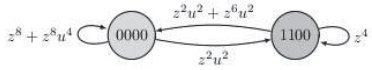


FIGURE A.6 – Reduced automaton for $k = 4$

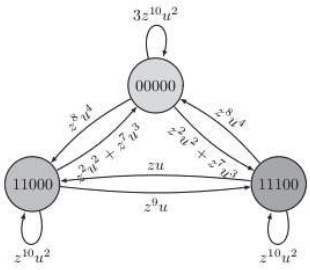


FIGURE A.7 – Reduced automaton for $k = 5$

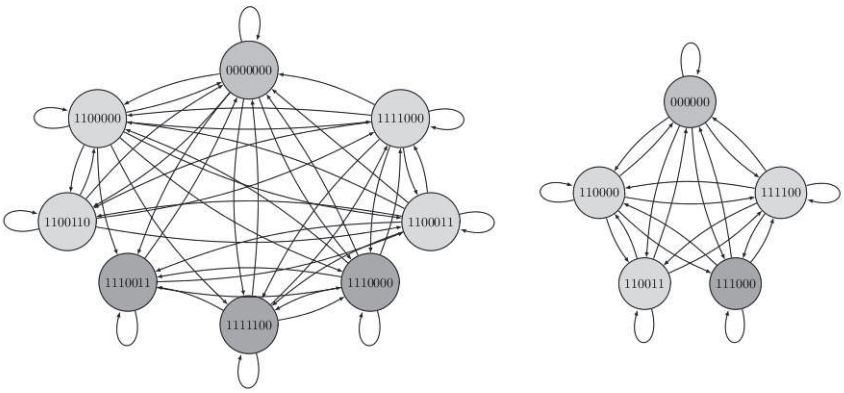


FIGURE A.8 – Reduced automata : for $k = 6$ (right), for $k = 7$ (left)

A.3 Matrix representation

Another point of view on this problem comes from the problem of non-attacking kings where one looks at the number of ways kings could be put on a rectangular board without having two king attacking each other. For the sake of completeness we mention this approach and results that were obtained previously.

There is a natural bijection between tilings of a $k \times k$ region by 1×1 and 2×2 squares and configurations of non-attacking kings on a $(k-1) \times (k-1)$ board. One simply has to consider that every big square has a king in his left bottom corner, and having no kings attacking each other is equivalent to having no big squares intersecting.

This problem has been studied using the matrix approach [42]. Namely, the use of the adjacency matrix of the graph where nodes are lines of the rectangle, two nodes are connected if the two lines can be put on top of each other (for the use of this approach see also [38]). These adjacency matrices verify the folowing recursion :

$$A_0 = (1), A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \ldots, A_n = \begin{pmatrix} A_{n-1} & A_{n-2} \\ A_{n-2} & 0 \end{pmatrix}.$$

This is in relation with the previous method, as the dominant eigenvalue of A_i is equal to the inverse of the singularity of $Q_i(z,1)$. In [42] it was used to compute good approximation of the entropy but it can be also used to compute the average proportions of space occupied by small squares for a given k. Indeed, one can compute for a fixed $k \times n$ the number of big squares by taking a matrix M which is a diagonal with the number of big squares in the corresponding node, and calculate

$$b = \mathbf{1} \left(\sum_{p \leq n} A_k^p M A_k^{n-p} \right) \mathbf{1}^T,$$

where $\mathbf{1} = (1, \ldots, 1)$.Then the proportion of big squares equals

$$\frac{4b}{kc},$$

where

$$c = \mathbf{1} \left(\sum_{p \leq n} A_k^p I A_k^{n-p} \right) \mathbf{1}^T.$$

is the number of lines in all the configurations. We can then obtain result when $n \to \infty$ by putting A_k in Jordan form and dividing both b and c by λ_1^n where λ_1 is the dominant eigenvalue of A_k.

This method allows us to find the result for k up to 9, the Jordan form beeing the limit factor as it is numerically unstable, hence not available in numerical package, and that exact resolution does not scale well with the size.

A.4 Cylindrical case

Let us now consider a $k \times \frac{n}{k}$ rectangle with sewn horizontal borders. We obtain a cylindrical region that we want to tile with 1×1 and 2×2 squares. The way of constructing BGFs stays the same with the only difference that now we allow having separated 2s in the first and last positions of the indexes of Generating Functions. The functional equations, therefore, change. For example,

$$Q^c_{0000} = zuQ^c_{1000} + z^4 Q^c_{2200} + z^4 Q^c_{2002},$$

where the index c is used to distinguish between cylindrical and plain cases.

Let us construct an automaton in the same way as before (see Figure A.9).

The notions of essential and additional states stay the same. The set E^c_k of essential states in the cylindrical case equals E_k. But the set A^c_k constists not only of states from A_k, but also of E_{k-2} and the initial state of size $k-2$ where $1s$ are added in the first and last positions which might of course

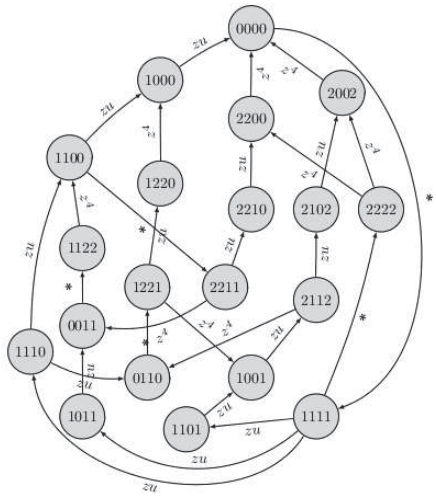


FIGURE A.9 – Automaton for $k = 4$, cylindrical case

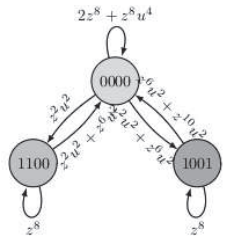


FIGURE A.10 – Reduced automaton for $k = 4$, cylindrical case

create a rightmost block of an odd length but that's due to the fact that there are states with a 2 in the first and the last positions. So $|A_k^c| = |A_k| + |E_{k-2}| + 1$.

In the case $k = 4$ there is one essential state 1100 as in plain case and one additional* state 1001. The rules of reduction stay the same and a reduced automaton is shown in Figure A.10.

In this case we have

$$Q^c_{0000} = \frac{1 - z^4}{1 - 3u^4z^8 + 2z^12 - u^4z^4 - 2z^8 - z^4},$$

and small squares occupy, in average, 0.466 of the space which is smaller than in the plain case. It is rather understandable – because of the sewn boarders there are less constraints on the way big squares can be placed.

Reduced automata for $k = 5, 6$ are schematically shown in Figure A.11.

A.5 Conclusion

Average proportions of space occupied by small squares for $k \leq 10$ for the plain case and $k \leq 8$ for the cylindrical case are shown in Figure A.12.

Representation by automata allows us to reduce the computational complexity and obtain combinatorical results for larger k. Although it remains unclear what to do when k grows given that even after reduction complexity stays exponential. Thefore, the main questions are still open : does the sequence of average proportions converge in the plain/cylindrical case ? If so, what is its limit ? Is there a relation between automata of sizes k and $k + 1$ in the plain/cylindrical case ? What can we say about the average proportion of small squares if both k and n tend to infinity ? And finally, what is the number of small squares that maximises the number of tilings of a $k \times n$ rectangle ? These questions probably need different approach.

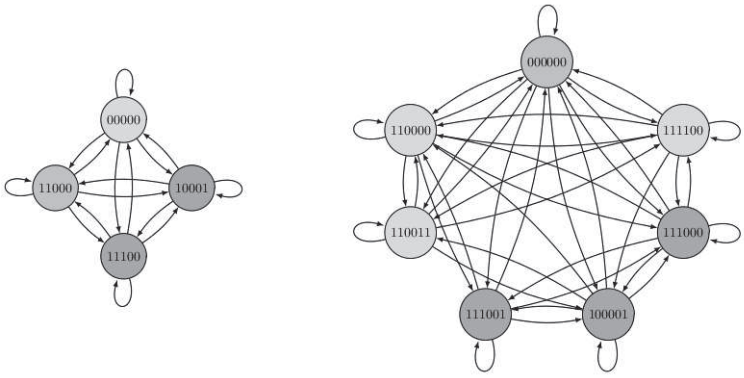


FIGURE A.11 – Reduced automata in cylindrical case : $k = 5$ (left), $k = 6$ (right)

k	% *plain*		% *cylindrical*
3	55.555		51.823
4	46.954		42.606
5	49.507		46.605
6	47.241		44.680
7	47.759		45.594
8	47.029		45.147
9	47.055		
10	46.764		

FIGURE A.12 – k is the height of the region, % *plain* – average percentage of space occupied by small squares in the plain case, % *cylindrical* – in the cylindrical case.

Annexe B

Algorithme évolutionaire 1+1

Probabilistic analysis of the (1 + 1)-evolutionary algorithm

Hsien-Kuei Hwang*
Institute of Statistical Science,
Institute of Information Science
Academia Sinica
Taipei 115
Taiwan

Alois Panholzer†
Institut für Diskrete Mathematik
und Geometrie
Technische Universität Wien
Wiedner Hauptstraße 8-10/104
1040 Wien
Austria

Nicolas Rolin
LIPN, Institut Galilée
Université Paris 13
93430, Villetaneuse
France

Tsung-Hsi Tsai
Institute of Statistical Science
Academia Sinica
Taipei 115
Taiwan

Wei-Mei Chen‡
Department of Electronic and Computer Engineering
National Taiwan University of
Science and Technology
Taipei 106
Taiwan

September 16, 2014

Abstract

We give a detailed analysis of the cost used by the (1 + 1)-evolutionary algorithm. The problem has been approached in the evolutionary algorithm literature under various views, formulation and degree of rigor. Our asymptotic approximations for the mean and the variance represent the strongest of their kind. The approach we develop is also applicable to characterize the limit laws and is based on asymptotic resolution of the underlying recurrence. While most approximations have their simple formal nature, we elaborate on the delicate error analysis required for rigorous justifications.

*Partially supported by a grant from the Franco-Taiwan Orchid Program.
†Supported by the Austrian Science Foundation FWF under the Grant P25337-N23.
‡Partially supported by MOST under the Grant 103-2221-E-011-113.

1 Introduction

The last two decades or so have seen an explosion of application areas of evolutionary algorithms (EAs) in diverse scientific or engineering disciplines. An EA is a random search heuristic, using evolutionary mechanisms such as crossover and mutation, for finding a solution that often aims at maximizing an objective function. EAs are proved to be extremely useful for combinatorial optimization problems because they can solve complicated problems with reasonable efficiency using only basic mathematical modeling and simple operators; see [5, 7, 20] for more information. Although EAs have been widely applied in solving practical problems, the analysis of their performance and efficiency, which often provides better modeling prediction for practical uses, are much less addressed; only computer simulation results are available for most of the EAs in use. See for example [3, 13, 16, 17, 18]. We are concerned in this paper with a precise probabilistic analysis of a simple algorithm called $(1+1)$-EA (see below for more details).

A typical EA comprises several ingredients: the coding of solution, the population of individuals, the selection for reproduction, the operations for breeding new individuals, and the fitness function to evaluate the new individual. Thus mathematical analysis of the total complexity or the stochastic description of the algorithm dynamics is often challenging. It proves more insightful to look instead at simplified versions of the algorithm, seeking for a compromise between mathematical tractability and general predictability. Such a consideration was first attempted by Bäck [2] and Mühlenbein in [27] in the early 1990's for the $(1+1)$-EA, using only one individual with a single mutation operator at each stage. An outline of the procedure is as follows.

Algorithm $(1+1)$**-EA**

1. Choose an initial string $\mathbf{x} \in \{0,1\}^n$ uniformly at random
2. Repeat until a terminating condition is reached
   - Create $\mathbf{y}$ by flipping each bit of x independently with probability p
   - Replace $\mathbf{x}$ by $\mathbf{y}$ iff $f(\mathbf{y}) \geqslant f(\mathbf{x})$

Step 1 is often realized by tossing a fair coin for each of the n bits, one independently of the others, and the terminating condition is usually either reaching an optimum state (if known) or by the number of iterations.

Mühlenbein [27] considered in detail the complexity of $(1+1)$-EA under the fitness function ONEMAX, which counts the number of ones, namely, $f(\mathbf{x}) = \sum_{1 \leqslant j \leqslant n} x_j$. The expected time needed to reach the optimum value, which is often referred to as the expected *optimization time*, for ONEMAX, denoted for convenience by $\mathbb{E}(X_n)$, was argued to be of order $n \log n$, indicating the efficiency of the $(1+1)$-EA. Bäck [2] derived expressions for the transition probabilities. Finer asymptotic approximation of the form

$$\mathbb{E}(X_n) = en \log n + c_1 n + o(n), \tag{1}$$

was derived by Garnier et al. in [16], where $c_1 \approx -1.9$ when the mutation rate $p = \frac{1}{n}$. They went further by characterizing the limiting distribution of $\frac{X_n - en \log n}{en}$ in terms of a log-exponential distribution (which is indeed a double exponential or a Gumbel distribution). However, some of their proofs, notably the error analysis, seem incomplete (as indicated in their paper). Thus a strong result such as (1) has remained obscure in the EA literature.

More recent attention has been paid to the analysis of the $(1+1)$-EA; see for example [1, 29]. We briefly mention some progresses. Neumann and Witt [28] proved that a simple Ant Colony Optimization algorithm behaves like the $(1+1)$-EA and all results for $(1+1)$-EA translate directly into those for the ACO algorithm. Sudholt and Witt [33] showed a similar translation into Particle Swarm Optimization algorithms. Moreover, variants such as $(\mu+1)$-EA in [34] and $(1+1)$-EA over a finite alphabet in [12] were investigated. The expected optimization time required by $(1+1)$-EA has undergone successive improvements, yet none of them reached the precision of Garnier et al.'s result (1); we summarize in the following table some recent findings.

ONEMAX function		Linear functionals	
Doerr et al. [8] (2010)	lower bound $(1-o(1))en\log(n)$	Jagerskupper [23] (2011)	upper bound $2.02en\log(n)$
Sudholt [32] 2010	lower bound $en\log(n)-2n\log\log(n)$	Doerr et al. [?] (2010)	upper bound $1.39en\log(n)$
Doerr et al. [9] (2011)	$en\log(n)-\Theta(n)$	Witt [35] (2013)	upper bound $en\log(n)+O(n)$

In this paper we focus on the mutation rate[1] $p=\frac{1}{n}$ and prove that the expected number of steps used by the $(1+1)$-EA to reach optimum for ONEMAX function satisfies

$$\mathbb{E}(X_n)=en\log n+c_1n+\tfrac{1}{2}e\log n+c_2+O\left(n^{-1}\log n\right), \tag{2}$$

where c_1 and c_2 are explicitly computable constants. More precisely,

$$c_1=-e\left(\log 2-\gamma-\phi_1\left(\tfrac{1}{2}\right)\right)\approx 1.89254\,17883\,44686\,82302\,25714\ldots,$$

where γ is Euler's constant,

$$\phi_1(z):=\int_0^z\left(\frac{1}{S_1(t)}-\frac{1}{t}\right)\mathrm{d}t, \tag{3}$$

with $S_1(z)$ an entire function defined by

$$S_1(z):=\sum_{\ell\geqslant 1}\frac{z^\ell}{\ell!}\sum_{0\leqslant j<\ell}(\ell-j)\frac{(1-z)^j}{j!}.$$

See (29) for an analytic expression and numerical value for c_2.

Note that these expressions, as well as the numerical value, are consistent with those given in [16]. Finer properties such as more precise expansions for $\mathbb{E}(X_n)$, the variance and limiting distribution will also be established. The extension to $p=\frac{c}{n}$ does not lead to additional new phenomena as already discussed in [16]; it is thus omitted in this paper.

Our approach relies essentially on the asymptotic resolution of the underlying recurrence relation for the optimization time and the method of proof is different from all previous approaches (including Markov chains, coupon collection, drift analysis, etc.). More precisely, we consider $f(\mathbf{x})=\sum_{1\leqslant j\leqslant n}x_j$ and study the random variables $X_{n,m}$, which counts the number of steps used by $(1+1)$-EA before reaching the optimum state $f(\mathbf{x})=n$ when starting

[1] From an algorithmic point of view, a mutation rate of order $\gg\frac{1}{n}$ leads to a complexity higher than polynomial, and is thus less useful.

from $f(\mathbf{x}) = n - m$. We will derive very precise asymptotic approximations for each $X_{n,m}$, $1 \leqslant m \leqslant n$. In particular, the distribution of $X_{n,m}$ is for large n well approximated by a sum of m exponential distributions, and this in turn implies a Gumbel limit law when $m \to \infty$; see Table 1 for a summary of our major results.

In addition to its own methodological merit of obtaining stronger asymptotic approximations and potential use in other problems in EA of similar nature, our approach, to the best of our knowledge, provides the first rigorous justification of Garnier et al.'s far-reaching results [16] fifteen years ago.

Although the results for linear functions strongly support the efficiency of $(1+1)$-EA, there exist several hard instances; for example, functions with $\Theta(n^n)$ expected time complexity for $(1+1)$-EA were constructed in Droste et al. [14], while a naive complete search requires only 2^n to find out the global optimum under an arbitrary function. Along another direction, long path problems were introduced by Horn et al. [21], and examined in detail in Rudolph [31]; in particular, he studied long k-paths problems (short-cuts all having distances at least k) and proved that the expected time to reach optimum is $O(k^{-1}n^{k+1})$. Droste et al. [14] then derived an exponential time bound when $k = \sqrt{n-1}$.

The $(1+1)$-EA is basically a randomized hill-climbing heuristic and cannot replace the crossover operator. Jansen and Wegener [25] showed a polynomial time for an EA using both mutation and crossover, while $(1+1)$-EA necessitates exponential running times. A more recent natural example [11] is the all-pairs shortest path problem for which an EA using crossover reaches an $O(n^3 \log n)$ expected time bound, while $(1+1)$-EA needs a higher cost $\Theta(n^4)$.

This paper is organized as follows. We begin with deriving the recurrence relation satisfied by the random variables $X_{n,m}$ (when the initial configuration is not random). From this recurrence, it is straightforward to characterize inductively the distribution of $X_{n,m}$ for small $1 \leqslant m = O(1)$. The hard case when $m \to \infty, m \leqslant n$ requires the development of more asymptotic tools, which we elaborate in Section 3. Asymptotics of the mean values of $X_{n,m}$ and X_n are presented in Section 4 with a complete error analysis and extension to a full asymptotic expansion. Section 5 then addresses the asymptotics of the variance. Limit laws are established in Section 6 by an inductive argument and fine error analysis. Finally, we consider in Section 7 the complexity of the $(1+1)$-EA using the number of leading ones as the fitness function. Denote the corresponding cost measure by Y_n and $Y_{n,m}$, respectively. We summarize our major results in the following table.

m	ONEMAX $m = O(1)$: Sum of Exp (Thm 1) $m \to \infty$: Gumbel (Thm 6)	LEADINGONES $m = O(1)$: Mixture of Gamma (Thm 8) $m \to \infty$: Normal (Thm 10)
$O(1)$	$\mathbb{P}\left(\frac{X_{n,m}}{en} \leqslant x\right) \to (1-e^{-x})^m$	$\mathbb{P}\left(\frac{Y_{n,m}}{en} \leqslant x\right) \to \sum_{0 \leqslant j < m} \frac{\binom{m-1}{j}}{2^{m-1}} \int_0^x e^{-t} \frac{t^j}{j!}\, \mathrm{d}t$
$\to \infty$ $\leqslant n$	$\mathbb{P}\left(\frac{X_{n,m}}{en} - \log m - \phi_1(\frac{m}{n}) \leqslant x\right)$ $\to e^{-e^{-x}}$	$\mathbb{P}\left(\frac{Y_{n,m} - \nu_{n,m}}{\varsigma_{n,m}} \leqslant x\right) \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}}\, \mathrm{d}t$

Table 1: *The limit laws for the number of stages used by Algorithm* $(1+1)$*-EA under* ONEMAX *(*$X_{n,m}$*) and* LEADINGONES *(*$Y_{n,m}$*) fitness function, respectively, when starting from the initial state with the evaluation* $n-m$*. The function* ϕ_1 *is defined in* (3)*, and the two quantities* $\nu_{n,m}$ *and* $\varsigma_{n,m}$ *are given in* (61) *and* (62)*, respectively.*

2 Recurrence and the limit laws when $m = O(1)$

Recall that we start from the initial state $f(\mathbf{x}) = n - m$ and that $X_{n,m}$ denotes the number of steps used by $(1+1)$-EA before reaching $f(\mathbf{x}) = n$. We derive first a recurrence relation satisfied by the probability generating function $P_{n,m}(t) := \mathbb{E}(t^{X_{n,m}})$ of $X_{n,m}$.

Lemma 1. *The probability generating function $P_{n,m}(t)$ satisfies the recurrence*

$$P_{n,m}(t) = \frac{t \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} P_{n,m-\ell}(t)}{1 - \left(1 - \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}\right) t} \qquad (1 \leqslant m \leqslant n), \tag{4}$$

for $1 \leqslant m \leqslant n$, with $P_{n,0}(t) = 1$, where

$$\lambda_{n,m,\ell} := \left(1 - \frac{1}{n}\right)^n (n-1)^{-\ell} \sum_{0 \leqslant j \leqslant \min\{n-m, m-\ell\}} \binom{n-m}{j} \binom{m}{j+\ell} (n-1)^{-2j}. \tag{5}$$

Proof. Start from the state $f(\mathbf{x}) = n - m$ and run the two steps inside the loop of Algorithm $(1+1)$-EA. The new state becomes $\mathbf{y}$ with $f(\mathbf{y}) = n - m + \ell$ if j bits in the group $\{x_i = 1\}$ and $j + \ell$ bits in the other group $\{x_i = 0\}$ toggled their values, where $0 \leqslant j \leqslant \max\{n-m, m-\ell\}$ and $\ell > 0$. Thus, the probability from state $\mathbf{x}$ to $\mathbf{y}$ is given by

$$\lambda_{n,m,\ell} = \sum_{0 \leqslant j \leqslant \min\{n-m, m-\ell\}} \binom{n-m}{j} \left(\frac{1}{n}\right)^j \left(1 - \frac{1}{n}\right)^{n-m-j} \binom{m}{j+\ell} \left(\frac{1}{n}\right)^{j+\ell} \left(1 - \frac{1}{n}\right)^{m-j-\ell},$$

which is identical to (5). We then obtain

$$P_{n,m}(t) = t \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} P_{n,m-\ell}(t) + \left(1 - \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}\right) t P_{n,m}(t),$$

and this proves the lemma. ∎

While this simple recurrence relation seems not new in the EA literature, tools have been lacking for a direct asymptotic resolution, which we will develop in detail in this paper.

For convenience, define

$$\Lambda_{n,m} := \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}.$$

In particular, when $m = 1$,

$$\Lambda_{n,1} = \lambda_{n,1,1} = \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1},$$

so that

$$P_{n,1}(t) = \frac{\frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} t}{1 - \left(1 - \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1}\right) t}.$$

This is a standard geometric distribution Geo(ρ) with probability $\rho = \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$ (assuming only positive integer values). Obviously, taking $t = e^{\frac{s}{en}}$, we obtain

$$P_{n,1}\left(e^{\frac{s}{en}}\right) = \frac{1}{1-s}\left(1 + O\left(\frac{1}{n|1-s|}\right)\right),$$

as $n \to \infty$, uniformly for $|s| \leqslant 1-\varepsilon$, implying, by Curtiss's convergence theorem (see [19, §5.2.3]), the *convergence in distribution*

$$\frac{X_{n,1}}{en} \xrightarrow{(d)} \mathrm{Exp}(1),$$

where Exp(c) denotes an exponential distribution with parameter c. Equivalently, this can be rewritten as

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{X_{n,1}}{en} \leqslant x\right) = 1 - e^{-x},$$

for $x > 0$. Such a limit law indeed extends to the case when $m = O(1)$, which we formulate as follows.

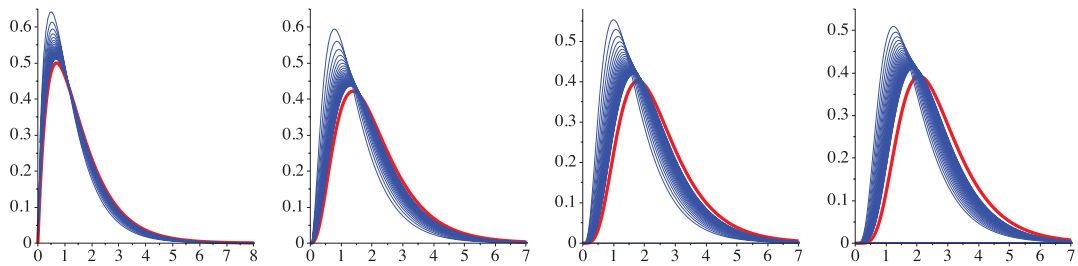


Figure 1: *Histograms of $X_{n,2j}/en$ for $j = 1, \ldots, 4$ (in left to right order) and $n = 5, \ldots, 50$, and their corresponding limit laws.*

Let $H_m^{(i)} = \sum_{1\leqslant j\leqslant m} j^{-i}$ denote the i-th order harmonic numbers and $H_m = H_m^{(1)}$. For convenience, we define $H_0^{(i)} = 0$.

Theorem 1. *If $m = O(1)$, the time used by $(1+1)$-EA to reach the optimum state $f(\mathbf{x}) = n$, when starting from $f(\mathbf{x}) = n-m$, converges, when normalized by en, to a sum of m exponential random variables*

$$\frac{X_{n,m}}{en} \xrightarrow{(d)} \sum_{1\leqslant r\leqslant m} \mathrm{Exp}(r), \tag{6}$$

with mean asymptotic to $eH_m n$ and variance asymptotic to $e^2 H_m^{(2)} n^2$.

The convergence in distribution (6) can be expressed alternatively as

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{X_{n,m}}{en} \leqslant x\right) = (1-e^{-x})^m \qquad (x > 0).$$

Proof. By the sum definition of $\lambda_{n,m,\ell}$, we see that

$$\begin{aligned}\lambda_{n,m,\ell} &= \left(1-\frac{1}{n}\right)^n (n-1)^{-\ell} \sum_{0\leqslant j\leqslant \min\{n-m,m-\ell\}} \binom{m}{j+\ell}\binom{n-m}{j}(n-1)^{-2j} \\ &= \binom{m}{\ell} e^{-1} n^{-\ell}\left(1+O\left(\frac{(n-m)(m-\ell)}{n^2\ell}\right)\right), \end{aligned} \tag{7}$$

where the O-term holds uniformly for $1 \leqslant m = o(n)$ and $1 \leqslant \ell \leqslant m$. In particular, for each fixed $m = O(1)$, we then have

$$P_{n,m}(t) = \frac{\frac{m}{en}t}{1-\left(1-\frac{m}{en}\right)t} P_{n,m-1}(t)(1+o(1)),$$

so that

$$P_{n,m}(t) = \left(\prod_{1\leqslant r\leqslant m} \frac{\frac{r}{en}t}{1-\left(1-\frac{r}{en}\right)t}\right)(1+o(1)),$$

where both $o(1)$-terms are uniform for $|s| \leqslant 1-\varepsilon$. Now take $t = e^{\frac{s}{en}}$. Then

$$P_{n,m}\left(e^{\frac{s}{en}}\right) = \left(\prod_{1\leqslant r\leqslant m} \frac{1}{1-\frac{s}{r}}\right)(1+o(1)), \tag{8}$$

uniformly for $|s| \leqslant 1-\varepsilon$. This and Curtiss's convergence theorem (see [19, §5.2.3]) imply (6). The asymptotic mean and the asymptotic variance can be computed either by a similar inductive argument or by following ideas used in the Quasi-Power Framework (see [15] or [22]) that relies on the *uniformity* of the estimate (8)

$$\frac{\mathbb{E}(X_{n,m})}{en} = [s]P_{n,m}\left(e^{\frac{s}{en}}\right) \sim [s]\prod_{1\leqslant r\leqslant m}\frac{1}{1-\frac{s}{r}} = H_m,$$

where $[s^k]f(s)$ denotes the coefficient of s^k in the Taylor expansion of $f(s)$, and

$$\begin{aligned}\frac{\mathbb{V}(X_{n,m})}{(en)^2} &= 2[s^2]\log P_{n,m}\left(e^{\frac{s}{en}}\right) \\ &\sim 2[s^2]\sum_{1\leqslant r\leqslant m}\log\frac{1}{1-\frac{s}{r}} \\ &= H_m^{(2)}.\end{aligned}$$

We will derive more precise expansions below by a direct approach. ∎

The simple inductive argument fails when $m \to \infty$ and we need more uniform estimates for the error terms.

3 Asymptotics of sums of the form $\sum_{1\leqslant \ell\leqslant m} a_\ell \lambda_{n,m,\ell}$

Sums of the form

$$\sum_{1\leqslant \ell\leqslant m} a_\ell \lambda_{n,m,\ell}$$

appear frequently in our analysis. We thus digress in this section to develop tools for deriving the asymptotic behaviors of such sums.

For technical simplicity, we define the sequence $e_n := \left(1-\frac{1}{n+1}\right)^{n+1}$ and the normalized sum

$$\lambda^*_{n,m,\ell} := \frac{\lambda_{n+1,m,\ell}}{e_n} = \sum_{0\leqslant j\leqslant \min\{n+1-m,m-\ell\}} \binom{n+1-m}{j}\binom{m}{j+\ell} n^{-\ell-2j}. \tag{9}$$

Let also

$$A^*_{n,m} := \sum_{1\leqslant \ell\leqslant m} a_\ell \lambda^*_{n,m,\ell}.$$

Throughout this paper, we use the abbreviation

$$\alpha := \frac{m}{n}.$$

Asymptotics of $A^*_{n,m}$. Observe that most contribution to $A_{n,m}$ comes from small ℓ, say $\ell = o(m)$, provided that a_ℓ does not grow too fast; see (7). We formulate a more precise version as follows.

Lemma 2. *Assume that $\{a_\ell\}_{\ell\geqslant 1}$ is a given sequence such that $A(z) = \sum_{\ell\geqslant 1} a_\ell z^{\ell-1}$ has a nonzero radius of convergence in the z-plane. Then*

$$A^*_{n,m} = \tilde{A}_0(\alpha) + \frac{\tilde{A}_1(\alpha)}{n} + O\left(\alpha n^{-2}\right), \tag{10}$$

where $\tilde{A}_0(\alpha)$ and $\tilde{A}_1(\alpha)$ are entire functions of α defined by

$$\tilde{A}_0(\alpha) := \sum_{\ell\geqslant 1} \frac{\alpha^\ell}{\ell!} \sum_{0\leqslant j<\ell} a_{\ell-j} \frac{(1-\alpha)^j}{j!}, \tag{11}$$

and ($a_0 := 0$)

$$\tilde{A}_1(\alpha) := -\frac{1}{2}\sum_{\ell\geqslant 1} \frac{\alpha^\ell}{\ell!} \sum_{0\leqslant j<\ell} \frac{(1-\alpha)^j}{j!}\left((\ell-j)a_{\ell+1-j} - (\ell+2-j)a_{\ell-1-j} + a_{\ell-j}\right). \tag{12}$$

Proof. The first term on the right-hand side of (10) can be readily obtained as follows. If $1\leqslant m\leqslant n$, then

$$\begin{aligned} A^*_{n,m} &= \sum_{j\geqslant 0}\binom{n+1-m}{j} n^{-j} \sum_{j<\ell\leqslant m} a_{\ell-j}\binom{m}{\ell} n^{-\ell} \\ &\sim \sum_{j\geqslant 0} \frac{(1-\alpha)^j}{j!} \sum_{\ell>j} a_{\ell-j}\frac{\alpha^\ell}{\ell!} \\ &= \tilde{A}_0(\alpha). \end{aligned}$$

The more precise approximation in (10) can be obtained by refining all estimates, but the details are rather messy, notably the error analysis. We resort instead to an analytic approach. Observe that the sum on the left-hand side of (10) is itself a convolution. Our analytic proof then starts from the relation

$$\begin{aligned}\lambda^*_{n,m,\ell} &= [z^{m-\ell}]\left(z+\frac{1}{n}\right)^m\left(1+\frac{z}{n}\right)^{n+1-m}\\ &= \frac{1}{2\pi i}\oint_{|z|=c} z^{\ell-1}\left(1+\frac{1}{nz}\right)^m\left(1+\frac{z}{n}\right)^{n+1-m}\,\mathrm{d}z,\end{aligned}\tag{13}$$

where $c>0$. The relation (13) holds *a priori* for $1\leqslant \ell\leqslant m$, but the right-hand side becomes zero for $\ell>m$. It follows that

$$A^*_{n,m} = \frac{1}{2\pi i}\oint_{|z|=c} A(z)\left(1+\frac{1}{nz}\right)^m\left(1+\frac{z}{n}\right)^{n+1-m}\,\mathrm{d}z,$$

where $0<c<\varrho$, ϱ being the radius of convergence of A. By the expansion

$$\begin{aligned}&\left(1+\frac{1}{nz}\right)^m\left(1+\frac{z}{n}\right)^{n+1-m}\\ &\quad= e^{\frac{\alpha}{z}+(1-\alpha)z}\left(1-\frac{1}{2n}\left((1-\alpha)z^2-2z+\frac{\alpha}{z^2}\right)+O\left(\frac{(1-\alpha)^2|z|^4+\alpha^2|z|^{-4}}{n^2}\right)\right),\end{aligned}$$

uniformly for z on the integration path, and the integral representations

$$\begin{aligned}\tilde{A}_0(\alpha) &= \frac{1}{2\pi i}\oint_{|z|=c} A(z)e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z\\ \tilde{A}_1(\alpha) &= -\frac{1}{4\pi i}\oint_{|z|=c} A(z)\left((1-\alpha)z^2-2z+\frac{\alpha}{z^2}\right)e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z,\end{aligned}$$

we deduce (10). The expression (11) is then obtained by straightforward term-by-term integration. For (12), we apply the relation

$$\frac{\mathrm{d}}{\mathrm{d}z}e^{\frac{\alpha}{z}+(1-\alpha)z} = \left(-\frac{\alpha}{z^2}+1-\alpha\right)e^{\frac{\alpha}{z}+(1-\alpha)z},$$

and integration by parts, and then obtain

$$\tilde{A}_1(\alpha) = -\frac{1}{4\pi i}\oint_{|z|=c}\left((1-z^2)A'(z)+(1-4z)A(z)\right)e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z.\tag{14}$$

Substituting the series expansion $A(z)=\sum_{\ell\geqslant 1}a_\ell z^{\ell-1}$ and integrating term by term, we get (12). ∎

When α tends to the two boundaries 0 and 1, we have

$$A^*_{n,m}\sim\tilde{A}_0(\alpha)\sim\begin{cases}\dfrac{a_k}{k!}\alpha^k, & \text{as } \alpha\to 0^+,\\[2ex] \displaystyle\sum_{\ell\geqslant 1}\frac{a_\ell}{\ell!}, & \text{as } \alpha\to 1^-,\end{cases}$$

where k is the smallest integer such that $a_k\neq 0$.

Asymptotics of $\sum_{1\leqslant \ell\leqslant m} \ell^r \lambda^*_{n,m,\ell}$ We now discuss special sums of the form

$$\Lambda^{(r)}_{n,m} := \sum_{1\leqslant \ell\leqslant m} \ell^r \lambda_{n,m,\ell},$$

which will be repeatedly encountered below. Define

$$\bar{\Lambda}^{(r)}_{n,m} := \sum_{1\leqslant \ell\leqslant m} \ell^r \lambda^*_{n,m,\ell},$$

so that $\Lambda^{(r)}_{n,m} = e_n \bar{\Lambda}^{(r)}_{n-1,m}$. For convenience, we also write

$$\bar{\Lambda}_{n,m} := \bar{\Lambda}^{(0)}_{n,m} = \sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}. \tag{15}$$

Let I_k denote the modified Bessel functions

$$I_k(2z) := \sum_{j\geqslant 0} \frac{z^{2j+k}}{j!(j+k)!} \qquad (k \in \mathbb{Z}).$$

Corollary 1. *Uniformly for* $1 \leqslant m \leqslant n$

$$\bar{\Lambda}^{(r)}_{n,m} = S_r(\alpha) + \frac{U_r(\alpha)}{n} + O\left(\alpha n^{-2}\right), \tag{16}$$

for $r = 0, 1, \ldots,$ *where both* S_r *and* U_r *are entire functions given by*

$$S_r(z) = \sum_{\ell\geqslant 1} \frac{z^\ell}{\ell!} \sum_{0\leqslant j<\ell} (\ell - j)^r \frac{(1-z)^j}{j!},$$

and

$$U_r(\alpha) = \begin{cases} \dfrac{S_0(\alpha)}{2} - \dfrac{3}{2}\sqrt{\dfrac{\alpha}{1-\alpha}}\, I_1\left(2\sqrt{\alpha(1-\alpha)}\right), & \text{if } r = 0 \\ -\dfrac{1}{2}\left((2r-1)S_r(\alpha) + \displaystyle\sum_{0\leqslant j<r} \binom{r}{j} \frac{j - (-1)^{r-j}(2r+2-3j)}{r+1-j} S_j(\alpha)\right), & \text{if } r \geqslant 1. \end{cases} \tag{17}$$

In particular,

$$\begin{aligned} U_1(\alpha) &= -S_0(\alpha) - \tfrac{1}{2} S_1(\alpha) \\ U_2(\alpha) &= S_0(\alpha) - 2S_1(\alpha) - \tfrac{3}{2} S_2(\alpha) \\ U_3(\alpha) &= -S_0(\alpha) + 2S_1(\alpha) - 3S_2(\alpha) - \tfrac{5}{2} S_3(\alpha). \end{aligned} \tag{18}$$

These are sufficient for our uses.

Proof. We start with the integral representation (see (14))

$$U_r(\alpha) = -\frac{1}{4\pi i}\oint_{|z|=c} \left((1-z^2)E_r'(z) + (1-4z)E_r(z)\right) e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z,$$

where $E_r(z) := \sum_{\ell\geqslant 1} \ell^r z^{\ell-1}$. When $r = 0$, we have $E_0(z) = (1-z)^{-1}$. Thus

$$U_0(\alpha) = -\frac{1}{4\pi i}\oint_{|z|=c} \left(-\frac{1}{1-z} + 3\right) e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z.$$

Note that

$$S_r(\alpha) = \frac{1}{2\pi i}\oint_{|z|=c} E_r(z) e^{\frac{\alpha}{z}+(1-\alpha)z}\,\mathrm{d}z \qquad (r = 0, 1, \dots). \tag{19}$$

Thus

$$U_0(\alpha) = \frac{S_0(\alpha)}{2} - \frac{3}{2}\sum_{\ell\geqslant 1} \frac{\alpha^\ell (1-\alpha)^{\ell-1}}{\ell!(\ell-1)!},$$

which proves (17) for $r = 0$. For $r \geqslant 1$, we have

$$\begin{aligned}
&(1-z^2)E_r'(z) + (1-4z)E_r(z) \\
&\quad = (1-z^2)\sum_{\ell\geqslant 2} \ell^r(\ell-1)z^{\ell-2} + (1-4z)\sum_{\ell\geqslant 1} \ell^r z^{\ell-1} \\
&\quad = \sum_{\ell\geqslant 1} \ell(\ell+1)^r z^{\ell-1} - \sum_{\ell\geqslant 2} (\ell+2)(\ell-1)^r z^{\ell-1} + E_r(z) \\
&\quad = \sum_{0\leqslant j\leqslant r} \binom{r}{j} E_{j+1}(z) - \sum_{0\leqslant j\leqslant r} \binom{r}{j} (-1)^{r-j}\left(E_{j+1}(z) + 2E_j(z)\right) + E_r(z).
\end{aligned}$$

From this and the relation (19), we obtain (17). Note that the coefficient of E_{r+1} is zero. ∎

The Corollary implies specially that

$$\Lambda_{n,m}^{(r)} = e^{-1} S_r(\alpha)\left(1 + O\left(n^{-1}\right)\right), \tag{20}$$

uniformly for $1 \leqslant m \leqslant n$ and $r \geqslant 0$. Since $S_r(z) = z + O(|z|^2)$ as $|z| \to 0$, we have the uniform bound

$$\bar{\Lambda}_{n,m}^{(r)} \asymp S_r(\alpha) \asymp \alpha \qquad (1 \leqslant m \leqslant n), \tag{21}$$

meaning that the ratio of $\bar{\Lambda}_{n,m}^{(r)}/\alpha$ remains bounded away from zero and infinity for all m in the specified range.

We also have the limiting behaviors

$$\lim_{\alpha\to 0} \frac{S_r(\alpha)}{\alpha} = 1,$$

and

$$\lim_{\alpha\to 1} S_r(\alpha) = \sum_{\ell\geqslant 1} \frac{\ell^r}{\ell!} = \{e-1, e, 2e, 5e, 15e, \cdots\}.$$

Without the first term, the right-hand side is, up to e, the Bell numbers (all partitions of a set; Sequence A000110 in Sloane's Encyclopedia of Integer Sequences).

The following expansions for $S_r(z)$ and $U_r(z)$ as $z \to 0$ will be used later

$$\begin{aligned} S_r(z) &= z + \frac{2^r+1}{2} z^2 + O(z^3), \\ U_r(z) &= -\frac{2^r+1}{2} z + O(z^2), \end{aligned} \tag{22}$$

for $r = 0, 1, \ldots$.

See also Appendix A for other properties of $S_r(\alpha)$.

4 The expected values and their asymptotics

Consider the mean $\mu_{n,m} := \mathbb{E}(X_{n,m}) = P'_{n,m}(1)$. It satisfies the recurrence

$$\mu_{n,m} = \frac{1}{\Lambda_{n,m}} \left(1 + \sum_{1\leqslant \ell\leqslant m} \lambda_{n,m,\ell}\, \mu_{n,m-\ell} \right),$$

for $1 \leqslant m \leqslant n$ with $\mu_{n,0} = 0$.

From Theorem 1, we already have $\mu_{n,m} \sim enH_m$ when $m = O(1)$, while for $m \to \infty$ and $m \leqslant n$ we expect that (recalling $\alpha = \frac{m}{n}$)

$$\mu_{n,m} \sim en(H_m + \phi_1(\alpha));$$

see Section 4.1 for how such a form arises. We will indeed derive in this section a more precise expansion. The uniform appearance of the harmonic numbers H_m may be traced to the asymptotic estimate (7); see also Lemma 3.

Theorem 2. *The expected value of $X_{n,m}$ satisfies the asymptotic approximation*

$$\frac{\mathbb{E}(X_{n,m})}{en} = H_m + \phi_1(\alpha) + \frac{H_m - \phi_1(\alpha) + 2\phi_2(\alpha) + 2\alpha\phi_1'(\alpha)}{2n} + O\left(n^{-2} H_m\right), \tag{23}$$

uniformly for $1 \leqslant m \leqslant n$, where ϕ_1 is defined in (3) and ϕ_2 is an analytic function defined by

$$\phi_2(\alpha) = \frac{1}{2} - \int_0^\alpha \left(\frac{S_2(x)S_1'(x)}{2S_1(x)^3} - \frac{S_0(x)}{S_1(x)^2} - \frac{1}{2S_1(x)} - \frac{1}{2x^2} + \frac{1}{x} \right) \mathrm{d}x. \tag{24}$$

For simplicity, we consider

$$\mu^*_{n,m} := \frac{e_n}{n}\, \mu_{n+1,m},$$

where $e_n := \left(1 - \frac{1}{n+1}\right)^{n+1}$, and we will prove that

$$\mu^*_{n,m} = H_m + \phi_1(\alpha) + \frac{H_m + \phi_2(\alpha)}{n} + O\left(n^{-2} H_m\right), \tag{25}$$

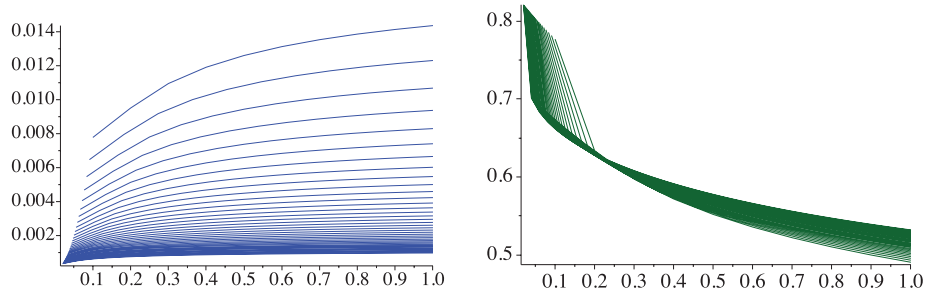


Figure 2: *The differences* $\mu^*_{n,m} - (H_m + \phi_1(\alpha) + \frac{H_m+\phi_2(\alpha)}{n})$ *for* $1 \leqslant m \leqslant n$ *(normalized to the unit interval) and* $n = 10, \ldots, 50$ *(left in top-down order), and the normalized differences* $(\mu^*_{n,m} - (H_m + \phi_1(\alpha) + \frac{H_m+\phi_2(\alpha)}{n}))n^2/H_m$ *for* $n = 10, \ldots, 50$ *(right).*

for $1 \leqslant m \leqslant n$, which is identical to (23); see Figure 2 for a graphical rendering. More figures are collected in Appendix B.

Our analysis will be based on the recurrence

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right) = \frac{1}{n}, \tag{26}$$

or alternatively

$$\mu^*_{n,m} = \frac{1}{\bar{\Lambda}_{n,m}}\left(\frac{1}{n} + \sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\,\mu^*_{n,m-\ell}\right),$$

with $\mu^*_{n,0} = 0$, where $\lambda^*_{n,m,\ell}$ and $\bar{\Lambda}_{n,m}$ are defined in (9) and (15), respectively. In particular, this gives $\mu^*_{n,1} = 1$,

$$\begin{aligned}\mu^*_{n,2} &= \frac{3n^2+n-1}{2n^2+2n-1},\\ \mu^*_{n,3} &= \frac{22n^6+40n^5-19n^4-42n^3+14n^2+15n-6)}{(2n^2+2n-1)(6n^4+12n^3-7n^2-9n+6)}.\end{aligned} \tag{27}$$

In general, the $\mu^*_{n,m}$ are all rational functions of n but their expressions become long as m increases. In Section 4.1, we give asymptotic expansions for $\mu^*_{n,m}$ for small values of m as $n \to \infty$, which are also required as initial values for obtaining more refined asymptotic expansions for other ranges of m in Section 4.4.

Starting from the asymptotic estimate $\mu^*_{n,m} \sim H_m$ when $m = O(1)$, we first postulate an Ansatz approximation of the form

$$\mu^*_{n,m} \sim H_m + \phi(\alpha) \qquad (1 \leqslant m \leqslant n), \tag{28}$$

for some smooth function ϕ. Then we will justify such an expansion by an error analysis relying on Lemma 3 after a proper choice of ϕ. This same procedure can then be extended and yields a more precise expansion; see Section 4.4.

Instead of starting from a state with a fixed number of ones, the first step of the Algorithm (1+1)-EA described in Introduction corresponds to the situation when the initial state $f(\mathbf{x})$ (the

number of 1s) is not fixed but random. Assume that this input follows a binomial distribution of parameter $1-\rho \in (0,1)$ (each bit being 1 with probability $1-\rho$ and 0 with probability ρ). Denote by X_n the number of steps used by $(1+1)$-EA to reach the optimum state. Such a situation can also be dealt with by applying Theorem 6 and we obtain the same limit law. The following result describes precisely the asymptotic behavior of the expected optimization time.

Theorem 3. *The expected value of X_n satisfies*

$$\frac{\mathbb{E}(X_n)}{en} = \log \rho n + \gamma + \phi_1(\rho) + \frac{\log \rho n + \gamma + 1 - \phi_1(\rho) + 2\rho\phi_1'(\rho) + \rho(1-\rho)\phi_1''(\rho) + 2\phi_2(\rho)}{2n} + O\left(\frac{\log n}{n^2}\right).$$

Note that $e(\log \rho + \gamma + \phi_1(\rho))$ is an increasing function of ρ, which is consistent with the intuition that it takes less steps to reach the final state if we start with more 1s (small ρ means $1-\rho$ closer to 1, or 1 occurring with higher probability). Also

$$1 + 2\rho\phi_1'(\rho) + \rho(1-\rho)\phi_1''(\rho) = -2 + \frac{1}{\rho} + \frac{2\rho}{S_1(\rho)} - \rho(1-\rho)\frac{S_1'(\rho)}{S_1(\rho)^2}.$$

The constant c_2 in (2) can now be computed and has the value

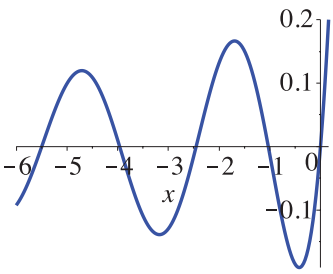
$$c_2 = \frac{e}{2}\left(-\log 2 + \gamma - \phi_1(\tfrac{1}{2}) + 2\phi_2(\tfrac{1}{2}) + \frac{1}{S_1(\frac{1}{2})} - \frac{S_1'(\frac{1}{2})}{4S_1(\frac{1}{2})^2}\right) \approx 0.59789875\ldots. \tag{29}$$

Numerically, to compute the value of $\phi_1(\alpha)$ for $\alpha \in (0,1]$, the most natural way consists in using the Taylor expansion

$$\frac{1}{S_1(x)} - \frac{1}{x} = \sum_{j\geqslant 0} \sigma_j x^j,$$

and after a term-by-term integration

$$\phi_1(\alpha) = \sum_{j\geqslant 0} \frac{\sigma_j}{j+1}\alpha^{j+1}.$$



$S_1(x)$ has an infinity number of zeros on $\mathbb{R}^-$.

While $S_1(x)$ is an entire functions with rapidly decreasing coefficients, such an expansion converges slowly when $\alpha \sim 1$, the main reason being that the smallest $|x| > 0$ for which $S_1(x) = 0$ occurs when $x \approx -1.0288$, implying that the radius of convergence of this series is slightly larger than unity. Note that $S_1(0) = 0$ but the simple pole is removed by subtracting $\frac{1}{x}$. A better idea is then expanding $\frac{1}{S_1(x)} - \frac{1}{x}$ at $x = 1$ and integrating term-by-term

$$\phi_1(\alpha) = \sum_{j\geqslant 0} \frac{\sigma_j'}{j+1}\left(1-(1-\alpha)^{j+1}\right) \quad \text{where} \quad \frac{1}{S_1(1-x)} - \frac{1}{1-x} = \sum_{j\geqslant 0}\sigma_j' x^j.$$

This expansion is numerically more efficient and stable because of better convergence for $\alpha \in [0,1]$. The same technique also applies to the calculation of ϕ_2 and other functions in this paper.

4.1 Asymptotic expansions for small m

Our asymptotic approximation (25) to $\mu^*_{n,m}$ was largely motivated by intensive symbolic computations for small m. We briefly summarize them here, which will also be crucial in specifying the initial conditions for the differential equations satisfied by functions $(\phi_1, \phi_2, \dots)$ involved in the full asymptotic expansion of $\mu^*_{n,m}$; see (41).

Starting from the closed-form expressions (27), we readily obtain $\mu^*_{n,0} = 0$, $\mu^*_{n,1} = 1$, and

$$\begin{aligned}
\mu^*_{n,2} &= \tfrac{3}{2} - n^{-1} + \tfrac{5}{4} n^{-2} - \tfrac{7}{4} n^{-3} + \tfrac{19}{8} n^{-4} - \tfrac{13}{4} n^{-5} + O(n^{-6}),\\
\mu^*_{n,3} &= \tfrac{11}{6} - \tfrac{13}{6} n^{-1} + \tfrac{155}{36} n^{-2} - \tfrac{323}{36} n^{-3} + \tfrac{4007}{216} n^{-4} - \tfrac{2783}{72} n^{-5} + O(n^{-6}).
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\mu^*_{n,4} &= \tfrac{25}{12} - \tfrac{41}{12} n^{-1} + \tfrac{329}{36} n^{-2} - \tfrac{917}{36} n^{-3} + \tfrac{61841}{864} n^{-4} - \tfrac{19501}{96} n^{-5} + O(n^{-6}),\\
\mu^*_{n,5} &= \tfrac{137}{60} - \tfrac{283}{60} n^{-1} + \tfrac{2839}{180} n^{-2} - \tfrac{19859}{360} n^{-3} + \tfrac{848761}{4320} n^{-4} - \tfrac{5107063}{7200} n^{-5} + O(n^{-6}).
\end{aligned}$$

From these expansions, we first observe that the leading sequence is exactly H_m ($H_0 := 0$)

$$\{H_m\}_{m\geqslant 0} = \left\{0, 1, \tfrac{3}{2}, \tfrac{11}{6}, \tfrac{25}{12}, \tfrac{137}{60}, \tfrac{49}{20}, \cdots\right\}.$$

These also suggest the following Ansatz

$$\mu^*_{n,m} \approx \sum_{k\geqslant 0} \frac{d_k(m)}{n^k},$$

for some functions $d_k(m)$ of m. Using this form and the above expansions to match the undetermined coefficients of the polynomials (in m), we obtain successively

$$\begin{aligned}
d_0(m) &= H_m \quad (m \geqslant 0),\\
d_1(m) &= H_m + \tfrac{1}{2} - \tfrac{3}{2} m \quad (m \geqslant 1),\\
d_2(m) &= \tfrac{2}{3} H_m + \tfrac{1}{12} - \tfrac{7}{4} m + \tfrac{11}{12} m^2 \quad (m \geqslant 2),\\
d_3(m) &= \tfrac{1}{2} H_m + \tfrac{7}{24} - \tfrac{575}{432} m + \tfrac{23}{18} m^2 - \tfrac{283}{432} m^3, \quad (m \geqslant 2),\\
d_4(m) &= \tfrac{5}{18} H_m - \tfrac{59}{720} - \tfrac{3439}{3456} m + \tfrac{15101}{11520} m^2 - \tfrac{19951}{17280} m^3 + \tfrac{5759}{11520} m^4, \quad (m \geqslant 4).
\end{aligned}$$

So we observe the general pattern

$$\mu^*_{n,m} \approx \sum_{k\geqslant 0} \frac{1}{n^k} \left(b_k H_m + \sum_{0\leqslant j\leqslant k} \varpi_{k,j} m^j \right),$$

for some explicitly computable sequence b_k and coefficients $\varpi_{k,j}$. A crucial complication arises here: the general form for each $d_k(m)$ holds only for $m \geqslant 2\lfloor \frac{k}{2} \rfloor$, and correction terms are needed for smaller m. For example,

$$\begin{aligned}
d_1(m) &= H_m + \tfrac{1}{2} - \tfrac{3}{2} m - \tfrac{1}{2} [\![m = 0]\!], \quad (m \geqslant 0)\\
d_2(m) &= \tfrac{2}{3} H_m + \tfrac{1}{12} - \tfrac{7}{4} m + \tfrac{11}{12} m^2 - \tfrac{1}{12} [\![m = 0]\!] + \tfrac{1}{12} [\![m = 1]\!], \quad (m \geqslant 0),
\end{aligned}$$

where we use the Iverson bracket notation $[\![A]\!] = 1$ if A holds, and 0, otherwise. It is such a complication that makes the determination of smaller-order terms more involved.

All the expansions here hold only for small m. When m grows, we see that

$$n^{-k} \sum_{0 \leqslant j \leqslant k} \varpi_{k,j} m^j = \varpi_{k,k} \alpha^k + \varpi_{k,k-1} \frac{\alpha^{k-1}}{n} + \text{smaller order terms},$$

and it is exactly this form that motivated naturally our choice of the Ansatz (25).

4.2 More asymptotic tools

We develop here some other asymptotic tools that will be used in proving Theorem 2.

The following lemma is very helpful in obtaining error estimates to be addressed below. It also sheds new light on the occurrence of the harmonic numbers H_m in (25).

Lemma 3. *Consider the recurrence*

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} (a_{n,m} - a_{n,m-\ell}) = b_{n,m} \qquad (m \geqslant 1),$$

where $b_{n,m}$ is defined for $1 \leqslant m \leqslant n$ and $n \geqslant 1$. Assume that $|a_{n,0}| \leqslant d$ for $n \geqslant 1$, where $d \geqslant 0$. If $|b_{n,m}| \leqslant \frac{c}{n}$ holds uniformly for $1 \leqslant m \leqslant n$ and $n \geqslant 1$, where $c > 0$, then

$$|a_{n,m}| \leqslant c H_m + d \qquad (0 \leqslant m \leqslant n).$$

Proof. The result is true for $m = 0$. For $m \geqslant 1$, we start from the simple inequality

$$\bar{\Lambda}_{n,m} = \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \geqslant \frac{m}{n} \qquad (1 \leqslant m \leqslant n),$$

because all terms in the sum expression (9) are positive and taking only one term ($j = 0$ and $\ell = 1$) gives the lower bound. Then, by the induction hypothesis,

$$\begin{aligned} |a_{n,m}| &\leqslant \frac{|b_{n,m}|}{\bar{\Lambda}_{n,m}} + |a_{n,m-1}| \\ &\leqslant \frac{c}{n} \cdot \frac{n}{m} + c H_{m-1} + d \\ &= c H_m + d, \end{aligned}$$

proving the lemma. ∎

Applying this lemma to the recurrence (26), we then get a simple upper bound to $\mu^*_{n,m}$.

Corollary 2. *For $0 \leqslant m \leqslant n$, the inequality*

$$\mu^*_{n,m} \leqslant H_m$$

holds.

Lemma 4. *If ϕ is a $C^2[0,1]$-function, then*

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\phi\left(\frac{m}{n}\right)-\phi\left(\frac{m-\ell}{n}\right)\right)=\frac{\phi'(\alpha)}{n}\sum_{1\leqslant \ell\leqslant m}\ell\lambda^*_{n,m,\ell}+O\left(n^{-2}\right),$$

uniformly for $1\leqslant m\leqslant n$.

Proof. A direct Taylor expansion with remainder gives

$$\phi(\alpha)-\phi\left(\alpha-\tfrac{\ell}{n}\right)=\phi'(\alpha)\tfrac{\ell}{n}+O\left(\ell^2n^{-2}\right),$$

uniformly for $1\leqslant \ell\leqslant m$, since $\phi''(t)=O(1)$ for $t\in[0,1]$. The lemma follows from the estimates (16). ∎

The approximation can be easily extended and refined if more smoothness properties of ϕ are known, which is the case for all functions appearing in our analysis (they are all $C^\infty[0,1]$).

Another standard technique we need is Stirling's formula for the factorials

$$\log n!=\log\Gamma(n+1)=\left(n+\tfrac{1}{2}\right)\log n-n+\tfrac{1}{2}\log(2\pi)+\tfrac{1}{12}n^{-1}+O(n^{-3}), \tag{30}$$

where Γ denotes Euler's Gamma function.

4.3 Proof of Theorem 2

Formal calculus. Applying formally (28) and Lemma 4 using $H_m-H_{m-\ell}\sim\frac{\ell}{m}$ and $\phi(\frac{m}{n})-\phi(\frac{m-\ell}{n})\sim\phi'(\alpha)\frac{\ell}{n}$, we have

$$\begin{aligned}\frac{1}{n}=\sum_{1\leqslant \ell\leqslant m}\lambda^*_{n,m,\ell}\left(\mu^*_{n,m}-\mu^*_{n,m-\ell}\right)&\sim\sum_{1\leqslant \ell\leqslant m}\lambda^*_{n,m,\ell}\left(\frac{\ell}{m}+\phi'(\alpha)\frac{\ell}{n}\right)\\&\sim\frac{1}{n}\left(\frac{1}{\alpha}+\phi'(\alpha)\right)S_1(\alpha),\end{aligned}$$

by (16). Thus we see that ϕ satisfies

$$\phi'(z)=\frac{1}{S_1(z)}-\frac{1}{z}.$$

We now specify the initial condition $\phi(0)$. Since the postulated form (28) holds for $1\leqslant m\leqslant n$ (indeed also true for $m=0$), we take $m=1$ and see that $\phi(0)=0$ because $\mu^*_{n,1}=1$. This implies that $\phi=\phi_1$. The first few terms in the Taylor expansion of $\phi_1(\alpha)$ read as follows.

$$\phi_1(z)=-\tfrac{3}{2}z+\tfrac{11}{12}z^2-\tfrac{283}{432}z^3+\tfrac{5759}{11520}z^4-\tfrac{57137}{144000}z^5+\tfrac{2353751}{7257600}z^6+\cdots, \tag{31}$$

which can then be checked with the explicit expressions of $\mu^*_{n,m}$ for small m (see Section 4.1).

Error analysis. To justify the form (28) (with $\phi = \phi_1$), we consider the difference

$$\Delta^*_{n,m} := \mu^*_{n,m} - H_m - \phi_1(\alpha),$$

which satisfies the recurrence

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\Delta^*_{n,m} - \Delta^*_{n,m-\ell}\right) = E_1(n,m),$$

where

$$E_1(n,m) := \frac{1}{n} - \sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(H_m - H_{m-\ell} + \phi_1(\alpha) - \phi_1\left(\alpha - \tfrac{\ell}{n}\right)\right).$$

By the asymptotic relation (16) with $r = 1$ and the definition of ϕ_1, we have

$$\frac{1}{n} = \sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\tfrac{\ell}{m} + \phi_1'(\alpha)\tfrac{\ell}{n}\right) + O(n^{-2}),$$

and thus

$$E_1(n,m) = -\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(H_m - H_{m-\ell} - \tfrac{\ell}{m} + \phi_1(\alpha) - \phi_1\left(\alpha - \tfrac{\ell}{n}\right) - \phi_1'(\alpha)\tfrac{\ell}{n}\right) + O(n^{-2})$$

By Lemma 4, we see that

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\phi_1(\alpha) - \phi_1\left(\alpha - \tfrac{\ell}{n}\right) - \phi'(\alpha)\tfrac{\ell}{n}\right) = O(n^{-2}),$$

uniformly for $1 \leqslant m \leqslant n$. On the other hand, we have the upper bounds

$$H_m - H_{m-\ell} - \tfrac{\ell}{m} = \begin{cases} O\left(\ell^2 m^{-2}\right), & \text{if } \ell = o(m), \\ O(H_m), & \text{for } 1 \leqslant \ell \leqslant m. \end{cases}$$

Note that the first estimate is only uniform for $1 \leqslant \ell = o(m)$. When ℓ is close to m, say $m - \ell = O(m^{1-\varepsilon})$, the left-hand side blows up with m but the right-hand side $O\left(\ell^2 m^{-2}\right)$ remains bounded. Thus we split the sum at $\lceil\sqrt{m}\rceil$ and then obtain ($H_m - H_{m-\ell} - \frac{\ell}{m} = 0$ when $\ell = 1$)

$$\begin{aligned} &\sum_{2\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(H_m - H_{m-\ell} - \tfrac{\ell}{m}\right) \\ &\qquad = O\left(m^{-2}\sum_{1\leqslant \ell\leqslant\lceil\sqrt{m}\rceil} \ell^2\lambda^*_{n,m,\ell} + H_m \sum_{\sqrt{m}+1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\right). \end{aligned} \tag{32}$$

Now, by (9),

$$\begin{aligned} m^{-2}\sum_{2\leqslant \ell\leqslant m} \ell^2\lambda^*_{n,m,\ell} &= O\left(m^{-2}\sum_{j\geqslant 0}\frac{(1-\alpha)^j}{j!}\sum_{j+2\leqslant \ell\leqslant m}\frac{(j+\ell)^2\alpha^\ell}{\ell!}\right) \\ &= O\left(m^{-2}\alpha^2\right) = O\left(n^{-2}\right), \end{aligned} \tag{33}$$

and

$$H_m \sum_{\sqrt{m}+1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} = O\left(H_m \sum_{j \geqslant 0} \frac{(1-\alpha)^j}{j!} \sum_{j+\sqrt{m}+1 \leqslant \ell \leqslant m} \frac{\alpha^\ell}{\ell!} \right)$$
$$= O\left(H_m \sum_{\ell \geqslant \sqrt{m}+1} \frac{\alpha^\ell}{\ell!} \right) = O\left(\frac{H_m \alpha^{\sqrt{m}+1}}{\Gamma(\sqrt{m}+2)} \right). \tag{34}$$

By Stirling's formula (30), the last O-term is of order

$$\frac{m^{\frac{1}{4}} H_m}{n} e^{-\sqrt{m}(\log n - \frac{1}{2}\log m - 1)} = O(n^{-2}),$$

for $m \geqslant 1$. Combining these estimates, we then obtain

$$E_1(n,m) = O(n^{-2}),$$

uniformly for $1 \leqslant m \leqslant n$. Thus $\Delta_{n,m} := n\, \Delta^*_{n,m}$ satisfies a recurrence of the form

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\Delta_{n,m} - \Delta_{n,m-\ell} \right) = O(n^{-1}) \qquad (1 \leqslant m \leqslant n),$$

with $\Delta_{n,0} = 0$. It follows, by applying Lemma 3, that $\Delta_{n,m} = O(H_m)$, and we conclude that, uniformly for $0 \leqslant m \leqslant n$,

$$\mu^*_{n,m} = H_m + \phi_1(\alpha) + O\left(n^{-1} H_m\right).$$

This proves the first two terms of the asymptotic approximation to $\mu^*_{n,m}$ in (25). The more refined expansion is obtained by refining the same calculations and justification, which we carry out the main steps subsequently.

Refined computations. We consider now the difference

$$\Delta^*_{n,m} := \mu^*_{n,m} - (H_m + \phi_1(\alpha)) - \frac{1}{n}\left(b_1 H_m + \phi_2(\alpha)\right),$$

and will determine the constant b_1 and the function $\phi_2(z)$ such that

$$\Delta^*_{n,m} = O(n^{-2} H_m), \tag{35}$$

uniformly for $1 \leqslant m \leqslant n$, which then proves Theorem 2. By (26), $\Delta^*_{n,m}$ satisfies, for $1 \leqslant m \leqslant n$, the recurrence

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\Delta^*_{n,m} - \Delta^*_{n,m-\ell} \right) = E_2(n,m), \tag{36}$$

where

$$E_2(n,m) := \frac{1}{n} - \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(H_m - H_{m-\ell} \right) - \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\phi_1\left(\frac{m}{n}\right) - \phi_1\left(\frac{m-\ell}{n}\right) \right)$$
$$- \frac{b_1}{n} \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(H_m - H_{m-\ell} \right) - \frac{1}{n} \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\phi_2\left(\frac{m}{n}\right) - \phi_2\left(\frac{m-\ell}{n}\right) \right).$$

In particular, $\Delta^*_{n,0} = -\frac{\phi_2(0)}{n}$.

The hard part here is to derive an asymptotic expansion for $E_2(n,m)$ that holds uniformly for $1 \leqslant m \leqslant n$ as $n \to \infty$. To that purpose, we first extend Lemma 4 by using a Taylor expansion of third order for a $C^\infty[0,1]$-function $\phi(z)$, which then gives, uniformly for $1 \leqslant m \leqslant n$,

$$\begin{aligned}
&\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\phi\Big(\frac{m}{n}\Big) - \phi\Big(\frac{m-\ell}{n}\Big)\right)\\
&\quad= \frac{\phi'(\alpha)}{n}\sum_{1\leqslant \ell\leqslant m} \ell\lambda^*_{n,m,\ell} - \frac{\phi''(\alpha)}{2n^2}\sum_{1\leqslant \ell\leqslant m} \ell^2\lambda^*_{n,m,\ell} + O(n^{-3})\\
&\quad= \frac{\phi'(\alpha)}{n} S_1(\alpha) + \frac{1}{2n^2}\big(2\phi'(\alpha)U_1(\alpha) - \phi''(\alpha)S_2(\alpha)\big) + O(n^{-3}),
\end{aligned} \tag{37}$$

where we used Corollary 1.

We now examine weighted sums involving the difference of the harmonic numbers. We start with the following identity whose proof is straightforward. For a given function $f(x)$, let ∇ denote the backward difference operator $\nabla f(x) = f(x) - f(x-1)$. Then for $0 \leqslant \ell \leqslant m$

$$f(m) + f(m-1) + \cdots + f(m-\ell+1) = \sum_{1\leqslant k\leqslant m} \binom{\ell}{k}(-1)^{k-1}\nabla^{k-1} f(m).$$

Note that the sum vanishes for $k > \ell$. Take $f(x) = \frac{1}{x}$. Then we obtain

$$H_m - H_{m-\ell} = \sum_{k=1}^{m} \frac{\ell(\ell-1)\cdots(\ell-k+1)}{km(m-1)\cdots(m-k+1)},$$

for $0 \leqslant \ell \leqslant m$. This relation implies that, for $0 \leqslant \ell \leqslant \frac{m}{2}$ and $m \geqslant 1$,

$$H_m - H_{m-\ell} = \frac{\ell}{m} + \frac{[\![m \geqslant 2]\!]\,\ell(\ell-1)}{2m(m-1)} + O\Big(\frac{\ell(\ell-1)(\ell-2)}{m^3}\Big).$$

By the same argument we used above for (32), we get the expansion

$$\begin{aligned}
&\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\,(H_m - H_{m-\ell})\\
&\quad= \sum_{1\leqslant \ell\leqslant m}\left(\frac{\ell}{m} + \frac{[\![m \geqslant 2]\!]\,\ell(\ell-1)}{2m(m-1)} + O\Big(\frac{\ell(\ell-1)(\ell-2)}{m^3}\Big)\right)\lambda^*_{n,m,\ell} + O(n^{-3})\\
&\quad= \frac{1}{m}\sum_{1\leqslant \ell\leqslant m} \ell\lambda^*_{n,m,\ell} + \frac{[\![m \geqslant 2]\!]}{2m(m-1)}\sum_{1\leqslant \ell\leqslant m} \ell(\ell-1)\lambda^*_{n,m,\ell} + O(n^{-3})\\
&\quad= \frac{S_1(\alpha)}{\alpha n} + \frac{1}{2n^2}\left(\frac{U_1(\alpha)}{\alpha} + \frac{S_2(\alpha) - S_1(\alpha)}{\alpha^2}\right) - \frac{[\![m = 1]\!]}{2n^2} + O(n^{-3}),
\end{aligned} \tag{38}$$

which holds uniformly for $1 \leqslant m \leqslant n$. Note that for $m = 1$ a correction term is needed; more correction terms have to be introduced in more refined expansions (see Section 4.4).

Combining the estimates (37) (with $\phi = \phi_1, \phi_2$) and (38), we see that

$$E_2(n,m) = \frac{J_1(\alpha)}{n} + \frac{J_2(\alpha)}{2n^2} + \frac{[\![m=1]\!]}{2n^2} + O(n^{-3}),$$

uniformly for $1 \leqslant m \leqslant n$, where

$$\begin{aligned} J_1(z) &= 1 - \frac{S_1(z)}{z} - \phi_1'(z)S_1(z), \\ J_2(z) &= -\frac{S_2(z)-S_1(z)}{z^2} - \frac{2b_1}{z}S_1(z) - \left(\frac{2}{z} + 2\phi_1'(z)\right)U_1(z) \\ &\quad + \phi_1''(z)S_2(z) - 2\phi_2'(z)S_1(z). \end{aligned}$$

Obviously, $J_1(z) = 0$ because $\phi_1'(z) = \frac{1}{S_1(z)} - \frac{1}{z}$. To determine b_1 and ϕ_2, we observe that

$$\lim_{z\to 0} J_2(z) = \lim_{z\to 0}\left(-\frac{S_2(z)-S_1(z)}{z^2} - \frac{2b_1}{z}S_1(z) - \frac{2}{z}U_1(z)\right) = 2b_1 - 2,$$

where we used the relation $U_1(\alpha) = -S_0(\alpha) - \frac{1}{2}S_1(\alpha)$ (see (18)). In order that $E_2 = o(n^{-2})$ uniformly for $1 \leqslant m \leqslant n$, we need $2b_1 - 2 = 0$, so that $b_1 = 1$.

Now the equation $J_2(z) = 0$ also implies, by (18), that

$$\phi_2'(z) = -\frac{S_1'(z)S_2(z)}{2S_1^3(z)} + \frac{S_0(z)}{S_1(z)^2} + \frac{1}{2S_1(z)} + \frac{1}{2z^2} - \frac{1}{z}. \tag{39}$$

With these choices of b_1 and $\phi_2(z)$, we have

$$E_2(n,m) = \frac{[\![m=1]\!]}{2n^2} + O(n^{-3}),$$

uniformly for $1 \leqslant m \leqslant n$.

The exact solution to the differential equation (39) requires the constant term $\phi_2(0)$, which we have not yet specified. To specify this value, we take $m = 1$ in (35) and then obtain, by the recurrence (36),

$$\Delta_{n,1}^* = \Delta_{n,0}^* + \frac{E_2(n,1)}{\bar{\Lambda}_{n,1}} = -\frac{\phi_2(0)}{n} + nE_2(n,1) = -\frac{\phi_2(0)}{n} + \frac{1}{2n} + O(n^{-2}).$$

This entails the choice $\phi_2(0) = \frac{1}{2}$ in order that $\Delta_{n,1}^* = O(n^{-2})$. Thus we obtain the integral solution (24) for $\phi_2(z)$. In particular, the first few terms of $\phi_2(z)$ in the Taylor expansion are given as follows.

$$\phi_2(z) = \tfrac{1}{2} - \tfrac{7}{4}z + \tfrac{23}{18}z^2 - \tfrac{19951}{17280}z^3 + \tfrac{64903}{57600}z^4 - \tfrac{13803863}{12096000}z^5 + \cdots.$$

As a function in the complex plane, the region where $\phi_2(z)$ is analytic is dictated by the first zeros of $S_1(z)$, which exceeds unity.

To complete the proof of (35), we require a variation of Lemma 3, since the assumption on $a_{n,0}$ given there is not satisfied here.

Lemma 5. *Consider the recurrence*

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\,(a_{n,m}-a_{n,m-\ell}) = b_{n,m} \qquad (m\geqslant 1),$$

where $b_{n,m}$ is defined for $1\leqslant m\leqslant n$ and $n\geqslant 1$. Assume that $|a_{n,0}|\leqslant cn$ for $n\geqslant 1$, and $|a_{n,1}|\leqslant 2c$ for $n\geqslant 1$. If there exists a $c>0$ such that $|b_{n,m}|\leqslant \frac{c}{n}$ holds uniformly for $2\leqslant m\leqslant n$ and $n\geqslant 1$, then

$$|a_{n,m}|\leqslant 2cH_m \qquad (1\leqslant m\leqslant n). \tag{40}$$

Proof. The inequality (40) holds when $m=1$ by assumption. For $m\geqslant 2$, we write the recurrence as follows

$$a_{n,m} = \frac{1}{\bar\Lambda_{n,m}}\sum_{1\leqslant \ell<m}\lambda^*_{n,m,\ell}\,a_{n,m-\ell} + \frac{\lambda^*_{n,m,m}a_{n,0}}{\bar\Lambda_{n,m}} + \frac{b_{n,m}}{\bar\Lambda_{n,m}}.$$

By induction hypothesis and the two inequalities (see Lemma 3)

$$\bar\Lambda_{n,m}\geqslant \frac{m}{n}, \quad \text{and} \quad \lambda^*_{n,m,m} = n^{-m}\leqslant n^{-2},$$

we obtain

$$|a_{n,m}|\leqslant 2cH_{m-1} + \frac{n}{m}\cdot\frac{1}{n^2}\cdot cn + \frac{n}{m}\cdot\frac{c}{n}\leqslant 2cH_m,$$

and this proves the lemma. ∎

In view of the estimates $\Delta^*_{n,0}=O(n^{-1})$, $\Delta^*_{n,1}=O(n^{-2})$ and $E_2(n,m)=O(n^{-3})$, for $2\leqslant m\leqslant n$, there exists a constant $c>0$ such that the quantity $\Delta_{n,m}:=n^2\Delta^*_{n,m}$ satisfies the assumptions of Lemma 5, which implies the bound $\Delta_{n,m}=O(H_m)$, or, equivalently $\Delta^*_{n,m}=O(n^{-2}H_m)$, uniformly for $1\leqslant m\leqslant n$. This completes the proof of Theorem 2. ∎

4.4 An asymptotic expansion for the mean

The above procedure can be extended to get more smaller-order terms, but the expressions for the coefficients soon become very involved. However, it follows from the discussions in § 4.1 that we expect the asymptotic expansion

$$\mu^*_{n,m}\sim\sum_{k\geqslant 0}\frac{b_kH_m+\phi_{k+1}(\alpha)}{n^k}, \tag{41}$$

in the sense that the truncated asymptotic expansion

$$\mu^*_{n,m}=\sum_{0\leqslant k\leqslant K}\frac{b_kH_m+\phi_{k+1}(\alpha)}{n^k}+O\left(n^{-K-1}H_m\right) \tag{42}$$

holds uniformly for $K\leqslant m\leqslant n$ and introduces an error of order $n^{-K-1}H_m$. This asymptotic approximation may not hold when $1\leqslant m<K$ because additional correction terms are needed in that case. Technically, the correction terms stem from asymptotic expansions for sums of the form $\sum_{1\leqslant\ell\leqslant m}\lambda^*_{n,m}(H_m-H_{m-\ell})$; see (38) and the comments given there.

We propose here an easily codable procedure for the coefficients in the expansion, whose justification follows the same error analysis as above. We start with the formal expansion (41) and expand in all terms for large $m = \alpha n$ in decreasing powers of n, match the coefficients of n^{-K-1} on both sides for each $K \geqslant 0$, and then adjust the initial condition $\phi_{K+1}(0)$ by taking into account the extremal case when $m = K$ (for $m < K$ the expansion up to that order may not hold). With this algorithmic approach it is possible to determine the coefficients b_K and the functions $\phi_{K+1}(z)$ successively one after another.

Observe first that

$$H_m - H_{m-\ell} = \sum_{0\leqslant j<\ell} \frac{1}{m-j} = \sum_{r\geqslant 1} m^{-r}\beta_r(\ell) = \sum_{r\geqslant 1} n^{-r}\alpha^{-r}\beta_r(\ell),$$

where ($0^0 = 1$)

$$\beta_r(\ell) := \sum_{0\leqslant j<\ell} j^{r-1} = \frac{1}{r}\sum_{0\leqslant j<r}\binom{r}{j}B_j\ell^{r-j},$$

the B_j representing the Bernoulli numbers. On the other hand,

$$\phi_{k+1}(\alpha) - \phi_{k+1}\left(\alpha - \frac{\ell}{n}\right) = -\sum_{r\geqslant 1}\frac{\phi_{k+1}^{(r)}(\alpha)}{r!}\left(-\frac{\ell}{n}\right)^r.$$

Thus

$$\mu^*_{n,m} - \mu^*_{n,m-\ell} \sim \sum_{r\geqslant 1} n^{-r}\sum_{1\leqslant j\leqslant r}\left(\frac{\beta_j(\ell)b_{r-j}}{\alpha^j(r-j)!} - \frac{(-\ell)^j}{j!}\phi^{(j)}_{r-j+1}(\alpha)\right).$$

Then, by (13),

$$\sum_{1\leqslant\ell\leqslant m}\lambda^*_{n,m,\ell}\left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right) \sim \sum_{r\geqslant 1} n^{-r}[t^{-1}]\left(1+\frac{1}{nt}\right)^m\left(1+\frac{t}{n}\right)^{n+1-m} f_r(t),$$

where

$$f_r(t) := \sum_{\ell\geqslant 1} t^{\ell-1}\sum_{1\leqslant j\leqslant r}\left(\frac{\beta_j(\ell)b_{r-j}}{\alpha^j(r-j)!} - \frac{(-\ell)^j}{j!}\phi^{(j)}_{r-j+1}(\alpha)\right).$$

Now

$$\left(1+\frac{1}{nt}\right)^m\left(1+\frac{t}{n}\right)^{n+1-m} = \exp\left(\sum_{j\geqslant 1}\frac{(-1)^{j-1}}{j}\left(\frac{\alpha t^{-j} + (1-\alpha)t^j}{n^{j-1}} + \frac{t^j}{n^j}\right)\right).$$

A direct expansion using Bell polynomials $B^*_k(t_1,\ldots,t_k)$ (see [6]) then gives

$$\begin{aligned}\left(1+\frac{1}{nt}\right)^m\left(1+\frac{t}{n}\right)^{n+1-m} &= e^{\frac{\alpha}{t}+(1-\alpha)t}\sum_{k\geqslant 0}\frac{B^*_k(t_1,\ldots,t_k)}{k!}n^{-k}\\ &= e^{\frac{\alpha}{t}+(1-\alpha)t}\sum_{k\geqslant 0}\frac{\tilde{B}_k(\mathbf{t})}{k!t^{2k}}n^{-k}\end{aligned}$$

where $\tilde{B}_0 = 1$,

$$t_j := \frac{(-1)^j j!}{j+1}\left(\frac{\alpha}{t^{j+1}} + (1-\alpha)t^{j+1}\right) + (-1)^{j-1}(j-1)!t^j \qquad (j = 1, 2, \dots),$$

and $\tilde{B}_k(\mathbf{t})$ is a polynomial of degree $4k$.

Collecting these expansions, we get

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right) \sim \sum_{K\geqslant 0} n^{-(K+1)} \sum_{0\leqslant r\leqslant K} [t^{-1}]e^{\frac{\alpha}{t}+(1-\alpha)t}\frac{\tilde{B}_r(\mathbf{t})}{r!} f_{K+1-r}(t).$$

All terms now have the form

$$\begin{aligned}[t^{2r-1}]e^{\frac{\alpha}{t}+(1-\alpha)t}F(t) &= \sum_{\ell\geqslant 0}\frac{\alpha^\ell}{\ell!}[t^{\ell+2r-1}]e^{(1-\alpha)t}F(t)\\ &= \sum_{\ell\geqslant 0}\frac{\alpha^\ell}{\ell!}\sum_{0\leqslant j<2r+\ell}\frac{(1-\alpha)^j}{j!}\cdot\frac{F^{(\ell+2r-1-j)}(0)}{(\ell+2r-1-j)!}.\end{aligned}$$

Since we are solving the recurrence

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right) = \frac{1}{n},$$

we have the relations

$$\begin{cases}\left(\dfrac{b_0}{\alpha} + \phi_1'(\alpha)\right)[t^{-1}]\dfrac{e^{\frac{\alpha}{t}+(1-\alpha)t}}{(1-t)^2} = 1,\\ \displaystyle\sum_{0\leqslant r\leqslant K}[t^{2r-1}]e^{\frac{\alpha}{t}+(1-\alpha)t}\frac{\tilde{B}_r(\mathbf{t})}{r!}f_{K+1-r}(t) = 0, \quad (K\geqslant 1).\end{cases}$$

By induction, each ϕ_{K+1} satisfies a differential equation of the form

$$\left(\frac{b_K}{\alpha} + \phi'_{K+1}(\alpha)\right)S_K(\alpha) = \Psi_{K+1}[\phi_1, \dots, \phi_K](\alpha),$$

for some functional Ψ_{K+1}. Since $S_1(\alpha) \sim \alpha$ as $\alpha \to 0$, we also have the relation

$$b_K = \Psi_{K+1}[\phi_1, \dots, \phi_K](0).$$

Once the value of b_K is determined, we can then write

$$\phi_{K+1}(\alpha) = \phi_{K+1}(0) + \int_0^\alpha \left(\frac{\Psi_{K+1}[\phi_1, \dots, \phi_K](x)}{S_1(x)} - \frac{b_K}{x}\right)\mathrm{d}x,$$

and it remains to determine the initial value $\phi_{K+1}(0)$, which is far from being obvious. The crucial property we need is that the truncated expansion (42) holds when $K \leqslant m \leqslant n$, and particularly when $m = K$. So we compute (42) with $m = K$ and drop all terms of order smaller

than or equal to n^{-K-1}. Then we match the coefficient of n^{-K} with that in the expansion of $\mu^*_{n,K}$ obtained by a direct calculation from the recurrence (26).

We illustrate this procedure by computing the first two terms in (41). First, we have

$$\left(\frac{b_0}{\alpha} + \phi_1'(\alpha)\right) S_1(\alpha) = 1,$$

which implies $b_0 = 1$ and $\phi_1'(\alpha) = \frac{1}{S_1(\alpha)} - \frac{1}{\alpha}$. Moreover, substituting the initial value $m = K = 0$, we get

$$0 = \mu^*_{n,0} = H_0 + \phi_1(0) + O(n^{-1}) = \phi_1(0) + O(n^{-1}),$$

entailing $\phi_1(0) = 0$, which is consistent with what we obtained above.

The next-order term when $K = 1$ is (after substituting the relations $b_0 = 1$, $\phi_1'(\alpha) = \frac{1}{S_1(\alpha)} - \frac{1}{\alpha}$ and $\phi_1''(\alpha) = \frac{1}{\alpha^2} - \frac{S_1'(\alpha)}{S_1(\alpha)^2}$)

$$\begin{aligned}\left(\frac{b_1}{\alpha} + \phi_2'(\alpha)\right) &[t^{-1}]\frac{e^{\frac{\alpha}{t}+(1-\alpha)t}}{(1-t)^2} \\ &= [t^{-1}]e^{\frac{\alpha}{t}+(1-\alpha)t}\left(\frac{1}{2\alpha^2(1-t)^2} - \frac{(1+t)S_1'(\alpha)}{2(1-t)^3 S_1(\alpha)} + \frac{\alpha - 2t^3 + (1-\alpha)t^4}{2t^2(1-t)^2 S_1(\alpha)}\right),\end{aligned}$$

implying that

$$\left(\frac{b_1}{\alpha} + \phi_2'(\alpha)\right) S_1(\alpha) = -\frac{S_1'(\alpha)S_2(\alpha)}{2S_1(\alpha)^2} + \frac{S_0(\alpha)}{S_1(\alpha)} + \frac{1}{2} + \frac{S_1(\alpha)}{2\alpha^2}. \tag{43}$$

As $\alpha \to 0$, the right-hand side of (43) has the local expansion $1 - \frac{1}{4}\alpha + \cdots$, forcing $b_1 = 1$, and, accordingly, we obtain the same differential equation (39). Substituting the value $m = K = 1$ in (42) yields

$$\begin{aligned}1 = \mu^*_{n,1} &= H_1 + \phi_1\left(\frac{1}{n}\right) + \frac{1}{n}\left(H_1 + \phi_2\left(\frac{1}{n}\right)\right) + O(n^{-2}) \\ &= 1 + \frac{\phi_1'(0)}{n} + \frac{1}{n} + \frac{\phi_2(0)}{n} + O(n^{-2}),\end{aligned}$$

implying, by using (31), $\phi_2(0) = -\phi_1'(0) - 1 = \frac{1}{2}$, which is consistent with Theorem 2.

Although the expressions become rather involved for higher-order terms, all calculations (symbolic or numerical) are easily coded. For example, we have

$$\phi_3(z) = \tfrac{1}{12} - \tfrac{575}{432}z + \tfrac{15101}{11520}z^2 - \tfrac{8827}{5400}z^3 + \tfrac{2229089}{1036800}z^4 - \tfrac{361022171}{127008000}z^5 + \cdots.$$

4.5 Proof of Theorem 3

We now give an outline of the proof of Theorem 3 concerning the asymptotics of $\mathbb{E}(X_n)$. The method of proof relies on standard normal approximation to the binomial distribution.

We begin with

$$\mathbb{E}\left(t^{X_n}\right) = \sum_{0 \leqslant m \leqslant n} \pi_{n,m} P_{n,m}(t),$$

where $\pi_{n,m} := \binom{n}{m}\bar{\rho}^{n-m}\rho^m$ ($\bar{\rho} := 1-\rho$). From this expression, we see that

$$\mathbb{E}(X_n) = \sum_{0\leqslant m\leqslant n} \pi_{n,m}\,\mu_{n,m} = \frac{n-1}{\left(1-\frac{1}{n}\right)^n} \sum_{0\leqslant m\leqslant n} \pi_{n,m}\mu^*_{n-1,m}.$$

Write $m = \rho n + x\sqrt{\rho\bar{\rho}n}$. By Stirling's formula (30), we have

$$\pi_{n,m} = \frac{e^{-x^2/2}}{\sqrt{2\pi\rho\bar{\rho}n}}\left(1 + \frac{p_1(x)}{\sqrt{\rho\bar{\rho}n}} + \frac{p_2(x)}{\rho\bar{\rho}n} + \frac{p_3(x)}{(\rho\bar{\rho}n)^{3/2}} + O\left(\frac{1+x^{12}}{n^2}\right)\right),$$

uniformly for $x = o(n^{\frac{1}{6}})$, where, here and throughout the proof, the p_j are polynomials of x containing only powers of the same parity as j. On the other hand, by Theorem 2, we have in the same range of m

$$\mu^*_{n-1,m} = \log\rho n + \gamma + \phi_1(\rho) + \frac{p_5(x)}{\sqrt{\rho\bar{\rho}n}} + \frac{2\rho\bar{\rho}(\log\rho n + \gamma) + p_4(x)}{2\rho\bar{\rho}n} + \frac{p_7(x)}{(\rho\bar{\rho}n)^{3/2}} + O\left(\frac{\log n + x^4}{n^2}\right).$$

With these expansions, the asymptotic evaluation of $\mathbb{E}(X_n)$ is reduced to sums of the form

$$\frac{1}{\sqrt{2\pi\rho\bar{\rho}n}} \sum_{x=\frac{m-\rho n}{\sqrt{\rho\bar{\rho}n}}=o(n^{\frac{1}{6}})} x^r\, e^{-\frac{x^2}{2}} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} x^r e^{-\frac{x^2}{2}}\,\mathrm{d}x + O(n^{-L}),$$

for any $L > 1$ by an application of the Euler-Maclaurin formula. Thus polynomials of odd indices (containing only odd powers of x) will lead to asymptotically negligible terms after integration. Outside the range where $x = o(n^{\frac{1}{6}})$, the binomial distribution is smaller than any negative power of n, so the contribution from this range is also asymptotically negligible. Except for this part, all other steps are easily coded. ∎

5 Asymptotics of the variance

We prove in this section that the variance $\sigma^2_{n,m} := \mathbb{V}(X_{n,m}) = \mathbb{E}(X^2_{n,m}) - (\mathbb{E}(X_{n,m}))^2$ of $X_{n,m}$ is asymptotically quadratic.

Theorem 4. *For $1 \leqslant m \leqslant n$, the variance of $X_{n,m}$ satisfies*

$$\frac{\mathbb{V}(X_{n,m})}{en} = eH^{(2)}_m n - (2e+1)H_m + eH^{(2)}_m + e\psi_1(\alpha) - \phi_1(\alpha) - \frac{11e+1}{2n}H_m + \frac{5eH^{(2)}_m + 2e\psi_2(\alpha) - 2\phi_2(\alpha) + 2e\alpha\psi_1'(\alpha) - 2\alpha\phi_1'(\alpha) + \phi_1(\alpha)}{2n} + O\left(n^{-2}H_m\right),$$

where

$$\psi_1(\alpha) = \int_0^\alpha \left(\frac{S_2(x)}{S_1(x)^3} - \frac{1}{x^2} + \frac{2}{x}\right)\mathrm{d}x, \tag{44}$$

and

$$\psi_2(\alpha) = \frac{7}{12} - \int_0^\alpha \left(\frac{5S_1'(x)S_2(x)^2}{2S_1(x)^5} - \frac{2S_1'(x)S_3(x) + S_2(x)S_2'(x) + 6S_0(x)S_2(x)}{2S_1(x)^4} - \frac{S_0(x)}{S_1(x)^3} + \frac{2}{S_1(x)^2} - \frac{1}{x^3} + \frac{3}{x^2} - \frac{11}{2x} \right) \mathrm{d}x.$$

Similar to the mean, we work on the sequence $V_{n,m}^* := e_n^2(\sigma_{n+1,m}^2 + \mu_{n+1,m})/n^2$ and prove that (see Figure 3 and Appendix D)

$$V_{n,m}^* = H_m^{(2)} + \frac{-2H_m + \psi_1(\alpha) + 2H_m^{(2)}}{n} + \frac{-\frac{11}{2}H_m + \psi_2(\alpha) + \frac{7}{3}H_m^{(2)}}{n^2} + O(n^{-3}H_m) \qquad (2 \leqslant m \leqslant n). \tag{45}$$

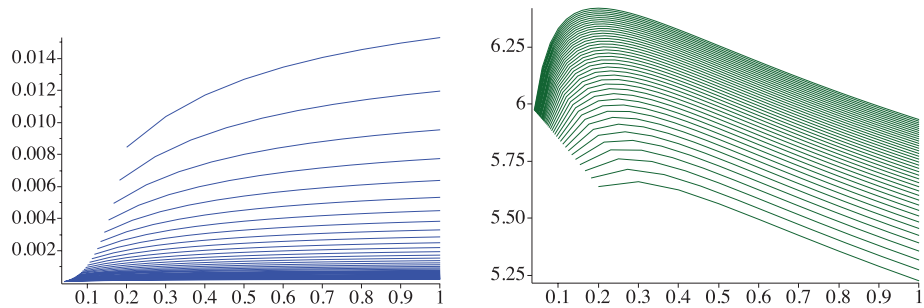


Figure 3: *The absolute differences* $|V_{n,m}^* -$ *RHS of* (45)$|$ *for* $2 \leqslant m \leqslant n$ *(normalized to the unit interval) and* $n = 10, \ldots, 50$ *(left in top-down order), and the absolute normalized differences* $n^3 H_m^{-1}|V_{n,m}^* -$ *RHS of* (45)$|$ *for* $n = 10, \ldots, 50$ *(right).*

The variance of X_n is computed by the relation

$$\mathbb{V}(X_n) = \sum_{0 \leqslant m \leqslant n} \pi_{n,m} \left(\sigma_{n,m}^2 + \mu_{n,m}^2\right) - \left(\sum_{0 \leqslant m \leqslant n} \pi_{n,m}\mu_{n,m} \right)^2,$$

where $\pi_{n,m} = \binom{n}{m}\rho^m(1-\rho)^{n-m}$, $\mu_{n,m} := \mathbb{E}(X_{n,m})$ and $\sigma_{n,m}^2 := \mathbb{V}(X_{n,m})$.

Theorem 5. *The variance of* X_n *satisfies asymptotically*

$$\frac{\mathbb{V}(X_n)}{en} = \frac{\pi^2}{6}en - (2e+1)(\log \rho n + \gamma) + v_1 - \frac{(11e+1)(\log \rho n + \gamma) - v_2}{2n} + O\left(n^{-2}\log n\right),$$

where ($\bar\rho := 1 - \rho$)

$$v_1 := e\left(\frac{\pi^2}{6} - 1\right) + e\psi_1(\rho) - \phi_1(\rho) + 2e\bar\rho\phi_1'(\rho) + e\bar\rho\rho\phi_1'(\rho)^2,$$

and

$$\begin{aligned} v_2 = {} & e\bar\rho^2\rho^2\phi_1''(\rho)^2 + 2e\bar\rho^2\rho(1+\rho\phi_1'(\rho))\phi_1'''(\rho) + 2e\rho\bar\rho(1+\phi_1'(\rho))\phi_1''(\rho) \\ & + 4e\bar\rho(1+\rho\phi_2'(\rho))\phi_1'(\rho) + 2e\bar\rho\rho\phi_1'(\rho)^2 + 4e\bar\rho\phi_2'(\rho) + e\bar\rho\rho\psi_1''(\rho) - \bar\rho\rho\phi_1''(\rho) \\ & + 2e\psi_2(\rho) - 2\phi_2(\rho) + 2e\rho\psi_1'(\rho) - 2\rho\phi_1'(\rho) + \phi_1(\rho) + \tfrac{5}{6}e\pi^2 - 3e - 1. \end{aligned}$$

Recurrences for the centered moment generating function. To compute the variance, one may start with the second moment and then consider the difference with the square of the mean; however, it is computationally more advantageous to study directly the recurrence satisfied by the variances themselves.

From (4), we have, by substituting $t = e^y$,

$$(1 - (1 - \Lambda_{n,m})\, e^y)\, P_{n,m}(e^y) = e^y \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}\, P_{n,m-\ell}(e^y),$$

which can be rewritten as

$$(e^{-y} - 1 - \Lambda_{n,m})\, P_{n,m}(e^y) = \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}\, P_{n,m-\ell}(e^y),$$

or

$$\sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} \big(P_{n,m}(e^y) - P_{n,m-\ell}(e^y) \big) = (1 - e^{-y}) P_{n,m}(e^y).$$

This is a simpler recurrence to start as fewer terms are involved for the moments.

We now consider the moment generating function for the centered random variables $X_{n,m} - \mu_{n,m}$

$$R_{n,m}(y) := P_n(e^y) e^{-\mu_{n,m} y},$$

which then satisfies the recurrence

$$\sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} \big(R_{n,m}(y) - R_{n,m-\ell}(y) e^{-(\mu_{n,m} - \mu_{n,m-\ell}) y} \big) = (1 - e^{-y}) R_{n,m}(y),$$

for $1 \leqslant m \leqslant n$ with $R_{n,0}(y) = 1$.

Variance. Let $\sigma_{n,m}^2 = \mathbb{V}(X_{n,m}) = R''_{n,m}(0)$ be the variance of $X_{n,m}$. Then $\sigma_{n,m}^2$ satisfies the recurrence

$$\sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} \left(\sigma_{n,m}^2 - \sigma_{n,m-\ell}^2 \right) = -1 + \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell} \left(\mu_{n,m} - \mu_{n,m-\ell} \right)^2.$$

In terms of $V_{n,m}^* := e_n^2 (\sigma_{n+1,m}^2 + \mu_{n+1,m})/n^2$, we have $V_{n,0}^* = 0$, and for $1 \leqslant m \leqslant n$

$$\sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}^* \left(V_{n,m}^* - V_{n,m-\ell}^* \right) = T_{n,m}^*, \tag{46}$$

where

$$T_{n,m}^* := \sum_{1 \leqslant \ell \leqslant m} \lambda_{n,m,\ell}^* \left(\mu_{n,m}^* - \mu_{n,m-\ell}^* \right)^2. \tag{47}$$

In particular, this gives

$$V_{n,1}^* = 1,$$
$$V_{n,2}^* = \frac{5n^4 + 8n^3 - n^2 - 4n + 1}{(2n^2 + 2n - 1)^2}.$$

The expressions become very lengthy as m increases. In Appendix C we give asymptotic expansions for $V_{n,m}^*$ for a few small m as $n \to \infty$. Based on these expansions, a suitable Ansatz for the asymptotic behavior of $V_{n,m}^*$ can be deduced (assisted again by computer algebra system), which then can be proven analogous to the method of proof presented in Section 4.

Proof of asymptotics with error analysis. By the same procedure used for $\mu^*_{n,m}$, we start from computing the asymptotic expansions for $V^*_{n,m}$ for small m. These expansions suggest the more uniform (for $1 \leqslant m \leqslant n$) asymptotic expansion

$$V^*_{n,m} \sim c_0 H^{(2)}_m + \frac{a_1 H_m + \psi_1(\alpha) + c_1 H^{(2)}_m}{n},$$

for some constants c_0, c_1 and a_1, and some function $\psi_1(z)$. Such an asymptotic form can be justified by the same approach we used above for $\mu^*_{n,m}$. More precisely, we now prove that

$$V^*_{n,m} = H^{(2)}_m + \frac{-2H_m + \psi_1(\alpha) + 2H^{(2)}_m}{n} + O(n^{-2} H_m), \tag{48}$$

uniformly for $0 \leqslant m \leqslant n$ and $n \geqslant 1$, where $\psi_1(z)$ is given in (44). Our proof start from considering the difference

$$\Delta^*_{n,m} := V^*_{n,m} - c_0 H^{(2)}_m - \frac{a_1 H_m + \psi_1(\alpha) + c_1 H^{(2)}_m}{n},$$

and specify the involved coefficients and $\psi_1(z)$ such that $\Delta^*_{n,m} = O(n^{-2} H_m)$. By (46), $\Delta^*_{n,m}$ satisfies, for $1 \leqslant m \leqslant n$, the recurrence

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\Delta^*_{n,m} - \Delta^*_{n,m-\ell}\right) = \tilde{E}_1(n,m), \tag{49}$$

with the initial value is $\Delta^*_{n,0} = -\frac{\psi_1(0)}{n}$, where ($T^*_{n,m}$ being defined in (47))

$$\tilde{E}_1(n,m) := T^*_{n,m} - \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \Bigg\{ (c_0 + \frac{c_1}{n}) \left(H^{(2)}_m - H^{(2)}_{m-\ell}\right) + \frac{a_1}{n} (H_m - H_{m-\ell}) + \frac{\psi_1\left(\frac{m}{n}\right) - \psi_1\left(\frac{m-\ell}{n}\right)}{n} \Bigg\}.$$

We will derive an asymptotic expansion for $\tilde{E}_1(n,m)$. For that purpose, we use the expansions (37), (38) as well as Theorem 2 in Section 4, and apply the same error analysis used for $\mu^*_{n,m}$. A careful analysis then leads to

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right)^2 = \frac{1}{mn} + \frac{1}{n^2}\left(-\frac{1}{\alpha} + \frac{(1+\alpha\phi_1'(\alpha))^2}{\alpha^2} S_2(\alpha)\right) + \frac{3}{2mn^2} + \frac{[\![m \geqslant 2]\!]}{2m(m-1)n^2} - \frac{[\![m=1]\!]}{n^2} + O(n^{-3}), \tag{50}$$

and

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell} \left(H^{(2)}_m - H^{(2)}_{m-\ell}\right) = \frac{1}{mn} + \frac{1}{n^2}\left(\frac{S_1(\alpha) - \alpha}{\alpha^2}\right) - \frac{1}{2mn^2} + \frac{[\![m \geqslant 2]\!]}{2m(m-1)n^2} - \frac{[\![m=1]\!]}{n^2} + O(n^{-3}), \tag{51}$$

both holding uniformly for $1 \leqslant m \leqslant n$ as $n \to \infty$.

Collecting the expansions (37), (38), (50) and (51), we obtain

$$\begin{aligned}\tilde{E}_1(n,m) = \frac{1-c_0}{mn} + \frac{1}{n^2}\Big\{ -\frac{1}{\alpha} + \frac{(1+\alpha\phi_1'(\alpha))^2}{\alpha^2} S_2(\alpha) - \frac{c_0}{\alpha^2}\left(S_1(\alpha)-\alpha\right) \\ - \left(\frac{a_1}{\alpha} + \psi_1'(\alpha)\right) S_1(\alpha)\Big\} + \frac{1}{mn^2}\left(\frac{3}{2} + \frac{c_0}{2} - c_1\right) \\ - \frac{[\![m=1]\!](1-c_0)}{n^2} + \frac{[\![m \geqslant 2]\!](1-c_0)}{2m(m-1)n^2} + O(n^{-3}),\end{aligned}$$

uniformly for $1 \leqslant m \leqslant n$.

We can now specify all the undetermined constants and $\psi_1(z)$ such that all terms except the last will vanish and $\tilde{E}_1(n,m) = O(n^{-3})$. This entails first the choices $c_0 = 1$ and $c_1 = 2$.

It remains only the $\frac{1}{n^2}$-term. We consider the limit when α tends to zero using the Taylor expansions (22), and deduce that $a_1 = -2$. These values give the equation satisfied by $\psi_1'(z)$

$$\psi_1'(z)S_1(z) = -\frac{S_1(z)}{z^2} + \frac{(1+z\phi_1'(z))^2}{z^2} S_2(z) + \frac{2S_1(z)}{z},$$

which in view of (3) leads to the differential equation

$$\psi_1'(z) = \frac{S_2(z)}{S_1^3(z)} - \frac{1}{z^2} + \frac{2}{z}. \tag{52}$$

Thus with the choices $c_0 = 1$, $a_1 = -2$, $c_1 = 2$, and the function $\psi_1'(z)$ by (52), we get the bound $\tilde{E}_1(n,m) = O(n^{-3})$ uniformly for $1 \leqslant m \leqslant n$. Accordingly, by (49), the sequences $\Delta_{n,m} := n^2 \Delta^*_{n,m}$ satisfy the recurrence

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell}\left(\Delta_{n,m} - \Delta_{n,m-\ell}\right) = O(n^{-1}) \qquad (1 \leqslant m \leqslant n),$$

with $\Delta_{n,0} = -\psi_1(0)n$. Choose now the initial value $\psi_1(0) = 0$, so that $\Delta_{n,0} = 0$ and Lemma 3 can be applied. This implies that $\Delta_{n,m} = O(H_m)$, and consequently $\Delta^*_{n,m} = O(n^{-2}H_m)$.

Also $\psi_1(z)$ is indeed given by (44). In particular, the first few terms in the Taylor expansion of $\psi_1(z)$ are given as follows.

$$\psi_1(z) = \tfrac{11}{4}z - \tfrac{49}{36}z^2 + \tfrac{2473}{4320}z^3 + \tfrac{1307}{14400}z^4 - \tfrac{12743687}{18144000}z^5 + \tfrac{194960323}{152409600}z^6 + \cdots.$$

This completes the proof of the asymptotic expansion (48) for $V^*_{n,m}$. The more refined approximation (45) follows the same line of proof but with more detailed expansions.

6 Limit laws when $m \to \infty$

We show in this section that the distribution of $X_{n,m}$, when properly normalized, tends to a Gumbel (or extreme-value or double exponential) distribution, as $m \to \infty$, $m \leqslant n$. The proof consists in showing that the result (6) when $m = O(1)$ extends to all $m \leqslant n$ but requires an additional correction term ϕ_1 coming from the linear part of the random variables, which complicates significantly the proof.

The standard Gumbel distribution $\mathscr{G}(1)$ (with mode zero, mean γ) is characterized by the distribution function $e^{-e^{-x}}$ and the moment generating function $\Gamma(1-s)$, respectively. Note that if $X \sim \mathrm{Exp}(1)$, then $-\log X \sim \mathscr{G}(1)$, which was the description used in [16].

The genesis of the Gumbel distribution is easily seen as follows.

Lemma 6. *Let $\eta_m := \sum_{1\leqslant r\leqslant m} \mathrm{Exp}(r)$, where the m exponential random variables are independent. Then $\eta_m - \log m$ converges in distribution to the Gumbel distribution*

$$\mathbb{P}\left(\eta_m - \log m \leqslant x\right) \to e^{-e^{-x}} \qquad (x > 0; m \to \infty).$$

Proof. We have

$$\mathbb{E}\left(e^{(\eta_m - H_m)s}\right) = \prod_{1\leqslant r\leqslant m} \frac{e^{-\frac{s}{r}}}{1-\frac{s}{r}} \to \prod_{r\geqslant 1} \frac{e^{-\frac{s}{r}}}{1-\frac{s}{r}} = e^{-\gamma s}\Gamma(1-s),$$

uniformly for $|s| \leqslant 1-\varepsilon$. Here we used the infinite-product representation of the Gamma function

$$\Gamma(1+s) = e^{-\gamma s} \prod_{r\geqslant 1} \frac{e^{\frac{s}{r}}}{1+\frac{s}{r}} \qquad (s \in \mathbb{C} \setminus \mathbb{Z}^-).$$

The lemma then follows from the asymptotic estimate

$$H_m = \log m + \gamma + O(m^{-1}) \qquad (m \to \infty),$$

and Curtiss's theorem (see [19, §5.2.3]). ∎

Unlike the case when $m = O(1)$, we need to subtract more terms to have the limit distribution.

Proposition 1. *For $1 \leqslant m \leqslant n$, we have the uniform asymptotic approximation*

$$\mathbb{E}\left(e^{\frac{X_{n,m}}{en} s - (H_m + \phi_1(\frac{m}{n}))s}\right) = \left(1 + O\left(\frac{H_m}{n}\right)\right) \prod_{1\leqslant r\leqslant m} \frac{e^{-\frac{s}{r}}}{1-\frac{s}{r}},$$

for $|s| \leqslant 1-\varepsilon$, where ϕ_1 is defined in (3).

Note that $\phi_1(x) = O(x)$ as $x \to 0$, and thus $\phi_1(\frac{m}{n}) = o(1)$ when $m = O(1)$. In this case, the proposition re-proves Theorem 1 (with an explicit error term).

A combination of Lemma 6 and Proposition 1 leads to the limit law for $X_{n,m}$ in the remaining range.

Theorem 6. *If $m \to \infty$ with n and $m \leqslant n$, then*

$$\mathbb{P}\left(\frac{X_{n,m}}{en} - \log m - \phi_1(\tfrac{m}{n}) \leqslant x\right) \to e^{-e^{-x}} \qquad (x > 0),$$

where ϕ_1 is defined in (3).

Theorem 7. *The number X_n of steps used by the $(1+1)$-EA to reach the final state $f(\mathbf{x}) = n$, when starting from the initial state $f(\mathbf{x}) \sim \mathrm{Binom}(n; 1-\rho)$, satisfies*

$$\mathbb{P}\left(\frac{X_n}{en} - \log \rho n - \phi_1(\rho) \leqslant x\right) \to e^{-e^{-x}} \qquad (x > 0).$$

From Figure 4, we see the fast convergence of the distribution to the limit law.

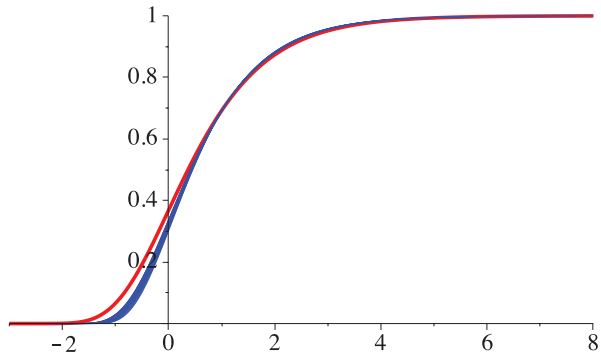


Figure 4: *Distributions of* $\frac{X_n}{en} - \log n - \log 2 - \phi_1(\frac{1}{2})$ *for* $n = 15, \dots, 35$*, and the limiting Gumbel curve.*

Outline of proofs. We focus on the proof of Proposition 1 for which we introduce the following normalized function

$$F_{n,m}(s) := \frac{\mathbb{E}\left(e^{\frac{X_{n,m}}{en}s}\right)e^{-\phi(\frac{m}{n})s}}{\prod\limits_{1\leqslant r\leqslant m}\frac{1}{1-\frac{s}{r}}} = \frac{P_{n,m}\left(e^{\frac{s}{en}}\right)e^{-H_m s-\phi(\frac{m}{n})s}}{\prod\limits_{1\leqslant r\leqslant m}\frac{e^{-\frac{s}{r}}}{1-\frac{s}{r}}}.$$

Here the probability generating function $P_{n,m}(t) := \mathbb{E}\left(t^{X_{n,m}}\right)$ of $X_{n,m}$ satisfies the recurrence (4) and the function $\phi(x)$ is any $C^2[0,1]$-function with $\phi(0) = 0$ (because in the proof we will require a Taylor expansion of order two). It turns out that if we choose $\phi(x) = \phi_1(x)$, where ϕ_1 (see (3)) appears as the second-order term in the asymptotic expansion of the mean (see (25)), then

$$F_{n,m}(s) \sim 1,$$

uniformly for all $1 \leqslant m \leqslant n$, $n \to \infty$, and $|s| \leqslant 1-\varepsilon$, where $\varepsilon > 0$ is independent of m, n. Indeed, our induction proof here does not rely on any information of the mean asymptotics and entails particularly the right choice of $\phi(x)$. This is why we specify ϕ only at a later stage.

The recurrence satisfied by $F_{n,m}$. By (4), $F_{n,m}(s)$ satisfies the following recurrence

$$F_{n,m}(s) = \frac{e^{\frac{s}{en}}\sum\limits_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}F_{n,m-\ell}(s)e^{-\left(\phi(\frac{m}{n})-\phi(\frac{m-\ell}{n})\right)s}\prod\limits_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)}{1-\left(1-\Lambda_{n,m}\right)e^{\frac{s}{en}}},$$

for $1 \leqslant m \leqslant n$, with $F_{n,0}(s) = 1$, where $\Lambda_{n,m} := \sum_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}$.

An auxiliary sum. Since we expect $F_{n,m}(s)$ to be close to 1, we replace all occurrences of F on the right-hand side by 1 and consider the following function

$$G_{n,m}(s) := \frac{e^{\frac{s}{en}}\sum\limits_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}e^{-\left(\phi(\frac{m}{n})-\phi(\frac{m-\ell}{n})\right)s}\prod\limits_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)}{1-\left(1-\Lambda_{n,m}\right)e^{\frac{s}{en}}}.$$

The following lemma is the crucial step in our proof.

Lemma 7. *Let $\phi(x)$ be a C^2-function on the unit interval satisfying $\phi(0) = 0$. Then*

$$G_{n,m}(s) = \frac{1 - \frac{s}{m}\left(1 + \alpha\phi'(\alpha)\right)\frac{S_1(\alpha)}{S(\alpha)} + O\left(\frac{1}{mn}\right)}{1 - \frac{s}{m}\cdot\frac{\alpha}{S(\alpha)} + O\left(\frac{1}{mn}\right)}, \tag{53}$$

where the O-terms hold uniformly for $1 \leqslant m \leqslant n$, and $|s| \leqslant 1 - \varepsilon$.

Proof. The proof consists in a detailed inspection of all factors, using estimates (20) and (21) we derived earlier for $\Lambda_{n,m}^{(r)}$. We consider first the case when $m = O(1)$. In this case, $S(\alpha), S_1(\alpha) = \alpha + O(\alpha^2)$ and the numerator and the denominator of (53) both have the form

$$1 - \frac{s}{m} + O(n^{-1}),$$

which can be readily checked by using the estimates (7) and

$$\Lambda_{n,m} = e^{-1}\alpha + O(\alpha^2).$$

From now on, we assume $m \geqslant m_0$, where m_0 is sufficiently large, say $m_0 \geqslant 10$. Throughout the proof, all O-terms hold uniformly for $|s| \leqslant 1 - \varepsilon$ and $m_0 \leqslant m \leqslant n$ and n large enough.

We begin with the denominator of $G_{n,m}(s)$, which satisfies

$$\begin{aligned} 1 - (1 - \Lambda_{n,m})e^{\frac{s}{en}} &= \Lambda_{n,m} - \frac{s}{en} + \Lambda_{n,m}\frac{s}{en} + O\left(n^{-2}\right) \\ &= \Lambda_{n,m}\left(1 + \frac{s}{en}\right)\left(1 - \frac{s}{en\Lambda_{n,m}} + O\left((mn)^{-1}\right)\right), \end{aligned}$$

where we used the estimate $\Lambda_{n,m} = \Omega(\alpha)$; see (21). By (20) and (21), the second-order term on the right-hand side satisfies

$$\frac{s}{en\Lambda_{n,m}} = \frac{s}{nS(\alpha)(1 + O(n^{-1}))} = \frac{s}{m}\cdot\frac{\frac{m}{n}}{S(\frac{m}{n})} + O\left((mn)^{-1}\right).$$

Thus we obtain

$$1 - (1 - \Lambda_{n,m})e^{\frac{s}{en}} = \Lambda_{n,m}\left(1 + \frac{s}{en}\right)\left(1 - \frac{s}{m}\cdot\frac{\alpha}{S(\alpha)} + O\left((mn)^{-1}\right)\right). \tag{54}$$

Now we turn to the numerator of $G_{n,m}(s)$ and look first at the exponential term

$$\begin{aligned} e^{-\left(\phi\left(\frac{m}{n}\right) - \phi\left(\frac{m-\ell}{n}\right)\right)s} &= e^{-\frac{\ell}{n}\phi'(\alpha)s + O\left(\frac{\ell^2}{n^2}\right)} \\ &= \left(1 - \frac{\ell}{n}\phi'(\alpha)s\right)\left(1 + O\left(\frac{\ell^2}{n^2}\right)\right), \end{aligned}$$

uniformly for $1 \leqslant \ell \leqslant m$, where we used the twice continuous differentiability of ϕ.

Consider now the finite product $\prod_{m-\ell+1 \leqslant r \leqslant m}\left(1 - \frac{s}{r}\right)$. Obviously, for $|s| \leqslant 1$, we have the uniform bound

$$\prod_{m-\ell+1 \leqslant r \leqslant m}\left|1 - \frac{s}{r}\right| \leqslant \prod_{m-\ell+1 \leqslant r \leqslant m}\left(1 + \frac{1}{r}\right) \leqslant e^{H_m} = O(m).$$

On the other hand, we also have the finer estimates

$$\begin{aligned}\prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right) &= e^{-(H_m-H_{m-\ell})s}\left(1+O\left(\frac{\ell^2}{m^2}\right)\right)\\ &= e^{-\frac{\ell}{m}s+O\left(\frac{\ell^2}{m^2}\right)}\left(1+O\left(\frac{\ell^2}{m^2}\right)\right)\\ &= \left(1-\frac{\ell}{m}s\right)\left(1+O\left(\frac{\ell^2}{m^2}\right)\right),\end{aligned}$$

uniformly for $1\leqslant \ell = o(m)$.

Combining these two estimates, we obtain the following approximation

$$\prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right) = \left(1-\frac{\ell}{m}s\right)\left(1+[\![\ell\geqslant 2]\!]O\left(\frac{\ell^2}{m^2}\right)+[\![\ell>\lceil\sqrt{m}\rceil]\!]O(m)\right),$$

which holds uniformly for $1\leqslant \ell\leqslant m$. Thus the numerator, up to the factor $e^{\frac{s}{en}}$, satisfies

$$\begin{aligned}&\sum_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}e^{-\left(\phi\left(\frac{m}{n}\right)-\phi\left(\frac{m-\ell}{n}\right)\right)s}\prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)\\ &\quad= \sum_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}-\frac{s}{m}\left(1+\alpha\phi'(\alpha)\right)\sum_{1\leqslant\ell\leqslant m}\ell\lambda_{n,m,\ell}\\ &\qquad+O\left(\frac{1}{m^2}\sum_{2\leqslant\ell\leqslant m}\ell^2\lambda_{n,m,\ell}+\frac{1}{mn}\sum_{1\leqslant\ell\leqslant m}\ell^2\lambda_{n,m,\ell}+m\sum_{\lceil\sqrt{m}\rceil+1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}\right).\end{aligned}$$

Each of the sums can be readily estimated as in (33) and (34), and we have

$$m^{-2}\sum_{2\leqslant\ell\leqslant m}\ell^2\lambda_{n,m,\ell} = O\left(n^{-2}\right).$$

Similarly,

$$(mn)^{-1}\sum_{1\leqslant\ell\leqslant m}\ell^2\lambda_{n,m,\ell} = O\left((mn)^{-1}\alpha\right) = O\left(n^{-2}\right).$$

Finally, for $m\geqslant 1$,

$$\begin{aligned}m\sum_{\lceil\sqrt{m}\rceil+1\leqslant\ell\leqslant m}\lambda_{n,m,\ell} &= O\left(\frac{m\alpha^{\sqrt{m}+1}}{\Gamma(\sqrt{m}+2)}\right)\\ &= O\left(m^{7/4}n^{-1}e^{-\sqrt{m}(\log n-\frac{1}{2}\log m-1)}\right) = O(n^{-2})\end{aligned}$$

Collecting these estimates, we get

$$\begin{aligned}
&\sum_{1\leqslant \ell\leqslant m} \lambda_{n,m,\ell} e^{-\left(\phi\left(\frac{m}{n}\right)-\phi\left(\frac{m-\ell}{n}\right)\right)s} \prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)\\
&\quad = \Lambda_{n,m} - \frac{s}{m}\left(1+\alpha\phi'(\alpha)\right)\Lambda_{n,m}^{(1)} + O\left(n^{-2}\right)\\
&\quad = \Lambda_{n,m}\left(1-\frac{s}{m}\left(1+\alpha\phi'(\alpha)\right)\frac{\Lambda_{n,m}^{(1)}}{\Lambda_{n,m}} + O\left((mn)^{-1}\right)\right)\\
&\quad = \Lambda_{n,m}\left(1-\frac{s}{m}\left(1+\alpha\phi'(\alpha)\right)\frac{S_1(\alpha)}{S(\alpha)} + O\left((mn)^{-1}\right)\right), \qquad (55)
\end{aligned}$$

by applying (20).

By (54), (55) and the simple estimate

$$e^{\frac{s}{en}} = \left(1+\tfrac{s}{en}\right)\left(1+O(n^{-2})\right),$$

we conclude (53). ∎

Corollary 3. *Let* $\phi(x) = \phi_1(x) = \int_0^x \left(\frac{1}{S_1(t)} - \frac{1}{t}\right) \mathrm{d}t$. *Then*

$$G_{n,m}(s) = 1 + O\big((mn)^{-1}\big),$$

where the O*-term holds uniformly for* $1 \leqslant m \leqslant n$, n *large enough and* $|s| \leqslant 1-\varepsilon$.

Proof. To obtain the error term $O((mn)^{-1})$, we choose ϕ in a way that the two middle terms in the fraction of (53) are identical, which means

$$\frac{x}{S(x)} = (1+x\phi'(x))\frac{S_1(x)}{S(x)}.$$

Observe that $S(x) > 0$ for $x > 0$. This, together with $\phi_1(0) = 0$, implies $\phi = \phi_1$, which is not only a C^2-function but also analytic in the unit circle. ∎

Proof of Proposition 1. We now prove Proposition 1 by induction.

Lemma 8. *Let* $\phi = \phi_1$. *Then*

$$F_{n,m}(s) = 1 + O\left(n^{-1}H_m\right),$$

uniformly for $0 \leqslant m \leqslant n$, n *large enough and* $|s| \leqslant \kappa$, $\kappa \in (0,1)$.

Proof. We use induction on m and show that there exists a constant $C > 0$, such that

$$|F_{n,m}(s) - 1| \leqslant Cn^{-1}H_m,$$

for all $1 \leqslant m \leqslant n$, $n \geqslant n_0$ large enough and $|s| \leqslant \kappa$.

When $m = 0$, the lemma holds, since $F_{n,0}(s) \equiv 1$.

Assume that the lemma holds for all functions $F_{n,k}(s)$ for $0 \leqslant k \leqslant m$ and $n \geqslant n_0$. By Corollary 3, there exists a constant $C_1 > 0$ such that for all $1 \leqslant m \leqslant n$, $n \geqslant n_1$ large enough and $|s| \leqslant \kappa_1$, $\kappa_1 > 0$,

$$|G_{n,m}(s) - 1| \leqslant C_1(mn)^{-1}.$$

Now

$$\begin{aligned}|F_{n,m}(s)-1| &= |F_{n,m}(s)-G_{n,m}(s)+G_{n,m}(s)-1| \\ &\leqslant |F_{n,m}(s)-G_{n,m}(s)|+C_1(mn)^{-1}.\end{aligned}$$

The first term on the right-hand side can be re-written as

$$\begin{aligned}&|F_{n,m}(s)-G_{n,m}(s)| \\ &\quad= \frac{\left|e^{\frac{s}{en}}\sum_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}(F_{n,m-\ell}(s)-1)e^{-\left(\phi\left(\frac{m}{n}\right)-\phi\left(\frac{m-\ell}{n}\right)\right)s}\prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)\right|}{\left|1-\left(1-\Lambda_{n,m}\right)e^{\frac{s}{en}}\right|}.\end{aligned}$$

Since we assume $|s|\leqslant 1$, the product involved in the sum on the right-hand side is nonnegative and we have, by the induction hypothesis,

$$\begin{aligned}|F_{n,m}(s)-G_{n,m}(s)| &\leqslant \frac{e^{\frac{s}{en}}\sum_{1\leqslant\ell\leqslant m}\lambda_{n,m,\ell}e^{-\left(\phi\left(\frac{m}{n}\right)-\phi\left(\frac{m-\ell}{n}\right)\right)s}\frac{CH_{m-\ell}}{n}\prod_{m-\ell+1\leqslant r\leqslant m}\left(1-\frac{s}{r}\right)}{1-\left(1-\Lambda_{n,m}\right)e^{\frac{s}{en}}} \\ &\leqslant \frac{CH_{m-1}}{n}G_{n,m}(s)\leqslant\frac{CH_{m-1}}{n}+\frac{CH_{m-1}}{n}\cdot|G_{n,m}(s)-1| \\ &\leqslant \frac{CH_{m-1}}{n}+\frac{CH_{m-1}}{n}\cdot\frac{C_1}{mn}.\end{aligned}$$

It follows that

$$|F_{n,m}(s)-1|\leqslant\frac{CH_m}{n}+\frac{1}{mn}\left(C_1-C+C_1C\frac{H_{m-1}}{n}\right).$$

Choose first $n_2\geqslant n_1$ such that $\frac{H_{n_2-1}}{n_2}\leqslant\frac{1}{2C_1}$, which implies that $C_1C\frac{H_{m-1}}{n}\leqslant\frac{C}{2}$ for $1\leqslant m\leqslant n$ and $n\geqslant n_2$. Then choose $C=2C_1$. We then have

$$C_1-C+\frac{C_1CH_{m-1}}{n}\leqslant C_1-\frac{C}{2}\leqslant 0,$$

and thus

$$|F_{n,m}(s)-1|\leqslant\frac{CH_m}{n}.$$

Note that apart from requiring $|s|\leqslant 1$ the only restriction on s comes from $G_{n,m}(s)$, thus we may choose $\kappa=\min(1,\kappa_1)$. This completes the proof. ∎

The Gumbel limit laws for $X_{n,m}$ ($m\to\infty$). We prove Theorem 6 by Proposition 1. Since $m\to\infty$, we have

$$\begin{aligned}\mathbb{E}\left(e^{\frac{X_{n,m}}{en}s-(\log m+\phi_1(\alpha))s}\right) &= P_{n,m}\left(e^{\frac{s}{en}}\right)e^{-H_ms+\gamma s-\phi_1\left(\frac{m}{n}\right)s}\left(1+O\left(m^{-1}\right)\right) \\ &= e^{\gamma s}F_{n,m}(s)\prod_{1\leqslant r\leqslant m}\frac{e^{-\frac{s}{r}}}{1-\frac{s}{r}}\left(1+O\left(m^{-1}\right)\right) \\ &= \Gamma(1-s)\left(1+O\left(\frac{\log m}{n}+\frac{1}{m}\right)\right).\end{aligned}\tag{56}$$

Thus Theorem 6 follows from another application of Curtiss's theorem (see [19, §5.2.3]).

The Gumbel limit law for X_n. We now prove Theorem 7, starting from the moment generating function ($\bar{\rho} := 1 - \rho$)

$$\mathbb{E}\left(e^{X_n s}\right) = \sum_{0 \leqslant m \leqslant n} \binom{n}{m} \rho^m \bar{\rho}^{n-m} P_{n,m}\left(e^s\right).$$

Then

$$\mathbb{E}\left(e^{\frac{X_n}{en} s - (\log \rho n + \phi_1(\rho))s}\right) = \sum_{0 \leqslant m \leqslant n} \binom{n}{m} \rho^m \bar{\rho}^{n-m} P_{n,m}\left(e^{\frac{s}{en}}\right) e^{-(H_m - \gamma + \phi_1(\alpha))s + \delta_{n,m} s},$$

where

$$\delta_{n,m} := H_m - \log \rho n - \gamma + \phi_1(\alpha) - \phi_1(\rho).$$

Since the binomial distribution is highly concentrated around the range $m = \rho n + x\sqrt{\rho\bar{\rho} n}$ where $x = o(n^{\frac{1}{6}})$, we see that

$$\delta_{n,m} = O\left(n^{-\frac{1}{2}}|x|\right),$$

for m in this central range. By a standard argument (Gaussian approximation of the binomial and exponential tail estimates) using the expansion (56), we then deduce that

$$\mathbb{E}\left(e^{\frac{X_n}{en} s - (\log \rho n + \phi_1(\rho))s}\right) = \Gamma(1-s)\left(1 + O\left(n^{-\frac{1}{2}}\right)\right).$$

This proves Theorem 7.

7 Analysis of the (1 + 1)-EA for LeadingOnes

We consider the complexity of the (1 + 1)-EA when the underlying fitness function is the number of leading ones. This problem has been examined repeatedly in the literature due to the simple structures it exhibits; see [4, 14, 26] and the references therein. The strongest results obtained were those by Ladret [26] (almost unknown in the EA literature) where she proved that the optimization time under LeadingOnes is asymptotically normally distributed with mean asymptotic to $\frac{e^c - 1}{2c^2} n^2$ and variance to $\frac{3(e^{2c}-1)}{8c^3} n^3$, where $p = \frac{c}{n}$, $c > 0$.

We re-visit this problem and obtain similar type of results by a completely different approach, which can be readily amended for obtaining the convergence rate.

Throughout this section, the probability p still carries the same meaning from Algorithm (1 + 1)-EA and $q = 1 - p$.

Lemma 9. *Let $Y_{n,m}$ denote the conditional optimization time when beginning with a random input (each bit being 1 with probability $\frac{1}{2}$) that has $n - m$ leading ones. Then the moment generating function $Q_{n,m}(s)$ of $Y_{n,m}$ satisfies the recurrence relation*

$$Q_{n,m}(s) = \frac{pq^{n-m}e^s}{1 - (1 - pq^{n-m})e^s}\left(2^{1-m} + \sum_{1 \leqslant \ell < m} \frac{Q_{n,\ell}(s)}{2^{m-\ell}}\right), \tag{57}$$

for $1 \leqslant m \leqslant n$, where $q = 1 - p$.

Proof. The probability of jumping from a state with $n-m$ leading ones to another state with $n-m+\ell$ leading ones is given by

$$(1-p)^{n-m} \cdot p \cdot 2^{-\ell} \qquad (1 \leqslant \ell < m),$$

which corresponds to the situation when the first $n-m$ bits do not toggle their values, the $(n-m+1)$st bit toggles (from 0 to 1), together with the following $\ell-1$ bits also being 1. When $\ell = m$, the probability becomes

$$(1-p)^{n-m} \cdot p \cdot 2^{-\ell+1}.$$

We thus obtain the recurrence relation

$$Q_{n,m}(s) = pq^{n-m}e^s \left(2^{1-m} + \sum_{1 \leqslant \ell < m} \frac{Q_{n,\ell}(s)}{2^{m-\ell}} \right) + (1 - pq^{n-m})e^s Q_{n,m}(s),$$

which implies (57). ∎

The most interesting case is when $p \asymp n^{-1}$ (roughly, pq^n is linear, giving rise to polynomial bounds for the cost), all other cases when $pn \to \infty$ lead to higher-order complexity.

Small m. We start with the simplest case when $m = 1$ and obtain, by (57),

$$Q_{n,1}(s) = \frac{pq^{n-1}e^s}{1-(1-pq^{n-1})e^s}. \tag{58}$$

Then the mean of $Y_{n,1}$ is simply given by

$$\mathbb{E}(Y_{n,1}) = \frac{1}{pq^{n-1}},$$

which, by substituting $p = \frac{c}{n}$, yields

$$\mathbb{E}(Y_{n,1}) = \frac{e^c}{c} n - \left(1 - \frac{c}{2}\right) e^c + O\left(\frac{e^c}{n}(1+c^3)\right).$$

Note that this estimate holds as long as $c = o(\sqrt{n})$. Similarly, the variance is given by

$$\mathbb{V}(Y_{n,1}) = \frac{1}{(pq^{n-1})^2} - \frac{1}{pq^{n-1}},$$

which satisfies, when $p = \frac{c}{n}$,

$$\mathbb{V}(Y_{n,1}) = \frac{e^{2c}}{c^2} n^2 - \frac{e^c + (2-c)e^{2c}}{c} n + \left(1 - \frac{c}{2}\right) e^c + \left(1 - \frac{4}{3}c + \frac{c^2}{2}\right) e^{2c} + O\left(\frac{e^{2c}}{n}(c^2 + c^4)\right),$$

uniformly when $c = o(\sqrt{n})$.

We then consider the normalized random variables $cY_{n,1}/(e^c n)$. By the expansion

$$\frac{\frac{c}{n}\left(1-\frac{c}{n}\right)^{n-m}\exp\left(\frac{ce^{-c}}{n}s\right)}{1-\left(1-\frac{c}{n}\left(1-\frac{c}{n}\right)^{n-m}\right)\exp\left(\frac{ce^{-c}}{n}s\right)}=\frac{1}{1-s}+O\left(\frac{c|s|(m+c)}{n|1-s|^2}\right),$$

uniformly when s is away from 1 and $m=o(n)$, we obtain

$$\mathbb{E}\left(e^{cY_{n,1}s/(e^c n)}\right)\to\frac{1}{1-s},$$

implying that the limit law is an exponential distribution with the density e^{-x}. While (58) shows that $Y_{n,1}\equiv X_{n,1}$ when $p=\frac{1}{n}$, they behave differently when $m\geqslant 2$.

Theorem 8. *For each $1\leqslant m=O(1)$, the limit distribution of $cY_{n,m}/(e^c n)$ is a binomial mixture of Gamma distributions; more precisely,*

$$\mathbb{P}\left(\frac{cY_{n,m}}{e^c n}\leqslant x\right)\to\frac{1}{2^{m-1}}\sum_{0\leqslant j<m}\binom{m-1}{j}\int_0^x e^{-t}\frac{t^j}{j!}\,\mathrm{d}t\qquad(x>0),\tag{59}$$

as $n\to\infty$. The mean and the variance satisfy

$$\mathbb{E}(Y_{n,m})\sim\frac{m+1}{2ce^{-c}}n,\qquad\mathbb{V}(Y_{n,m})\sim\frac{3m+1}{4c^2e^{-2c}}n^2.\tag{60}$$

Note that when $m=1$, (59) degenerates to the exponential distribution. On the other hand, the normalizing factor $ce^{-c}n$ is not asymptotically equivalent to the mean.

Proof. By induction and (57), we see that

$$\mathbb{E}\left(e^{cY_{n,m}s/(e^c n)}\right)\to\frac{(1-\frac{s}{2})^{m-1}}{(1-s)^m},$$

when $m=O(1)$. Since

$$\int_0^\infty e^{-x(1-s)}\sum_{0\leqslant j<m}\frac{1}{2^{m-1}}\binom{m-1}{j}\frac{x^j}{j!}\,\mathrm{d}x=\frac{(1-\frac{s}{2})^{m-1}}{(1-s)^m},$$

we then deduce (59). The mean and the variance then follows from straightforward calculations. ∎

Mean and the variance of $Y_{n,m}$: $1\leqslant m\leqslant n$. The recurrence (57) is much simpler than (4) and we can indeed obtain very precise expressions and approximations for the mean and the variance.

Theorem 9. *The mean $\nu_{n,m}$ and the variance $\varsigma_{n,m}$ of $Y_{n,m}$ are given explicitly as follows. For $1\leqslant m\leqslant n$*

$$\nu_{n,m}:=\mathbb{E}(Y_{n,m})=\frac{1}{pq^{n-1}}\left(\frac{1-q^{m-1}}{2p}+q^{m-1}\right),\tag{61}$$

and

$$\varsigma_{n,m}^2:=\mathbb{V}(Y_{n,m})=-\nu_{n,m}+\frac{3q^2-(4q^2-1)q^{2m}}{4p^3(1+q)q^{2n}}.\tag{62}$$

With these closed-form expressions, we easily obtain, assuming $p = \frac{c}{n}$, where $c > 0$,

$$\nu_{n,m} = \frac{e^c(m+1)}{2c} n - \frac{e^c}{4}\left(m^2 + (3-c)m - c\right) + O\left(\frac{ce^c}{n}\left(m^3 + mc^2\right)\right),$$

and

$$\varsigma_{n,m}^2 = \frac{e^{2c}(3m+1)}{4c^2} n^2 - \frac{e^{2c}}{8c}\left(3m^2 + (5-3c)m - c + 2(m+1)e^{-c}\right) n \\ + O\left(e^{2c}\left(m^3 + c^2\right)\right),$$

uniformly for $cm = o(n)$. We see that the asymptotic equivalents (60) indeed hold in the wider range $cm = o(n)$.

More uniform approximations have the following forms.

Corollary 4. *Assume that $p = \frac{c}{n}$, where $c = o(\sqrt{n})$. Then, uniformly for $0 \leqslant \alpha := \frac{m}{n} \leqslant 1$,*

$$\nu_{n,m} = \frac{e^c}{2c^2}\left(1 - e^{-c\alpha}\right) n^2 + \frac{e^c}{4c}\left(c - 2 + e^{-c\alpha}\left(4 - c + c\alpha\right)\right) n + O\left(c(c+1)e^c\right), \tag{63}$$

and

$$\varsigma_{n,m}^2 = \frac{3e^{2c}}{8c^3}\left(1 - e^{-2c\alpha}\right) n^3 + O\left(c^{-2}e^{2c}(1+c)n^2\right). \tag{64}$$

Proof of Theorem 9. Our approach is based on (57) and it turns out that all moments satisfy the same simple recurrence of the following type.

Lemma 10. *The solution to the recurrence relation*

$$a_m = b_m + \sum_{1 \leqslant \ell < m} \frac{a_\ell}{2^{m-\ell}} \qquad (m \geqslant 1),$$

is given by the closed-form expression

$$a_m = b_m + \frac{1}{2} \sum_{1 \leqslant j < m} b_j. \tag{65}$$

Proof. The corresponding generating functions $f(z) := \sum_{m \geqslant 1} a_m z^m$ and $g(z) := \sum_{m \geqslant 1} b_m z^m$ satisfy the equation

$$f(z) = g(z) + \frac{z}{2-z} f(z),$$

or

$$f(z) = \frac{1 - \frac{z}{2}}{1 - z} g(z).$$

This proves (65). ∎

From (57) (by taking derivative with respect to s and then substituting $s = 1$), we see that the mean $\nu_{n,m}$ satisfies the recurrence

$$\nu_{n,m} = \frac{1}{pq^{n-m}} + \sum_{1 \leqslant \ell < m} \frac{\nu_{n,\ell}}{2^{m-\ell}} \qquad (m \geqslant 1).$$

Substituting $b_m = 1/(pq^{n-m})$ into (65), we obtain (61).

Similarly, for the second moment $s_{n,m} := \mathbb{E}(Y_{n,m}^2)$, we have the recurrence

$$s_{n,m} = \frac{2\nu_{n,m} - 1}{pq^{n-m}} + \sum_{1 \leqslant \ell < m} \frac{s_{n,\ell}}{2^{m-\ell}}.$$

By the same procedure, we obtain

$$s_{n,m} = -\nu_{n,m} + \frac{q^2(2-q) - (q+1)q^{m+1}(2q-1) + (2q-1)(2q^2-1)q^{2m}}{2p^4(1+q)q^{2n}}, \tag{66}$$

implying (62). This proves Theorem 9. ∎

The proofs of the two Corollaries are straightforward and omitted.

A finite-product representation for $Q_{n,m}(s)$. The recurrence relation (57) can indeed be solved explicitly as follows.

Proposition 2. *The moment generating function $Q_{n,m}(s)$ of $Y_{n,m}$ has the closed-form*

$$Q_{n,m}(s) = \frac{1}{1 - \frac{1-e^{-s}}{pq^{n-m}}} \prod_{1 \leqslant j < m} \frac{1 - \frac{1-e^{-s}}{2pq^{n-j}}}{1 - \frac{1-e^{-s}}{pq^{n-j}}}, \tag{67}$$

for $m \geqslant 1$.

Proof. Let $\omega := (1 - e^{-s})/(pq^n)$. We start with the recurrence (from (57))

$$Q_{n,m}(s) = \omega q^m Q_{n,m}(s) + 2^{1-m} + \sum_{1 \leqslant \ell < m} \frac{Q_{n,\ell}(s)}{2^{m-\ell}},$$

which, by (65), has the alternative form

$$Q_{n,m}(s) = 1 + \omega q^m Q_{n,m}(s) + \frac{\omega}{2} \sum_{1 \leqslant h < m} q^h Q_{n,h}(s). \tag{68}$$

From (68), we see that the bivariate generating function

$$Q_n(z,s) := \sum_{m \geqslant 1} Q_{n,m}(s) z^m$$

of $Q_{n,m}(s)$ satisfies

$$Q_n(z,s) = \frac{z}{1-z} + \omega Q_n(qz,s) + \frac{\omega}{2} \cdot \frac{z}{1-z} Q_n(qz,s),$$

which implies the simpler functional equation

$$Q_n(z,s) = \frac{z}{1-z} + \omega \frac{1 - \frac{z}{2}}{1-z} Q_n(qz,s).$$

Multiplying both sides by $1-z$ gives

$$(1-z)Q_n(z,s) = z + \omega\left(1-\frac{z}{2}\right)Q_n(qz,s),$$

implying the relation

$$\frac{Q_{n,m}(s)}{Q_{n,m-1}(s)} = \frac{1-\frac{1}{2}\omega q^{m-1}}{1-\omega q^m} \qquad (m \geqslant 2).$$

Accordingly, we obtain the closed-form expression (67). ∎

Let

$$G_m(t) := \frac{pq^{n-m}t}{1-(1-pq^{n-m})t}$$

denote the probability generating function of a geometric distribution $\mathrm{Geo}(pq^{n-m})$ with parameter pq^{n-m} and support $\{1,2,\dots\}$.

Corollary 5. *The random variables $Y_{n,m}$ can be decomposed as the sum of m independent random variables*

$$Y_{n,m} \stackrel{d}{=} Z_{n,m}^{[0]} + \cdots + Z_{n,m}^{[m-1]}, \tag{69}$$

where $Z_{n,m}^{[0]} \sim Geo(pq^{n-m})$ and the $Z_{n,m}^{[j]}$ are mixture of $Geo(pq^{n-j})$

$$\mathbb{E}\left(t^{Z_{n,m}^{[j]}}\right) = \frac{1}{2}\cdot\frac{1-(1-2pq^{n-j})t}{1-(1-pq^{n-j})t} = \frac{1}{2} + \frac{R_j(t)}{2} \qquad (j=1,\dots,m-1).$$

Thus the mean of $Y_{n,m}$ is given by

$$\mathbb{E}(Y_{n,m}) = \sum_{0\leqslant j<m}\mathbb{E}\left(Z_{n,m}^{[j]}\right) = \frac{1}{pq^{n-m}} + \frac{1}{2}\sum_{1\leqslant j<m}\frac{1}{pq^{n-j}},$$

which is identical to (61). Similarly, the variance of $Y_{n,m}$ satisfies

$$\varsigma_{n,m}^2 = \sum_{0\leqslant j<m}\mathbb{V}\left(Z_{n,m}^{[j]}\right) = \frac{3-2pq^{n-m}}{8(pq^{n-m})^2} + \frac{1}{2}\sum_{1\leqslant j<m}\frac{1-pq^{n-j}}{(pq^{n-j})^2},$$

which is also identical to (62).

Theorem 10. *The distributions of $\frac{Y_{n,m}-\nu_{n,m}}{\varsigma_{n,m}}$ are asymptotically normal*

$$\mathbb{P}\left(\frac{Y_{n,m}-\nu_{n,m}}{\varsigma_{n,m}} \leqslant x\right) \to \Phi(x),$$

uniformly as $m\to\infty$ (with n) and $m\leqslant n$, where $\Phi(x) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,\mathrm{d}t$ denotes the standard normal distribution function.

Proof. Again from the decomposition (69), we derive the following expression for the third central moment

$$\kappa_{n,m} := \mathbb{E}\left(Y_{n,m} - \nu_{n,m}\right)^3 = \frac{7q^3 - (8q^3-1)q^{3m}}{4(1-q^3)(pq^n)^3} - 3\varsigma_{n,m}^2 - 2\nu_{n,m}.$$

Similarly, the fourth central moment satisfies

$$\begin{aligned}\mathbb{E}\left(Y_{n,m} - \nu_{n,m}\right)^4 - 3\varsigma_{n,m}^4 = &\frac{3(15q^4 - (16q^4-1)q^{4m})}{8(1-q^4)(pq^n)^4} \\ &- 6\mathbb{E}\left(Y_{n,m} - \nu_{n,m}\right)^3 - 11\varsigma_{n,m}^2 - 6\nu_{n,m},\end{aligned}$$

which implies that

$$\mathbb{E}\left(Y_{n,m} - \nu_{n,m}\right)^4 = 3\varsigma_{n,m}^4(1+o(1)),$$

uniformly for $1 \leqslant m \leqslant n$. We then deduce a central limit theorem by, say Lyapounov's condition, or by Levy's continuity theorem; see, for example, [30]. We can indeed derive an optimal Berry-Esseen bound by more refined Fourier argument, details being omitted here. ∎

In particular, we have

$$\mathbb{E}\left(Y_{n,m} - \nu_{n,m}\right)^4 - 3\varsigma_{n,m}^4 \sim \frac{45(1-e^{-4c\alpha})}{32c^5e^{-4c}}n^5, \tag{70}$$

when $m \to \infty$ and $m \leqslant n$. This will be needed later.

Random input. Now consider the cost Y_n used by Algorithm $(1+1)$-EA when starting from a random input (each bit being 1 with probability $\frac{1}{2}$). Then its moment generating function satisfies

$$\mathbb{E}\left(e^{Y_n s}\right) := 2^{-n} + \sum_{1\leqslant m\leqslant n} 2^{m-n-1}Q_{n,m}(s).$$

Theorem 11. *The random variables Y_n are asymptotically normally distributed*

$$\mathbb{P}\left(\frac{Y_n - \nu_n}{\varsigma_n} \leqslant x\right) \to \Phi(x),$$

with mean ν_n and variance ς_n asymptotic to

$$\begin{aligned}\nu_n &= \frac{e^c-1}{2c^2}n^2 + \frac{(c-2)e^c+2}{4c}n + O(1) \\ \varsigma_n &= \frac{e^{2c}-1}{8c^3}n^3 + \frac{3e^{2c}(2c-3) - 8e^c + 17}{16c^2}n^2 + O(n),\end{aligned} \tag{71}$$

respectively.

In particular, we also have, by replacing the exact mean and variance by the corresponding asymptotic approximations

$$\mathbb{P}\left(\frac{Y_n - \frac{e^c-1}{2c^2}n^2}{\sqrt{\frac{e^{2c}-1}{8c^3}n^3}} \leqslant x\right) \to \Phi(x).$$

Proof. By (61), we have

$$\nu_n = \sum_{1\leqslant m\leqslant n} 2^{-n+m-1}\nu_{n,m} = \frac{q}{2p^2}\left(q^{-n}-1\right),$$

and then the first estimate in (71) follows. Similarly, by (66),

$$\varsigma_n^2 = \sum_{0\leqslant m\leqslant n} 2^{-n+m-1}\mathbb{E}(Y_{n,m}^2) - \nu_n^2 = \frac{3q^2}{4p^3(1+q)}\left(q^{-2n}-1\right) - \nu_n,$$

and the second estimate in (71) also follows.

For the asymptotic normality, we consider the characteristic function

$$\mathbb{E}\left(e^{\frac{Y_n-\nu_n}{\varsigma_n}it}\right) = 2^{-n} + \sum_{0\leqslant m<n} 2^{-n+m-1}Q_{n,m}\left(\frac{it}{\varsigma_n}\right)e^{-\frac{\nu_n}{\varsigma_n}it}.$$

We split the sum into two parts: $0 \leqslant n-m \leqslant n^{\frac{1}{3}}$ and $1 \leqslant m < n-n^{\frac{1}{3}}$. Observe that when $n-m \leqslant n^{\frac{1}{3}}$, we have the uniform estimate

$$\nu_n - \nu_{n,m} = O(n|n-m+1|) \quad \text{and} \quad \varsigma_n^2 - \varsigma_{n,m}^2 = O\left(n^2|n-m+1|\right),$$

by (63) and (64). We then have the local expansion (see (70))

$$Q_{n,m}\left(\frac{it}{\varsigma_n}\right)e^{-\frac{\nu_n}{\varsigma_n}it} = \exp\left(\frac{\nu_{n,m}-\nu_n}{\varsigma_n}it - \frac{\varsigma_{n,m}^2}{2\varsigma_n^2}t^2 + O\left(\frac{|t|^3}{n^{\frac{3}{2}}}\right)\right).$$

Thus

$$\begin{aligned}
Q_{n,m}\left(\frac{it}{\varsigma_n}\right)e^{-\frac{\nu_n}{\varsigma_n}it} &= \exp\left(-\frac{t^2}{2} + O\left(\frac{|n-m+1|}{\sqrt{n}}|t| + \frac{|n-m+1|}{n}t^2\right)\right)\\
&= \exp\left(-\frac{t^2}{2} + O\left(n^{-\frac{1}{6}}|t| + n^{-\frac{2}{3}}|t|^2\right)\right)\\
&= e^{-\frac{t^2}{2}}(1+o(1)),
\end{aligned}$$

uniformly in m. Consequently,

$$\sum_{n-n^{\frac{1}{3}}\leqslant m\leqslant n} 2^{-n+m-1}Q_{n,m}\left(\frac{it}{\varsigma_n}\right)e^{-\frac{\nu_n}{\varsigma_n}it} = e^{-\frac{t^2}{2}}(1+o(1)).$$

The remaining part is negligible since $|Q_{n,m}(e^{it/\sigma})| \leqslant 1$ and

$$\sum_{1\leqslant m\leqslant n-n^{\frac{1}{3}}} 2^{-n+m-1}Q_{n,m}\left(\frac{it}{\varsigma_n}\right)e^{-\frac{\nu_n}{\varsigma_n}it} = O\left(\sum_{m>n^{\frac{1}{3}}} 2^{-m}\right) = O\left(2^{-n^{\frac{1}{3}}}\right).$$

We conclude that

$$\mathbb{E}\left(e^{\frac{Y_n-\nu_n}{\varsigma_n}it}\right) \to e^{-\frac{t^2}{2}},$$

which implies the convergence in distribution of $\frac{Y_n-\nu_n}{\varsigma_n}$ to the standard normal distribution. ∎

References

[1] A. Auger and B. Doerr (Eds.), *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, World Scientific, 2011.

[2] T. Bäck, The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm, in Proceeding of the 2nd Conference on Parallel Problem Solving from Nature (PPSN-II), Brussels, Belgium, September 28–30, 1992.

[3] H.-G. Beyer, H.-P. Schwefel and I. Wegener, How to analyse evolutionary algorithms, *Theoretical Computer Science*, **287**:1 (2002), 101–130.

[4] S. Böttcher, B. Doerr and F. Neumann, Optimal fixed and adaptive mutation rates for the LeadingOnes problem, *PPSN XI*, Part I, 2010, LNCS 6238, 1–10.

[5] C. A. Coello Coello, Evolutionary multiobjective optimization: a historical view of the field, *IEEE Computational Intelligence Magazine*, **1**:1 (2006), 28–36.

[6] L. Comtet, *Advanced Combinatorics, The art of finite and infinite expansions*, Revised and enlarged edition, D. Reidel Publishing Co., Dordrecht, 1974.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[8] B. Doerr, M. Fouz and C. Witt, Quasirandom evolutionary algorithms, In *Proc. GECCO'10* (2010), pp. 1457–1464, ACM Press.

[9] B. Doerr, M. Fouz and C. Witt, Sharp bounds by probability-generating functions and variable drift, in *Proc. GECCO'11* (2011), pp. 2083–2090.

[10] B. Doerr, D. Johannsen and C. Winzen, Multiplicative drift analysis, *Algorithmica*, **64** (2012), 673–697.

[11] B. Doerr, D. Johannsen, T. Kotzing, F. Neumann, and M. Theile, More effective crossover operators for the all-pairs shortest path problem, *Theoretical Computer Science* **471**, (2013) 12–26.

[12] B. Doerr and S. Pohl, Run-time analysis of the $(1+1)$ evolutionary algorithm optimizing linear functions over a finite alphabet, in *Proc. GECCO'12* (2012), pp. 1317–1324.

[13] S. Droste, T. Jansen and I. Wegener, A rigorous complexity analysis of the $(1+1)$ evolutionary algorithm for linear functions with Boolean inputs, in *Proc. IEEE International Conference on Evolutionary Computation ICEC'98* (IEEE Press, Piscataway, NJ, 1998), 499–504.

[14] S. Droste, T. Jansen and I. Wegener, On the analysis of the $(1+1)$ evolutionary algorithm, *Theoretical Computer Science*, **276**:1-2 (2002), 51–81.

[15] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, Cambridge, 2009.

[16] J. Garnier, L. Kallel and M. Schoenauer, Rigorous hitting times for binary mutations, *Evolutionary Computation*, **7**:2 (1999) 167–203.

[17] J. He and X. Yao, Drift analysis and average time complexity of evolutionary algorithms, *Artificial Intelligence*, **127** (2001), 57–85.

[18] J. He and X. Yao, From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, **6**:5 (2002), 495–511.

[19] R. V. Hogg, J. W. McKean and A. T. Craig, *Introduction to Mathematical Statistics*, 7th Edition, Pearson Education Inc., NY, 2013.

[20] J. Horn, Multicriterion decision making, *Handbook of Evolutionary Computation*, Volume 1, F1.9:1–F1.9:15, IOP Publishing Ltd. and Oxford University Press, 1997.

[21] J. Horn, D. E. Goldberg and K. Deb, Long path problems, in *Parallel Problem Solving from Nature PPSN III*, edited by Y. Davidor, H.-P. Schwefel, R. Männer, *Lecture Notes in Computer Science*, Vol. 866, Springer, Berlin, 1994, pp. 149–158.

[22] H.-K. Hwang, On convergence rates of central limit theorems for combinatorial structures, *European Journal of Combinatorics*, **19** (1998), 329–343

[23] J. Jagerskupper, Combining Markov-chain analysis and drift analysis: the $(1 + 1)$ evolutionary algorithm on linear functions reloaded, *Algorithmica* **59** (2011), 409–424.

[24] T. Jansen and I. Wegener, The analysis of evolutionary algorithms—A proof that crossover really can help, *Algorithmica*, **34** (2002), 47–66.

[25] T. Jansen and I. Wegener, Real royal road functions—where crossover provably is essential, *Discrete Applied Mathematics*, **149** (2005), 111–125.

[26] V. Ladret, Asymptotic hitting time for a simple evolutionary model of protein folding, *J. Appl. Probab.*, **42** (2005), 39–51.

[27] H. Mühlenbein, How genetic algorithms really work: mutation and hill-climbing, in *Proc. PPSN'92*, 15–26, 1992.

[28] F. Neumann and C. Witt (2009), Runtime analysis of a simple ant colony optimization algorithm, *Algorithmica*, **54** (2009) 243–255.

[29] F. Neumann and C. Witt, *Bioinspired Computation in Combinatorial Optimization—Algorithms and Their Computational Complexity*, Springer, 2010.

[30] V. V. Petrov, *Sums of Independent Random Variables*, Springer-Verlag, New York-Heidelberg, 1975 (translated from the Russian by A. A. Brown).

[31] G. Rudolph, How mutation and selection solve long-path problem in polynomial expected time, *Evolutionary Computation* **4**:3 (1997), 195–205.

[32] D. Sudholt, General lower bounds for the running time of evolutionary algorithms, In *Proc. PPSN'10*, Lecture Notes in Computer Science, Volume 6238, pp. 124–133 (2010).

[33] D. Sudholt, C. Witt (2010), Runtime analysis of a binary particle swarm optimizer, *Theoretical Computer Science*, **411** (2010) 2084–2100.

[34] C. Witt, Runtime analysis of the $(\mu + 1)$ EA on simple pseudo-Boolean functions, *Evolutionary Computation*, **14** (2006), 65–86.

[35] C. Witt, Tight bounds on the optimization time of a randomized search heuristic on linear functions, *Combinatorics, Probability and Computing*, **22**:2 (2013), 294–318.

Appendix. A. Some properties of $S_r(z)$.

We collected here some interesting expressions for $S_r(z)$.

We begin with proving that all S_r can be expressed in terms of S_0 and the two modified Bessel functions

$$\bar{I}_0(\alpha) := I_0\left(2\sqrt{\alpha(1-\alpha)}\right) = \sum_{\ell \geqslant 0} \frac{\alpha^\ell (1-\alpha)^\ell}{\ell!\ell!},$$

$$\bar{I}_1(\alpha) := \sqrt{\frac{\alpha}{1-\alpha}} I_1\left(2\sqrt{\alpha(1-\alpha)}\right) = \sum_{\ell \geqslant 1} \frac{\alpha^\ell (1-\alpha)^{\ell-1}}{\ell!(\ell-1)!}.$$

The starting point is the obvious relation ($E_r(z) := \sum_{\ell \geqslant 1} \ell^r z^{\ell-1}$)

$$E_r(z) = zE'_{r-1}(z) + E_{r-1}(z) \qquad (r \geqslant 1).$$

Applying the integral representation (19) and integration by parts, we have

$$S_r(\alpha) = \frac{1}{2\pi i} \oint_{|z|=c} \left(\frac{\alpha}{z} - (1-\alpha)z\right) E_{r-1}(z) e^{\frac{\alpha}{z} + (1-\alpha)z}\, \mathrm{d}z.$$

By the same argument used for Corollary 1, we deduce the recurrence

$$S_r(\alpha) = \alpha \bar{I}_0(\alpha) + \sum_{0 \leqslant j < r} \binom{r-1}{j} S_j(\alpha) \left(\alpha + (-1)^{r-j}(1-\alpha)\right), \tag{72}$$

for $r \geqslant 2$ with

$$S_1(\alpha) = (2\alpha - 1)S_0(\alpha) + \alpha \bar{I}_0(\alpha) + (1-\alpha)\bar{I}_1(\alpha).$$

A closed-form expression can be obtained for the recurrence (72) but it is very messy. More precisely, let $f(z) := \sum_{r \geqslant 0} S_r(\alpha) z^r / r!$. Then f satisfies the first-order differential equation

$$f'(z) = (\alpha e^z - (1-\alpha)e^{-z})\, f(z) + \alpha \bar{I}_0(\alpha) + (1-\alpha)\bar{I}_1(\alpha).$$

The solution to the differential equation with the initial condition $f(0) = S_0(\alpha)$ is given by

$$\begin{aligned} f(z) = {} & S_0(\alpha) e^{\alpha(e^z - e^{-z}) + e^{-1} - 1} \\ & + e^{\alpha e^z + (1-\alpha)e^{-z}} \int_0^z \left(\alpha \bar{I}_0(\alpha) e^u + (1-\alpha)\bar{I}_1(\alpha)\right) e^{-\alpha e^u - (1-\alpha)e^{-u}}\, \mathrm{d}u. \end{aligned}$$

This implies that $S_r(\alpha)$ has the general form

$$S_r(\alpha) = p_r^{[0]}(\alpha)\bar{I}_0(\alpha) + p_r^{[1]}(\alpha)\bar{I}_1(\alpha) + p_r^{[2]}(\alpha)S_0(\alpha) \qquad (r \geqslant 1),$$

where the $p_r^{[i]}$ are polynomials of α of degree r. Closed-form expressions can be derived but are less simpler than the recurrence (72) for small values of r.

On the other hand, the same argument also leads to

$$S_r'(\alpha) = \bar{I}_0(\alpha) + \sum_{0\leqslant j<r} \binom{r}{j} S_j(\alpha)\left(1-(-1)^{r-j}\right) \qquad (r \geqslant 1).$$

In particular, $S_1'(\alpha) = \bar{I}_0(\alpha) + 2S_0(\alpha)$. Note that

$$S_0'(\alpha) = \bar{I}_0(\alpha) + \bar{I}_1(\alpha),$$

implying that

$$S_0(\alpha) = \int_0^\alpha \left(\bar{I}_0(u) + \bar{I}_1(u)\right) \mathrm{d}u.$$

This in turn gives

$$S_1(\alpha) = \int_0^\alpha \left((1+2(\alpha-u))\bar{I}_0(u) + 2(\alpha-u)\bar{I}_1(u)\right) \mathrm{d}u.$$

This expression can be further simplified by taking second derivative with respect to α of the integral representation

$$S_1(\alpha) = \frac{1}{2\pi i}\oint_{|z|=c} \frac{e^{\frac{\alpha}{z}+(1-\alpha)z}}{(1-z)^2}\,\mathrm{d}z,$$

giving

$$S_1''(\alpha) = 2\bar{I}_0(\alpha) + \alpha^{-1}\bar{I}_1(\alpha),$$

which implies that (with $S_1(0) = 0$, $S_1'(0) = 1$)

$$S_1(\alpha) = \int_0^\alpha (\alpha-u)\left(2\bar{I}_0(u) + u^{-1}\bar{I}_1(u)\right) \mathrm{d}u$$

Similarly, since $S_2' = I_0 + 4S_1$, we have

$$S_2(\alpha) = \int_0^\alpha (1+4(\alpha-u)(\alpha-u+1))\bar{I}_0(u)\,\mathrm{d}u + 4\int_0^\alpha (\alpha-u)^2\bar{I}_1(u)\,\mathrm{d}u.$$

These expressions show not only the intimate connections of S_r to Bessel functions but also their rich algebraic aspects.

We now consider $S_r(1-\alpha)$. By the same integral representation and a change of variables, we see that, for $r \geqslant 1$,

$$(-1)^r S_r(\alpha) + S_r(1-\alpha) = [z^0]E_r(1-z)e^{\frac{\alpha}{1-z}+(1-\alpha)(1-z)}.$$

Now

$$E_r(1-z) = r![w^r]\frac{e^w}{1-(1-z)e^w} = \sum_{0\leqslant j\leqslant r} (-1)^{r+j} j!\,\mathrm{Stirling}_2(r,j)z^{-j-1}.$$

Thus we deduce the identity (for $r \geqslant 1$)

$$(-1)^r S_r(\alpha) + S_r(1-\alpha)$$
$$= e \sum_{0 \leqslant \ell \leqslant r} (-1)^{r+\ell} \ell! \, \mathrm{Stirling}_2(r,\ell) \sum_{\substack{0 \leqslant h \leqslant \ell \\ 0 \leqslant j < h/2}} \binom{h-j-1}{j-1} \frac{(2\alpha-1)^{\ell-h}\alpha^j}{(\ell-h)!\,j!}$$

or

$$(-1)^r S_r(\alpha) + S_r(1-\alpha)$$
$$= e \sum_{0 \leqslant \ell \leqslant r} (-1)^{r+\ell} \ell! \, \mathrm{Stirling}_2(r,\ell) \left(\frac{(\alpha-1)^\ell}{\ell!} + \sum_{\substack{0 \leqslant h \leqslant \ell \\ 0 \leqslant j < h}} \binom{h-1}{j} \frac{\alpha^{h-j}(\alpha-1)^{\ell-h}}{(\ell-h)!(h-j)!} \right).$$

Note that for $r = 0$

$$S_0(\alpha) + S_0(1-\alpha) = e - \bar{I}_0(\alpha).$$

In particular, this gives $S(\frac{1}{2}) = \frac{1}{2}(e - I_0(1)) \approx 0.726107$. For $r \geqslant 1$

$$S_1(\alpha) - S_1(1-\alpha) = e(2\alpha - 1)$$
$$S_2(\alpha) + S_2(1-\alpha) = e(4\alpha^2 - 4\alpha + 2)$$
$$S_3(\alpha) - S_3(1-\alpha) = e(8\alpha^3 - 12\alpha^2 + 14\alpha - 5)$$
$$S_4(\alpha) - S_4(1-\alpha) = e(16\alpha^4 - 32\alpha^3 + 64\alpha^2 - 48\alpha + 15).$$

Appendix. B. Closeness of the approximation (25) for $\mu^*_{n,m}$: graphical representations

The successive improvements attained by adding more terms on the right-hand side of (25) can be viewed in Figures 5 and 6.

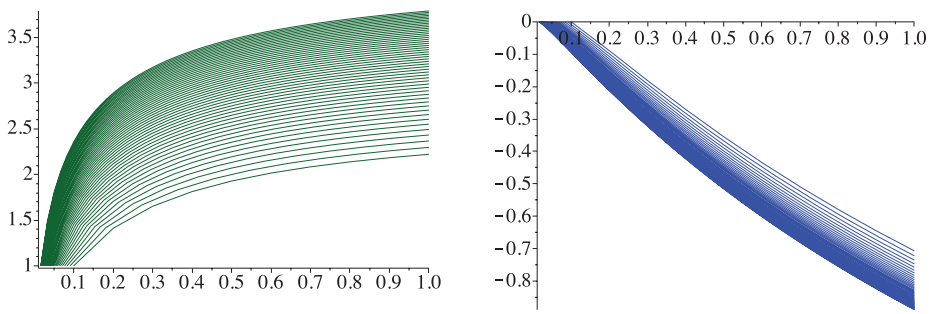


Figure 5: *Left: the sequence $\mu^*_{n,m}$ for $1 \leqslant m \leqslant n$ and $n = 10, \ldots, 60$; Right: the difference between $\mu^*_{n,m} - H_m$ for n, m in the same ranges.*

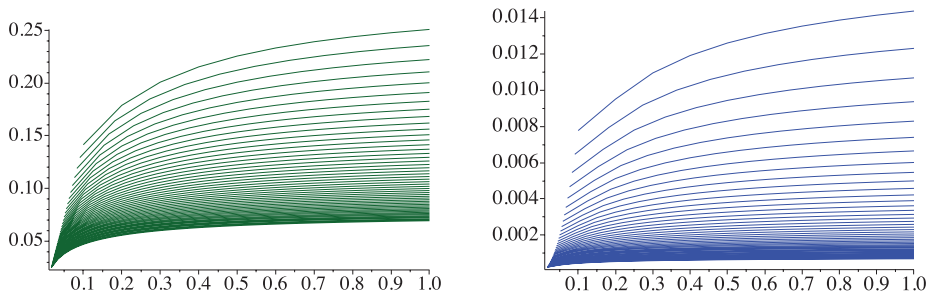


Figure 6: *The difference* $\mu^*_{n,m} - (H_m + \phi_1(\frac{m}{n}))$ *(left) and* $\mu^*_{n,m} - \left(H_m + \phi_1(\frac{m}{n}) + \frac{H_m + \phi_2(\frac{m}{n})}{n}\right)$ *(right) for* $1 \leqslant m \leqslant n$ *and* $n = 10, \ldots, 60$.

Appendix. C. Asymptotic expansions for $V^*_{n,m}$ for small m and the refined approximation (45) to $V^*_{n,m}$

Recall that

$$V^*_{n,m} = \frac{e_n^2}{n^2}\big(\mathbb{V}(X_{n+1,m}) + \mathbb{E}(X_{n+1,m})\big).$$

This sequence satisfies $V^*_{n,0} = 0$, and for $1 \leqslant m \leqslant n$,

$$\sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell}\left(V^*_{n,m} - V^*_{n,m-\ell}\right) = T^*_{n,m}, \tag{73}$$

where

$$T^*_{n,m} := \sum_{1 \leqslant \ell \leqslant m} \lambda^*_{n,m,\ell}\left(\mu^*_{n,m} - \mu^*_{n,m-\ell}\right)^2.$$

From this recurrence, we obtain the following expansions.

$$\begin{aligned}
V^*_{n,1} &= 1,\\
V^*_{n,2} &= \tfrac{5}{4} - \tfrac{1}{2}n^{-1} + \tfrac{3}{4}n^{-2} - \tfrac{5}{4}n^{-3} + \tfrac{31}{16}n^{-4} - 3n^{-5} + O(n^{-6}),\\
V^*_{n,3} &= \tfrac{49}{36} - \tfrac{17}{18}n^{-1} + \tfrac{52}{27}n^{-2} - \tfrac{139}{36}n^{-3} + \tfrac{3157}{432}n^{-4} - \tfrac{361}{27}n^{-5} + O(n^{-6}),\\
V^*_{n,4} &= \tfrac{205}{144} - \tfrac{95}{72}n^{-1} + \tfrac{1489}{432}n^{-2} - \tfrac{1243}{144}n^{-3} + \tfrac{33091}{1728}n^{-4} - \tfrac{28979}{864}n^{-5} + O(n^{-6}).
\end{aligned}$$

Observe that the leading constant terms are exactly given by

$$\left\{H_m^{(2)}\right\} = \left\{1, \tfrac{5}{4}, \tfrac{49}{36}, \tfrac{205}{144}, \tfrac{5269}{3600}, \tfrac{5369}{3600}, \ldots\right\}.$$

These expansions suggest the general form

$$V^*_{n,m} \approx H_m^{(2)} + \sum_{k \geqslant 1} \frac{\tilde{d}_k(m)}{n^k}.$$

With this form using the technique of matched asymptotics, we are then led to the following explicit expressions.

$$\begin{aligned}
\tilde{d}_1(m) &= -2H_m + 2H_m^{(2)}, \quad \text{for } m \geqslant 0,\\
\tilde{d}_2(m) &= -\tfrac{11}{2}H_m + \tfrac{7}{3}H_m^{(2)} + \tfrac{7}{12} + \tfrac{11}{4}m, \quad \text{for } m \geqslant 2,\\
\tilde{d}_3(m) &= -\tfrac{73}{9}H_m + \tfrac{7}{3}H_m^{(2)} + \tfrac{1}{6} + \tfrac{239}{36}m - \tfrac{49}{36}m^2, \quad \text{for } m \geqslant 2,\\
\tilde{d}_4(m) &= -\tfrac{1349}{144}H_m + 2H_m^{(2)} + \tfrac{197}{144} + \tfrac{14135}{1728}m - \tfrac{6283}{2880}m^2 + \tfrac{2473}{4320}m^3, \quad \text{for } m \geqslant 4.
\end{aligned}$$

The above expansions for small m suggest the more uniform asymptotic expansion for $V_{n,m}^*$ for $1 \leqslant m \leqslant n$

$$V_{n,m}^* \sim H_m^{(2)} + \sum_{k\geqslant 1} \frac{a_k H_m + \psi_k(\alpha) + c_k H_m^{(2)}}{n^k}, \tag{74}$$

in the sense that when omitting all terms with indices $k > K$ introduces an error of order $n^{-(K+1)}H_m$; furthermore, the expansion holds uniformly for $K \leqslant m \leqslant n$. We elaborate this approach by carrying out the required calculations up to $k = 2$, which then characterizes particularly the constant a_2 and the function $\psi_2(z)$.

We start with the formal expansion (74) and expand in recurrence (73) all terms for large $m = \alpha n$ in decreasing powers of n; we then match the coefficients of $n^{-(K+1)}$ on both sides for each $K \geqslant 1$. To specify the initial condition $\psi_K(0)$ we incorporate the information from the asymptotic expansion for $V_{n,K}^*$ (obtained by exact solution). With this algorithmic approach it is possible to determine the coefficients a_k and c_k and the functions $\psi_k(z)$ successively one after another. We remark that a formalization of this procedure at the generating function level as carried out for the expectation in Section 4.4 could be given also, but here we do not pursue this any further.

We use the expansions

$$\begin{aligned}
\phi\Big(\frac{m}{n}\Big) - \phi\Big(\frac{m-\ell}{n}\Big) &= \phi'(\alpha)\frac{\ell}{n} - \phi''(\alpha)\frac{\ell^2}{2n^2} + \phi'''(\alpha)\frac{\ell^3}{6n^3} + \cdots,\\
H_m - H_{m-\ell} &= \frac{\ell}{\alpha\, n} + \frac{\ell(\ell-1)}{2\alpha^2\, n^2} + \frac{\ell(\ell-1)(2\ell-1)}{6\alpha^3\, n^3} + \cdots,\\
H_m^{(2)} - H_{m-\ell}^{(2)} &= \frac{\ell}{\alpha^2\, n^2} + \frac{\ell(\ell-1)}{\alpha^3\, n^3} + \frac{\ell(\ell-1)(2\ell-1)}{2\alpha^4\, n^4} + \cdots
\end{aligned}$$

as well as those for $\mu_{n,m}^*$ and $\bar{\Lambda}_{n,m}^{(r)}$ in (41) and (16), respectively. The expansion of the right-hand side of (73) then starts as follows.

$$T_{n,m}^* = \frac{T_1(\alpha)}{n^2} + \frac{T_2(\alpha)}{n^3} + \cdots,$$

where

$$\begin{aligned}
T_1(z) &= \frac{S_2(z)}{S_1^2(z)},\\
T_2(z) &= -\frac{S_2^2(z)S_1'(z)}{S_1^4(z)} + \frac{S_3(z)S_1'(z)}{S_1^3(z)} + \frac{2S_0(z)S_2(z)}{S_1^3(z)} + \frac{S_0(z)}{S_1^2(z)} - \frac{S_2(z)}{2S_1^2(z)} - \frac{2}{S_1(z)}.
\end{aligned}$$

For the left-hand side of (73), the asymptotic form (74) leads to

$$\sum_{1\leqslant \ell\leqslant m} \lambda^*_{n,m,\ell}\left(V^*_{n,m} - V^*_{n,m-\ell}\right) = \frac{V_1(\alpha)}{n^2} + \frac{V_2(\alpha)}{n^3} + \cdots,$$

where

$$\begin{aligned}
V_1(z) &= \left(\frac{1}{z^2} + \frac{a_1}{z} + \psi_1'(z)\right) S_1(z),\\
V_2(z) &= \left(-\frac{1}{z^2} - \frac{a_1}{z} - \psi_1'(z)\right) S_0(z)\\
&\quad + \left(-\frac{1}{z^3} - \frac{1}{2z^2} + \frac{c_1}{z^2} - \frac{a_1}{2z^2} - \frac{a_1}{2z} + \frac{a_2}{z} - \frac{\psi_1'(z)}{2} + \psi_2'(z)\right) S_1(z)\\
&\quad + \left(\frac{1}{z^3} + \frac{a_1}{2z^2} - \frac{\psi_1''(z)}{2}\right) S_2(z).
\end{aligned}$$

Observe that all functions $V_k(z)$, $T_k(z)$ have a simple pole at $z = 0$.

We match the terms in the expansion and consider $V_1(z) = T_1(z)$. First we compare the first two terms of the Laurent expansions of both functions. Using (22), we get

$$\begin{aligned}
V_1(z) &= \frac{1}{z} + \left(\frac{3}{2} + a_1\right) + O(z),\\
T_1(z) &= \frac{1}{z} - \frac{1}{2} + O(z),
\end{aligned}$$

and by matching the two constant terms, we see that $a_1 = -2$. The equation $V_1(z) = T_1(z)$ characterizes then the function $\psi_1'(z)$ of the form

$$\psi_1'(z) = \frac{S_2(z)}{S_1^3(z)} - \frac{1}{z^2} + \frac{2}{z}.$$

Next we consider $V_2(z) = T_2(z)$ and obtain

$$\begin{aligned}
V_2(z) &= \left(-\frac{1}{2} + c_1\right)\frac{1}{z} + \left(-\frac{5}{12} - a_1 + \frac{3c_1}{2} + a_2\right) + O(z),\\
T_2(z) &= \frac{3}{2z} - \frac{11}{12} + O(z),
\end{aligned}$$

and thus, by matching the terms and using the values already computed in the first-order approximation for $V^*_{n,m}$, $c_1 = 2$ and $a_2 = -\frac{11}{2}$. Then the function $\psi_2'(z)$ can be characterized by equating $V_2(z) = T_2(z)$, which then gives

$$\begin{aligned}
\psi_2'(z) = &-\frac{5S_2^2(z)S_1'(z)}{2S_1^5(z)} + \frac{S_3(z)S_1'(z)}{S_1^4(z)} + \frac{3S_2(z)S_0(z)}{S_1^4(z)} + \frac{S_2(z)S_2'(z)}{2S_1^4(z)} + \frac{S_0(z)}{S_1^3(z)} - \frac{2}{S_1^2(z)}\\
&+ \frac{1}{z^3} - \frac{3}{z^2} + \frac{11}{2z}.
\end{aligned}$$

All constants and functions here match with those obtained earlier in previous paragraphs, and we can pursue the same calculations further and obtain finer approximations. For example, we have $c_2 = \frac{7}{3}$. But the calculations are long and laborious.

Finally, it remains to determine the constant terms in the Taylor expansion of the functions $\psi_k(z)$ by adjusting them to the expansion of $V^*_{n,m}$ for small m. This yields $\psi_1(0) = 0$, and

$$\psi_2(0) = \frac{7}{12}.$$

This characterizes the function $\psi_2(z)$ in Theorem 4 as follows.

$$\psi_2(z) = \frac{7}{12} + \int_0^z \Bigg[-\frac{5S_2^2(t)S_1'(t)}{2S_1^5(t)} + \frac{S_3(t)S_1'(t)}{S_1^4(t)} + \frac{3S_2(t)S_0(t)}{S_1^4(t)} + \frac{S_2(t)S_2'(t)}{2S_1^4(t)} + \frac{S_0(t)}{S_1^3(t)} - \frac{2}{S_1^2(t)} + \frac{1}{t^3} - \frac{3}{t^2} + \frac{11}{2t} \Bigg] \mathrm{d}t.$$

In particular, the first few terms in the Taylor expansion of $\psi_2(z)$ are given by

$$\psi_2(z) = \tfrac{7}{12} + \tfrac{239}{36}z - \tfrac{6283}{2880}z^2 - \tfrac{4529}{3600}z^3 + \tfrac{9283591}{1814400}z^4 - \tfrac{137478949}{14112000}z^5 + \cdots.$$

Appendix. D. Closeness of the approximation (45) for $V^*_{n,m}$: graphical representations

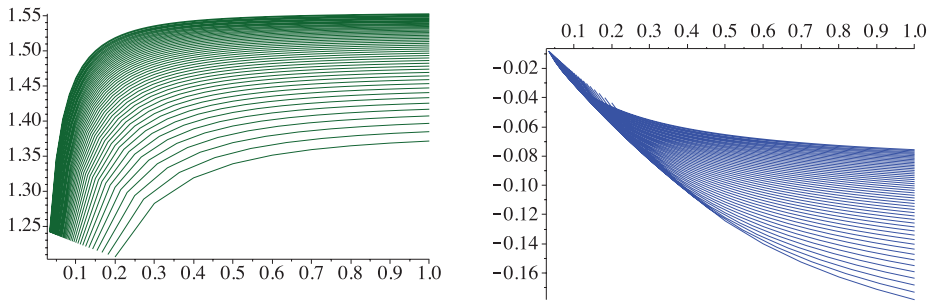


Figure 7: *Left: the sequence $V^*_{n,m}$ for $2 \leqslant m \leqslant n$ and $n = 10, \ldots, 60$; Right: the difference between $V^*_{n,m} - H_m^{(2)}$ for n, m in the same ranges.*

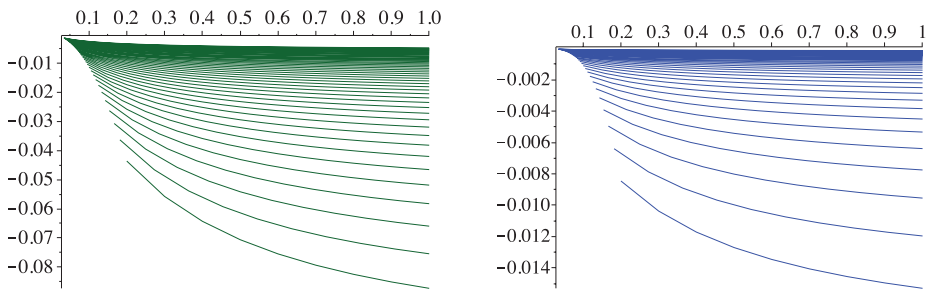


Figure 8: *The difference $V^*_{n,m} - \left(H_m^{(2)} + \frac{-2H_m+\psi_1(\alpha)+2H_m^{(2)}}{n}\right)$ (left) and $V^*_{n,m} - \left(H_m^{(2)} + \frac{-2H_m+\psi_1(\alpha)+2H_m^{(2)}}{n} + \frac{-\frac{11}{2}H_m+\psi_2(\alpha)+\frac{7}{3}H_m^{(2)}}{n^2}\right)$ (right) for $2 \leqslant m \leqslant n$ and $n = 10, \ldots, 60$.*

Résumé

On étudie en combinatoire les objets munis d'une taille (la taille dans le cadre informatique peut se traduire par exemple par la mémoire occupée par l'objet). On appelle *classe combinatoire* un ensemble d'objets qui pour toute taille possède un nombre fini d'éléments. On peut par exemple considérer les textes régis par une certaine grammaire, dans ce cas la taille est le nombre de caractères, ou des arbres avec comme taille le nombre de nœuds. Une méthode naturelle pour décrire les classes, la *méthode symbolique*, consiste à décomposer les objets en sous-objets plus élémentaires à l'aide d'opérateurs (tels que l'union disjointe, le produit cartésien,...). On peut ensuite traduire ces décompositions sur des séries formelles.

Le premier volet de résultats présentés dans cette thèse traite de la méthode symbolique et de son utilisation. On y présente des résultats asymptotiques sur des modèles d'arbres croissants issus de la théorie de la concurrence, puis une discussion sur comment décomposer certains opérateurs en réplications élémentaires. Le deuxième volet de résultats s'intéresse au sujet de la génération aléatoire uniforme d'objets dans une classe donnée. On montre tout d'abord comment générer des structures croissantes en adaptant les méthodes de génération récursive classiques aux opérateurs de produit croissant. On présente ensuite des résultats sur la génération de Boltzmann, avec une comparaison quantitative de deux méthodes, puis une extension permettant de conserver les propriétés d'uniformité de la génération en utilisant des approximations.

Title

On usage of operators in combinatorics : construction, analysis and random generation.

Abstract

We study in combinatorics objects with a size (size in informatics setting can be the memory space used to represent an object). We call a *combinatorial class* a set of objects who for a given size have only a finite number of elements. We can for example look at text generated by a given grammar, with the number of characters as size, or trees with the number of nodes as size. A natural way of describing classes, the *symbolic method*, consists in decomposing objects in more elementary sub-objects with operators (disjoint union, cartesian product,...). Then we can translate theses decompositions to formal power series.

The first batch of results in this thesis deals with the symbolic method and its usage. We present asymptotic results on models of increasing trees coming from concurrency theory, then we discuss on how to decompose some operators in elementary replications. The second batch of results deals with uniform random generation of objects in a given class. We first show how to generate increasing structures by adapting the recursive generation techniques to increasing product operators. Then we present two results on Boltzmann generation, with a quantitative comparison of two methods and with an extension allowing us to use approximatives values while retaining the uniformity of the generation.

Informatique / Institut Galilée

Mots clés :

Mots clés : combinatoire analytique, séries génératrices, génération aléatoire, méthode symbolique
Keywords : analytic combinatorics, generating function, random generation, symbolic method