

# **Publications du Laboratoire de Combinatoire et d' Informatique Mathématique**

**27**

---

**Édité par P. Leroux**

## **Colloque LaCIM 2000 Combinatoire, Informatique et Applications**

7 — 10 septembre 2000  
Université du Québec à Montréal

### **Actes**

---



**Université du Québec à Montréal**

Le Colloque LaCIM 2000 - Combinatoire, Informatique et Applications a été organisé par le Laboratoire de combinatoire et d'informatique mathématique (LaCIM), Université du Québec à Montréal du 7 au 10 septembre 2000, au Pavillon Sherbrooke, 200, rue Sherbrooke Ouest, Montréal (Québec), Canada.

### Comité d'organisation

Pierre Leroux	<i>LaCIM, UQAM</i> , président
Robert Bédard	<i>LaCIM, UQAM</i>
Srecko Brlek	<i>LaCIM, UQAM</i> ,
Manon Blais	<i>LaCIM, UQAM</i> , secrétaire

### Comité Scientifique

Pierre Leroux, *UQAM*, président  
 André Arnold, *Un. Bordeaux I*; Anne Bergeron, *UQAM*; François Bergeron, *UQAM*, Srecko Brlek, *UQAM*; Gilbert Labelle, *UQAM*; Alain Lascoux, *Un. Marne-la-Vallée*; Bruce Sagan, *Michigan State Un.*; David Sankoff, *Un. de Montréal*; Jiang Zeng, *Un. Lyon 1*.

Avec le soutien financier des organismes suivants:

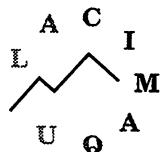


Dépôt légal, deuxième semestre 2000, Bibliothèque nationale du Québec

ISBN 2-89276-179-4 LaCIM Montréal

©LaCIM, Montréal, Septembre 2000.

Laboratoire de combinatoire et d'informatique mathématique  
 Université du Québec à Montréal  
 C.P. 8888, Succ. Centre-Ville  
 Montréal (Québec) Canada  
 H3C 3P8



## AVANT-PROPOS

L'année 2000, année mondiale des mathématiques sous l'égide de l'UNESCO, marque, parmi tant d'autres anniversaires, les dix ans du Laboratoire de Combinatoire et d'Informatique Mathématique de l'UQAM. Les disciplines couvertes par le LaCIM ont connu des développements remarquables au cours des dix dernières années, que ce soit sur le plan théorique, en combinatoire et en informatique, ou au niveau des applications, en analyse et en algèbre, en calcul formel, en géométrie algorithmique, en physique statistique, et, plus récemment, en bio-informatique.

Le but du Colloque LaCIM 2000, Combinatoire, Informatique et Applications, est de faire le point sur quelques-uns de ces développements en réunissant des experts internationaux de ces domaines. Nous souhaitons également marquer le dixième anniversaire du LACIM en donnant la chance aux chercheurs jeunes ou plus expérimentés, de faire connaître leurs résultats récents.

Le programme comprend dix conférences plénières ainsi que 27 communications sélectionnées par le comité scientifique et regroupées dans une séance d'affichage. Nous présentons dans ce volume les résumés des conférences invitées ainsi que les résumés substantiels des communications.

Nous remercions chaleureusement tous les participants et toutes les personnes qui ont contribué à faire de ce colloque un franc succès. Un merci particulier est adressé à nos commanditaires:

- Le Ministère de la recherche, de la science, et de la technologie (Québec),
- Le Centre de Recherches Mathématiques (Montréal),
- L'Université du Québec à Montréal.

Pierre Leroux, pour les comités de programme et d'organisation.

## **FOREWORD**

Among the celebrations related to UNESCO World Mathematical Year 2000 , is the 10th anniversary of the Laboratoire de Combinatoire et d'Informatique Mathématique (LaCIM). In the last ten years many important developments have occurred in the areas covered by LaCIM, both at the theoretical level, in combinatorics and in computer science, and in their applications, in algebra and analysis, in statistical physics, in computational geometry, in computer algebra, and, more recently, in bioinformatics.

The purpose of the LaCIM 2000 Conference, Combinatorics, Computer Science and Applications, is to review some of these developments by bringing together international experts. We also wish to mark our 10th anniversary by giving the opportunity to all researchers, young or more experienced, to present their recent results.

The conference program includes ten plenary talks and 27 contributed communications selected by the scientific committee and regrouped in a poster session. This volume contains the abstracts of the plenary talks and the extended abstracts of the contributed communications.

We warmly thank all participants and all those who have contributed in making this Conference a success. Special thanks are also addressed to our sponsors:

- The Ministère de la recherche, de la science, et de la technologie (Québec),
- The Centre de Recherches Mathématiques (Montréal),
- The Université du Québec à Montréal.

Pierre Leroux, for the program and organizing committees.

## TABLE DES MATIÈRES

<b>Avant-propos .....</b>	<b>iii</b>
<b>Foreword .....</b>	<b>iv</b>
<b>Table des matières .....</b>	<b>v</b>
<b>Conférences invitées .....</b>	<b>1</b>
<b>Jean Berstel</b>	
<i>Formal Properties of XML Grammars and Languages .....</i>	1
<b>Richard Ehrenborg</b>	
<i>Two Permutation Set Statistics .....</i>	3
<b>Nadia El-Mabrouk</b>	
<i>Reconstruction d'un génome dupliqué ancestral .....</i>	4
<b>Dominique Foata</b>	
<i>Permutations d'André et actions de groupes .....</i>	5
<b>Adriano Garsia</b>	
<i>Recent Developments on the Theory of Diagonal Harmonics: Results and Conjectures ....</i>	7
<b>Pierre Lalonde</b>	
<i>q-énumération des matrices à signes alternants n'ayant qu'un seul -1 .....</i>	8
<b>Christophe Reutenauer</b>	
<i>Sous-ensembles rationnels de <math>\mathbb{Z}^k</math>, théorème de Popoviciu et valuation de polyèdres .....</i>	14
<b>Richard Stanley</b>	
<i>Flow Polytopes .....</i>	19
<b>Xavier G. Viennot</b>	
<i>La beauté des mathématiques combinatoires .....</i>	20
<b>Doron Zeilberger</b>	
<i>4<sup>2</sup> proofs of <math>n^{(n-2)}</math> .....</i>	21
<b>Communications .....</b>	<b>23</b>
<b>Ali Aberkane</b>	
<i>Exemples de suites de complexité inférieure à <math>2n</math> .....</i>	23
<b>Didier Arquès, Anne Micheli</b>	
<i>Enumeration of multi-colored rooted maps .....</i>	31
<b>Jean-Christophe Aval</b>	
<i>On certain spaces of lattice diagram polynomials .....</i>	43
<b>Margaret M. Bayer, Gábor Hetyei</b>	
<i>Inequalities for Eulerian posets via doubling .....</i>	53
<b>Anne Bergeron, Sylvie Hamel</b>	
<i>Fast Implementations of Automata Computations .....</i>	61
<b>Miklós Bóna, Bruce E. Sagan, Vincent R. Vatter</b>	
<i>Frequency sequences with no internal zeros .....</i>	71
<b>Michel Bousquet, Cédric Chauve, Gilles Schaeffer</b>	
<i>Énumération et génération aléatoire de cactus m-aires .....</i>	81
<b>Yves Chiricota</b>	
<i>Algorithmes géométriques appliqués à la manipulation de tissus dans un environnement virtuel .....</i>	93

Bob Clarke	
<i>Towards the inverse of a word</i> .....	103
Robert Cori, Dominique Poulalhon	
<i>Bisuites de parking</i> .....	111
Sylvie Corteel, Dominique Gouyou-Beauchamps	
<i>Énumérations de piles de sables</i> .....	119
Jean-Michel Couvreur	
<i>Un point de vue symbolique sur la logique temporelle linéaire</i> .....	131
Emeric Deutsch, Svjetlan Feretic, Marc Noy	
<i>A bijection between diagonally convex directed polyominoes and even trees, and its applications</i> .....	141
Emeric Deutsch, Louis W. Shapiro	
<i>A bijection between ordered trees and 2-Motzkin Paths and its many consequences</i> .....	151
Serge Dulucq, Jean-Guy Penaud	
<i>Interprétation bijective d'une récurrence des nombres de Motzkin</i> .....	165
Francis Gascon	
<i>Énumération de tableaux standards de hauteur bornée</i> .....	171
Louis Kalikow	
<i>Symmetries in trees and parking functions</i> .....	181
Gérald Ksavrelوف, Jiang Zeng	
<i>Nouvelles statistiques de parcours pour les <math>q</math>-nombres de Stirling</i> .....	195
Yvan Le Borgne, Dominique Rossin	
<i>Identité du groupe du Tas de Sable sur des grilles rectangulaires</i> .....	205
Hakan Lennerstad, Lars Lundberg	
<i>Optimal Performance Comparisons for Massively Parallel Multiprocessors</i> .....	217
Claudia Malvenuto, Francesco Pappalardi	
<i>On the Enumeration of Permutation Polynomials</i> .....	233
Marni Mishna	
<i>Cayley Graph Enumeration</i> .....	241
Jean-Guy Penaud, Olivier Roques	
<i>Génération efficace d'un langage de Fibonaci</i> .....	253
Margaret A. Readdy	
<i>The Yuri-Manin ring and its <math>B_n</math>-analogue</i> .....	263
Andrew Rechnitzer	
<i>Squashing animals &amp; anisotropic generating functions</i> .....	273
Robert A. Sulanke	
<i>Counting Lattice Paths with Various Step Sets</i> .....	287
Petko Valtchev, Rokia Missaoui, Pierre Lebrun	
<i>A Fast Algorithm for Building the Hasse Diagram of a Galois Lattice</i> .....	293
Mike Zabrocki	
<i>Generalized ribbons and Macdonald symmetric functions</i> .....	307

# Formal Properties of XML Grammars and Languages

Jean Berstel

Institut Gaspard Monge (IGM), Université Marne-la-Vallée  
and

Luc Boasson

Laboratoire d'informatique algorithmique: fondements et applications (LIAFA),  
Université Denis-Diderot

## Abstract

XML (eXtensible Markup Language) is a format recommended by W3C in order to structure a document. The syntactic part of the language describes the relative position of pairs of corresponding tags. This description is by means of a document type definition (DTD). In addition to its syntactic part, each tag may also have attributes. If the attributes in the tags are ignored, a DTD appears to be a special kind of context-free grammar. The aim of this talk is to study this family of grammars.

One of the consequences will be a better appraisal of the structure of XML documents. It will also show some limitations of the power of expression of XML. Consider for instance an XML-document that consists of a sequence of paragraphs. A first group of paragraphs is being typeset in bold, a second one in italic, and there should be as many paragraphs in bold than in italic. As we shall see, it is not possible to specify this condition by a DTD. This is due to the fact that the context-free grammars corresponding to DTD's are rather restricted.

The main results presented in this talk are two characterizations of XML-languages. The first is set-theoretic. It shows that an XML-language is the biggest language in some class of languages. It relies on the fact that, for each XML-language, there is only one XML-grammar that generates it. The second characterization is syntactic. It shows that XML-languages have a kind of "saturation property".

As usual, these results can be used to show that some languages cannot be XML. This means in practice that, in order to achieve some features of pages, additional nonsyntactic techniques have to be used.

It appears that a new concept plays an important role in XML-languages. This is the notion of surface. The surface of an opening tag  $a$  is the set of sequences of opening tags that are children of  $a$ . The surfaces of an XML-language must be regular sets, and in fact describe the XML-grammar. The characterization results we mentioned heavily rely on surfaces, but the second one also uses the syntactic concept of a context.

We also investigate decision problems. It is decidable whether the language generated by a context-free grammar is well-formed (that is whether it is contained in a set of Dyck primes), but it is undecidable whether there is an XML-grammar for it. On the contrary, it is decidable whether the surfaces of a context-free grammar are finite. At the end, we give some historical remarks. Indeed, several species of context-free grammars investigated in the sixties, such as parenthesis grammars or bracketed grammars are strongly related to XML-grammars. This relationship is sketched.

## Résumé

XML (eXtensible Markup Language) est un format recommandé par le consortium W3C, pour la structuration de documents. La partie syntaxique du langage décrit la position relative de paires de balises dans le texte. Cette syntaxe est présentée, en XML, sous forme d'une description de type de document (DTD). En plus de la partie syntaxique, chaque balise peut aussi porter des attributs. Si l'on ignore les éventuels attributs des tags, une DTD s'apparente à une grammaire context-free d'une forme particulière. Le but de cet exposé est l'étude de cette famille de grammaires.

Une de ses conséquences sera une meilleure compréhension de la structure des documents XML. L'étude montrera aussi quelques limitations dans la puissance d'expression de XML. Considérons par exemple un document XML qui consiste en une suite de paragraphes. Un premier groupe de paragraphes est composé en caractères gras, un deuxième groupe est en italiques, et l'on demande qu'il y ait autant de paragraphes en gras qu'en italiques. Nous verrons qu'il n'est pas possible d'imposer cette dernière contrainte au moyen d'une DTD. Ceci est dû au fait que les grammaires context-free correspondant aux DTD sont plutôt restreintes.

Les résultats principaux présentés dans cet exposé sont deux caractérisations des langages XML. La première est ensembliste. Elle montre qu'un langage XML est l'élément maximal dans une certaine classe de langages. Elle repose sur le fait que, pour chaque langage XML, il n'existe qu'une seule grammaire XML qui l'engendre. La deuxième caractérisation est syntaxique. Elle montre que les langages XML possèdent une certaine propriété de "saturation".

Comme d'habitude, ces résultats peuvent servir à montrer que certains langages ne sont pas XML. Ceci implique qu'en pratique, certains aspects de pages Web ne pourront être réalisés qu'en ayant recours à des techniques non syntaxiques.

Un nouveau concept joue un rôle important dans l'étude des langages XML. Il s'agit de la notion de surface. La surface d'une balise ouvrante *a* est l'ensemble des suites des balises ouvrantes qui sont des enfants de *a*. Les surfaces d'un langage XML doivent être des langages rationnels. En fait, elles décrivent complètement la grammaire. Les deux caractérisations que nous avons mentionnées reposent sur la notion de surface. La deuxième utilise aussi le concept de contexte.

Nous étudions également des problèmes de décision. Il est décidable si un langage engendré par une grammaire context-free est bien formé (c'est-à-dire contenu dans un ensemble des mots Dyck premiers), mais il est indécidable s'il possède une grammaire XML. En revanche, il est décidable si les surfaces d'une grammaire context-free sont finies. Nous terminons notre exposé par quelques remarques historiques. En effet, plusieurs espèces de langages context-free étudiées dans les années soixante sont très liées aux grammaires XML. Il s'agit de grammaires parenthétiques ou de grammaires crochétées. Ce lien est esquissé.

# Two permutation set statistics

Richard Ehrenborg  
KTH

## Abstract

It is a classic result that the distribution of the descent set statistic is maximized on the two alternating sets. Ira Gessel conjectured where the maximum occurs when maximizing this distribution over sets with a fixed number of runs. I will explain the techniques for proving this conjecture. They include the Viennot triangle for computing this distribution and two orderings for comparing sets. Also asymptotic results for this distribution will be presented.

Another statistic is the excedance set statistic. Even though the distributions of the number of descents and of the number of excedances are the same, the distributions of the descent set statistic and the excedance set statistic behave quite differently. I will state what is currently known about the excedance set distribution and end with a few conjectures.

## Résumé

C'est un résultat classique que la distribution de la statistique « ensemble de descentes », sur les permutations, atteint son maximum aux deux ensembles alternés. Ira Gessel a conjecturé la nature du maximum de cette distribution lorsque l'on se restreint aux ensembles ayant un nombre fixe de sous-séquences d'éléments consécutifs. Je vais présenter des techniques permettant de démontrer cette conjecture. Elles comprennent le triangle de Viennot pour le calcul de cette distribution et deux relations d'ordre sur les sous-ensembles. Des résultats asymptotiques pour cette distribution seront également présentés.

Une autre statistique est la distribution des permutations par « ensemble d'excédañces ». Même si les distributions du nombre de descentes et du nombre d'excédañces sont les mêmes, les distributions par ensembles de descentes et par ensembles d'excédañces sont en fait très différentes. Je vais présenter ce qui est connu à ce jour et énoncer quelques conjectures.

# Reconstruction d'un génome dupliqué ancestral

Nadia El-Mabrouk  
Université de Montréal

## Résumé

Cette présentation traite d'un phénomène d'évolution particulier : la duplication complète de génome. Un tel phénomène a été observé chez de nombreuses lignées d'eucaryotes. La co-évolution des gènes dupliqués entraîne alors une évolution rapide de l'espèce. Le génome est représenté par un ensemble de séquences (chromosomes) formés d'une suite de gènes. À la suite du phénomène de duplication, le génome subit une suite de mutations interchromosomales et intrachromosomales qui modifie grandement l'ordre et l'orientation des gènes dans le génome. Au bout du compte, on ne détecte plus que des segments, plus ou moins longs, de chromosomes, apparaissant exactement deux fois dans le génome complet. Le problème est alors de retrouver le génome ancestral juste après la duplication, ainsi que le nombre et le type de mutations surveillées. La méthode de parsimonie consiste à considérer le nombre minimal d'opérations de réarrangement nécessaire pour retrouver un génome dupliqué. La complexité et les méthodes de résolution de ce problème sont étroitement liées au type d'opérations de réarrangement et de données biologiques considérées. Nous présentons des algorithmes exacts et en temps polynomial pour résoudre le problème dans le cas où les gènes sont orientés et signés, et dans le cas où les opérations de réarrangement considérées sont les inversions et les translocations. Notre méthode se base sur les résultats de Hannenhalli et Pevzner pour les remaniements génomiques.

## Abstract

The genome can be modeled as a set of strings (chromosomes) of distinguished elements called genes. Genome doubling is an important source of new gene functions and novel physiological pathways. Originally a duplicated genome contains two identical copies of each chromosome, but through reversals or other intrachromosomal movements, the gene orders in each copy of a chromosome change independently, and through reciprocal translocation, the identical synteny of the two copies is disrupted. Eventually, all that can be detected are several chromosome segments, each of which appears twice in the genome, containing a series of paralogous genes in parallel orders. The problem considered here is to reconstruct some or most of the original gene order at the time of genome duplication, based on traces conserved in the ordering of those duplicate genes still identifiable, i.e., still recognizably homologous despite evolutionary divergence in nucleotide sequence. The complexity of this problem and the choice of methods depend on the type of rearrangement operations considered and the kind of biological data available. We present exact algorithms for reconstructing the ancestral doubled genome in polynomial time, in the case of ordered and signed genes, and when rearrangement operations considered are reversals and translocations. Our method is based on the result of Hannenhalli and Pevzner for the comparison of two given genomes.

# Permutations d'André et actions de groupes

Dominique Foata  
Université de Strasbourg

## Résumé

La notion de permutation d'André a été introduite en 1971 par Foata et Schutzenberger pour l'étude de l'équation différentielle  $D''=D'D''$ , avec les conditions initiales  $D(0)=0$ ,  $D'(0)=s$ ,  $D''(0)=t$ . La solution analytique trouvée avait comme coefficients dans son développement en série de Taylor des polynômes en deux variables, appelés polynômes d'André, qui n'étaient autres que des polynômes génératrices d'une classe de permutations, appelées permutations d'André, qui pour chaque ordre  $n$ , sont énumérées par les nombres tangents ou sécants.

En 1974, Foata et Strehl ont fait apparaître ces permutations d'André comme des systèmes de représentants d'orbites de groupes agissant sur l'ensemble des arbres binaires croissants et en 1998, Hetyei et Reiner les ont obtenues comme représentants d'orbites pour une autre famille de groupes, agissant sur une autre famille d'arbres binaires, que nous appellerons arbres de Hetyei-Reiner.

Il a donc semblé naturel de rassembler ces deux constructions dans un même modèle et d'introduire un surensemble  $M(n)$  d'arbres binaires, dits minimax (la définition est différente de celle qui avait été prise par Hetyei-Reiner) contenant à la fois les arbres croissants et ceux de Hetyei-Reiner, tel que si  $T$  est un arbre minimax, il lui correspond un arbre croissant  $c(T)$  («  $c$  » pour « complément ») et un arbre de Hetyei-Reiner  $r(T)$  («  $r$  » pour « retournement ») ayant la propriété que  $r$  et  $c$  commutent et que  $rc(T)=cr(T)$  est un arbre d'André, qui est en bijection avec une permutation d'André.

On obtient également les récurrences utiles et l'expression de la fonction génératrice de ces arbres minimax par différentes statistiques.

## Abstract

The notion of André permutation was introduced by Foata and Schutzenberger in 1971 in the study of the differential equation  $D''=D'D''$ , with the initial conditions :  $D(0)=0$ ,  $D'(0)=s$ ,  $D''(0)=t$ . The analytic solution of the equation has a Taylor expansion whose coefficients are polynomials in two variables, called André polynomials. Those polynomials are generating polynomials for a class of permutations, the so-called André permutations, that are enumerated by the tangent and secant numbers.

In 1974 Foata and Strehl let those André permutations appear as systems of representatives of orbits for a group acting on the set of increasing binary trees. In 1998 Hetyei and Reiner introduced another family of groups, acting on another family of binary trees and showed that the orbits were still enumerated by the André permutations.

It was then natural to include those two approaches into a common set-up. For this purpose a new class  $M(n)$  of so-called minimax trees (the definition differs from the definition proposed by Hetyei-Reiner) has been introduced that contains both the increasing trees and the trees defined by those two authors. It is shown that to each

minimax tree  $T$  there corresponds first an increasing tree  $c(T)$ , then a Heytei-Reiner tree  $r(T)$  with the property that  $r$  and  $c$  commute and that the tree  $rc(T)=cr(T)$  is an André tree.

The necessary recurrences, as well as the generating functions for minimax trees by various statistics are also derived.

# Recent Developments on the Theory of Diagonal Harmonics: Results and Conjectures

Adriano Garsia  
UCSD

## Abstract

We denote by  $\mathbf{DH}_n$  the space of polynomials  $P(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n)$  which satisfy the differential equations  $\sum_i \partial_{x_i}^h \partial_{y_i}^k P(x; y) = 0$ , for  $h + k \geq 1$ . It is easy to show that  $DH_n$  is finite dimensional and consists of polynomials of total degree  $\leq \binom{n}{2}$ . In J. Algebraic Combin. 5 (1996), no. 3, 191-244 Garsia-Haiman gave an explicit conjectured formula for the bigraded character of  $\mathbf{DH}_n$  under the diagonal action of  $S_n$ . This formula in particular implies that the subspace  $\mathbf{DHA}_n$  consisting of the alternants of  $\mathbf{DH}_n$  has dimensions given by the Catalan number  $\frac{1}{n+1} \binom{2n}{n}$ . Part of this conjecture may have recently been proved by Haiman using algebraic geometrical methods. In any event to this date much remains to be done from the combinatorial and computational point of view. In this paper we report on some conjectures resulting from an intensive computer study of  $\mathbf{DHA}_n$ . Theoretical considerations and experimental data indicate that  $\mathbf{DHA}_n$  is the linear span of the polynomials  $D_1^{p_1} D_2^{p_2} \cdots D_{n-1}^{p_{n-1}} \Delta_n(x)$  where  $D_k = \sum_{i=1}^n y_i \partial_{x_i}^k$  (for  $k \geq 1$ ) and  $\Delta_n(x)$  is the Vandermonde determinant in  $x_1, x_2, \dots, x_n$ . We also present some precise results concerning the ideal  $\mathcal{I}_n$  of polynomials  $P(D_1, D_2, \dots, D_{n-1})$  which “kill”  $\Delta_n(x)$ .

## Résumé

L'espace des polynômes satisfaisant les équations différentielles  $\sum_i \partial_{x_i}^h \partial_{y_i}^k P(x; y) = 0$ , pour  $h + k \geq 1$ , est dénoté  $\mathbf{DH}_n$ . Il est facile de montrer que  $DH_n$  est de dimension finie et ne contient que des polynômes de degré  $\leq \binom{n}{2}$ . Dans J. Algebraic Combin. 5 (1996), no. 3, 191-244 Garsia-Haiman ont formulé une conjecture explicite pour le caractère bigradué de  $\mathbf{DH}_n$  sous l'action diagonale de  $S_n$ . Cette formule implique en particulier que le sous espace  $\mathbf{DHA}_n$  des alternants de  $\mathbf{DH}_n$  a comme dimension le nombre de Catalan  $\frac{1}{n+1} \binom{2n}{n}$ . Une partie de cette conjecture semble avoir été prouvée récemment par Haiman au moyen de méthodes de géométrie algébrique. Quoi qu'il en soit, il reste beaucoup à faire du point de vue de la combinatoire et du calcul formel. Dans ce travail, nous faisons le point sur des conjectures résultant d'une étude intensive par ordinateur de  $\mathbf{DHA}_n$ . Des considérations théoriques de même qu'expérimentales suggèrent que  $\mathbf{DHA}_n$  est le sous espace engendré par les polynômes  $D_1^{p_1} D_2^{p_2} \cdots D_{n-1}^{p_{n-1}} \Delta_n(x)$  où  $D_k = \sum_{i=1}^n y_i \partial_{x_i}^k$  (for  $k \geq 1$ ) et  $\Delta_n(x)$  est le déterminant de Vandermonde en les variables  $x_1, x_2, \dots, x_n$ . Nous présentons aussi des conjectures précises à propos de l'idéal  $\mathcal{I}_n$  des polynômes  $P(D_1, D_2, \dots, D_{n-1})$  qui « tuent »  $\Delta_n(x)$ .

# q-énumération des matrices à signes alternants n'ayant qu'un seul -1

Pierre Lalonde  
LaCIM

## Résumé

Grâce à une suite de récurrences, on énumère les matrices à signes alternants qui comportent exactement un  $-1$ , suivant leur nombre d'inversions. On peut tenir compte de la position de la seule entrée non nulle en première ligne. D'autre part, conformément aux conjectures de Mills, Robbins et Rumsey, on montre que l'énumération (selon le nombre de parts) des partitions planes descendantes n'ayant qu'une seule part spéciale vérifie les mêmes récurrences.

## Abstract

Using a set of recurrence relations, we enumerate the alternating sign matrices that contain exactly one  $-1$  according to their number of inversions. We can take into account the position of the unique non-zero entry in the first row. On the other hand, as suggested by the Mills, Robbins and Rumsey conjectures, we show that the enumeration (according to the number of parts) of descending plane partitions with exactly one special part follows the same relations.

## 1 Matrices à signes alternants

Une matrice carrée  $M = (m_{i,j})_{1 \leq i,j \leq n}$  est dite à *signes alternants* si  $m_{i,j} \in \{1, 0, -1\}$  et si, dans chaque ligne et dans chaque colonne, les entrées non-nulles alternent entre 1 et  $-1$ , en commençant et finissant par un 1.

On remarquera que la première ligne d'une matrice à signes alternants ne comporte qu'une seule entrée non-nulle, qui est nécessairement un 1. On parlera du *premier 1*.

À une matrice à signes alternants  $M = (m_{i,j})$ , on associe certains paramètres (outre l'ordre  $n$ ):

- $r(M)$  est le nombre d'entrées à gauche du premier 1. On a  $0 \leq r(M) \leq n-1$ .
- $s(M)$  désigne le nombre d'entrées égales à  $-1$ .
- $\text{inv}(M) = \sum_{k>i, \ell < j} m_{i,j} m_{k,\ell} = \sum_{i,j} m_{i,j} \left( \sum_{k>i, \ell < j} m_{k,\ell} \right)$  est le nombre d'inversions de  $M$ .

On définit les fonctions génératrices:

$$A_n = A_n(q, t) = \sum_{s \geq 0} A_n^{(s)}(q) = \sum_M q^{\text{inv}(M)} t^{s(M)}$$

(la dernière somme portant sur les matrices à signes alternants  $M$  d'ordre  $n$ ); et

$$A_{n,r} = A_{n,r}(q, t) = \sum_{s \geq 0} A_{n,r}^{(s)}(q) = \sum_M q^{\text{inv}(M)} t^{s(M)}$$

(la dernière somme portant sur les matrices à signes alternants  $M$  d'ordre  $n$  ayant  $r$  entrées à gauche du premier 1). Évidemment  $A_n = \sum_{0 \leq r \leq n-1} A_{n,r}$ .

## 2 Partitions planes descendantes

Nous allons lier les matrices à signes alternants à un autre objet combinatoire, les partitions planes descendantes (voir [MRR]). Comme la méthodologie de Gessel-Viennot (voir [GV]) permet de les encoder par des configurations de chemins qui ne se coupent pas, nous définissons directement les partitions planes descendantes à l'aide de chemins.

Pour les partitions planes descendantes d'ordre  $n$  (i.e. les parts ne dépassant pas  $n$ ), on utilisera des chemins dont les sommets se trouvent sur la grille  $\{0, \dots, n\} \times \{0, \dots, n\}$ . Un pas Est joint un sommet  $(k, \ell)$  à  $(k+1, \ell)$ ; un pas Sud joint un sommet  $(k, \ell)$  à  $(k, \ell-1)$ .

Dans ce contexte, une partition plane descendante d'ordre  $n$  est une configuration de chemins n'ayant aucun sommet commun, de points de départ  $\{(0, i)\}_{i \in X \cup \{n\}}$  et d'arrivée  $\{(j, 0)\}_{j \in \{0\} \cup X}$  (pour un certain  $X \subseteq \{1, 2, \dots, n-1\}$ ). (Voir la figure 1.)

À une partition plane descendante  $P$ , on associe un certain nombre de statistiques, outre son ordre  $n$ . En particulier, on désignera par:

- a.  $r(P)$ : le nombre de pas Est au niveau  $n$ ;
- b.  $s(P)$ : le nombre de pas Est sous la diagonale  $\{(i, i)\}_{0 \leq i \leq n}$  (pas *spéciaux*);
- c.  $\text{inv}(P)$ : le nombre de pas Est.

Ceci conduit à définir des fonctions génératrices associées aux partitions planes descendantes. Ainsi,

$$P_n = P_n(q, t) = \sum_{s \geq 0} P_n^{(s)}(q) = \sum_P q^{\text{inv}(P)} t^{s(P)},$$

(la dernière somme portant sur les partitions planes descendantes  $P$  d'ordre  $n$ ); et

$$P_{n,r} = P_{n,r}(q, t) = \sum_{s \geq 0} P_{n,r}^{(s)}(q) = \sum_P q^{\text{inv}(P)} t^{s(P)},$$

(la dernière somme portant sur les partitions planes descendantes  $P$  d'ordre  $n$  ayant  $r$  pas Est au niveau  $n$ ).

Autrement dit, chaque pas Sud est de poids 1 et chaque pas Est de poids  $q$  ou  $qt$  selon qu'il se trouve en haut ou en bas de la diagonale. (Le poids d'une configuration étant le produit des poids des pas qui la composent.) On a bien sûr:  $P_n = \sum_{0 \leq r \leq n-1} P_{n,r}$ .

On peut traduire certaines conjectures de Mills, Robbins et Rumsey [MRR] en terme de ces fonctions génératrices:

- a.  $A_n(1, 1) = P_n(1, 1) = \prod_{0 \leq j \leq n-1} \frac{(3j+1)!}{(n+j)!}$ . (Démontrée par Zeilberger [Ze1]; simplification ultérieure par Kuperberg [Ku].)
- b.  $A_{n,r}(1, 1) = P_{n,r}(1, 1) = \binom{n+r-1}{r} \frac{(2n-r-2)!}{(n-r-1)!} \prod_{0 \leq j \leq n-1} \frac{(3j+1)!}{(n+j)!}$ . (Démontrée par Zeilberger [Ze2].)

$$\text{c. } A_{n,r}(q,t) = P_{n,r}(q,t).$$

Nous allons montrer ici que  $A_{n,r}^{(1)}(q,t) = P_{n,r}^{(1)}(q,t)$ .

### 3 Déterminants associés aux partitions planes descendantes

Diverses modifications conduisent à un deuxième modèle graphique pour les partitions planes descendantes. On utilisera cette fois la demi-grille supérieure:  $\{(k,\ell)\}_{0 \leq \ell \leq k \leq n}$  sur laquelle on définit deux types de chemins:

- a. Les chemins de type  $T$ , constitués (comme précédemment) de pas Est (poids  $q$ ) et Sud (poids 1).
- b. Les chemins de type  $B$ , constitués cette fois de pas Est (poids 1) et Nord-Est (poids  $qt$ ). Le dernier pas doit être Est.

Une *configuration TB* est une suite de chemins  $\Omega = (\omega_1, \dots, \omega_n)$  telle que:

- a. Chaque chemin  $\omega_i$  est de type T ou de type B.
- b. Il y a une permutation  $\sigma$  telle que  $\omega_i$  part de  $(0, i)$  et se termine en  $(\sigma(i)-1, \sigma(i)-1)$ .
- c. Si deux chemins ont un sommet commun, ils sont de types différents.  
(Voir la figure 2.)

**Lemme 3.1** *Les partitions planes descendantes sont en bijection avec les configurations TB. Cette bijection préserve les poids (dans  $\mathbb{Z}[q, t]$ ), l'ordre  $n$  et le nombre de pas Est au niveau  $n$ .*

Soit  $t_{i,j}$  la somme des poids des chemins de type T de  $(0, i)$  à  $(j, 0)$  et  $b_{i,j}(t)$  la somme équivalente pour les chemins de type B. Pour une série formelle  $F(x, y)$  à coefficients dans  $\mathbb{Z}[q, t]$ , l'expression  $[x^i y^j] F(x, y)$  désignera le coefficient de  $x^i y^j$  dans  $F(x, y)$ .

**Théorème 3.1** *On a, pour  $1 \leq i \leq n$  et  $0 \leq j \leq n-1$ :*

1.  $t_{i,j} = [x^i y^j] (1 - x - q x y)^{-1}$ .
  2. *Pour tenir compte du nombre  $r$  de pas Est au niveau  $n$ , il faut remplacer  $t_{n,j}$  par  $[x^n y^j] (q x y)^r x (1 - x - q x y)^{-1}$ .*
  3.  $b_{i,j}(t) = [x^i y^j] (1 - q t y) (1 - x - q x y)^{-1}$ .
- La fonction génératrice  $P_n$  est donnée par  $P_n = \det(t_{i,j} - b_{i,j}(-t))$ , où  $1 \leq i \leq n$  et  $0 \leq j \leq n-1$ . On a la même relation pour  $P_{n,r}$  en changeant  $t_{n,j}$  comme indiqué en 2.*

(Les signes négatifs permettent de compenser les signes produits par le déterminant.)

**Théorème 3.2** *Soit  $m_{i,j} = [x^{i-1} y^{j-1}] (1 - x - q x y)^{-1} (1 - x y + q t y)^{-1}$  (pour  $1 \leq i, j \leq n$ ), alors*

$$P_n = \det(m_{i,j})_{1 \leq i, j \leq n}.$$

*Pour obtenir  $P_{n,r}$ , on remplace  $m_{n,j}$  par  $[x^{n-1} y^{j-1}] (q x y)^r (1 - x - q x y)^{-1} (1 - x y + q t y)^{-1}$ .*

## 4 Le dernier modèle

Le théorème précédent suggère un troisième (et dernier) modèle pour les partitions planes descendantes.

On utilisera la demi-grille triangulaire:  $\{(k, \ell)\}_{0 \leq \ell < k \leq n}$ . Sur celle-ci, on considère les chemins qui se factorisent en deux parties consécutives (chemins *mixtes*). La première est constituée de pas Sud (de poids 1) et Est (de poids  $q$ ) et provient du facteur  $(1 - x - qxy)^{-1}$ . La suivante, de pas Est (de poids 1) et Nord-Est (de poids  $-qt$ ), provient du facteur  $(1 - xy + qt y)^{-1}$ . Le nombre de pas Est au niveau  $n$  de la première partie peut être fixé.

Une *configuration mixte* d'ordre  $n$  est une suite de chemins mixtes  $\Omega = (\omega_1, \dots, \omega_n)$  telle que:

- Il y a une permutation  $\sigma$  telle que  $\omega_i$  part de  $(0, i)$  et termine en  $(\sigma(i) - 1, \sigma(i))$ .
- La sous-configuration obtenue en effaçant les secondes parties des chemins est sans intersections.
- La sous-configuration obtenue en effaçant les premières parties des chemins est sans intersections.

(Voir la figure 3.) On posera encore  $\text{sgn}(\Omega) = \text{sgn}(\sigma)$  (le signe de  $\Omega$ ).

**Corollaire 4.1**  $P_n$  est la somme des poids signés des configurations mixtes d'ordre  $n$ . (De même pour  $P_{n,r}$  où le nombre de pas Est de la première partie au niveau  $n$  est fixé à  $r$ .)

## 5 Les coefficients de $t$

Nous allons maintenant nous concentrer sur les coefficients  $P_{n,r}^{(1)}$  (et  $A_{n,r}^{(1)}$ ). Dans ce cas, tous les chemins d'une configuration mixte sont horizontaux sauf deux chemins consécutifs qui se croisent. Le signe de telles configurations est toujours positif.

Soit  $X_{n,r}$  le poids des configurations mixtes pour lesquelles la paire de chemins qui se croisent commence au niveau  $n$  et comporte  $r$  pas Est au niveau  $n$  dans la première partie. (On utilisera les notations  $[k] = 1 + q + q^2 + \dots + q^{k-1}$  et  $[k]! = [1][2] \cdots [k]$ .)

**Théorème 5.1** *On a:*

$$1. P_{n,r}^{(1)} = q^r P_{n-1}^{(1)} + X_{n,r}.$$

$$2. X_{n,r} = \frac{q^{r+1}}{1-q} [n-2]! \begin{vmatrix} n-1 & [n-1] \\ n-1-r & [n-1-r] \end{vmatrix}.$$

$$3. X_n = \sum_r X_{n,r} = \frac{q^2}{(1-q)^2} [n-2]! ([n-1]^2 - (n-1)^2 q^{n-2}).$$

Avec  $P_1^{(1)} = 0$ , (et la relation  $P_n = \sum_{0 \leq r \leq n-1} P_{n,r}$ ) ces résultats déterminent  $P_{n,r}$ .

**Théorème 5.2**  $X_{n,r}$  est déterminé par la récurrence:

$$X_{n,r} = q[r] X_{n-1,r-1} + q^r [n-1-r] X_{n-1,r} + q^{r+1} [r] [n-1-r] [n-2]!$$

et par les valeurs initiales:

$$X_{n,0} = 0 \quad \text{et} \quad X_{n,n-1} = 0.$$

Passons aux matrices à signes alternants. Soit  $Y_{n,r} = \sum_M q^{\text{inv}(M)} t^{s(M)}$ , la somme portant sur les matrices à signes alternants  $M$  d'ordre  $n$  ayant  $r$  entrées à gauche du

premier 1, avec un seul  $-1$ , celui-ci se trouvant dans la même colonne que le premier 1.  
Soit  $Y_n = \sum_r Y_{n,r}$ .

**Théorème 5.3** *On a:*

1.  $A_{n,r}^{(1)} = q^r A_{n-1}^{(1)} + Y_{n,r}$ .
2.  $A_n^{(1)} = [n] A_{n-1}^{(1)} + Y_n$ .
3.  $Y_{n,r} = q[r] Y_{n-1,r-1} + q^r [n-1-r] Y_{n-1,r} + q^{r+1} [r][n-1-r][n-2]!$ .
4.  $A_1^{(1)} = 0$  et  $Y_{n,0} = Y_{n,n-1} = 0$ .

**Corollaire 5.1** *On a*

$$X_{n,r} = Y_{n,r} \quad \text{et} \quad A_{n,r}^{(1)} = P_{n,r}^{(1)}.$$

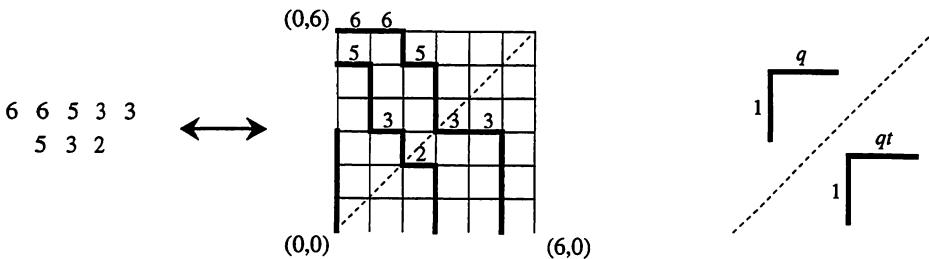


fig. 1. Une partition plane descendante (à gauche) avec le codage graphique (au centre). Ici,  $n = 6$ ,  $r = 2$ ,  $s = 3$  et  $\text{inv} = 8\dots$  Les poids des pas associés.

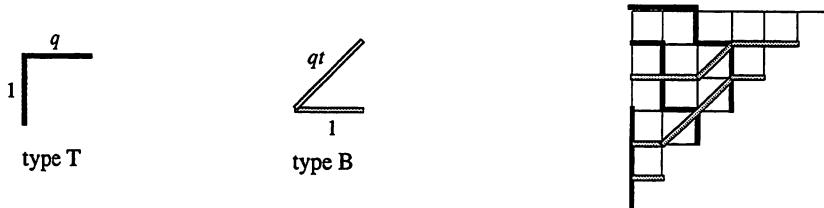


fig. 2. Poids des pas. Une configuration TB (correspondant à la partition plane descendante de la figure 1).

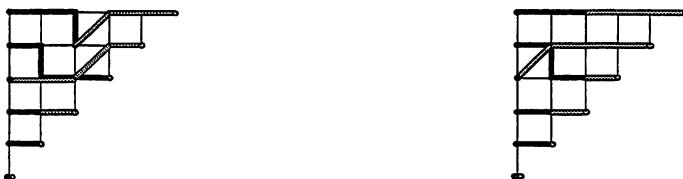


fig. 3. Une configuration mixte... Une configuration mixte avec exactement un pas Nord-Est.

## Bibliographie

- [GV] Gessel I., Viennot X. *Binomial determinants, paths and hook length formulae.* Adv. in Math., 58, (1985), 300–321.
- [Ku] Kuperberg, G., *Another proof of the alternating sign matrix conjecture.* International Mathematics Research Notes, (1996), 139–150.
- [MRR] Mills W.H., Robbins D.P., Rumsey H. *Alternating sign matrices and descending plane partitions.* J. Comb. Th.(A), 34, (1983), 340–359.
- [St] Stanley R. P. *A baker’s dozen of conjectures concerning planes partitions.* in Combinatoire Énumérative, G. Labelle, P. Leroux, eds., Lecture Notes in Mathematics, 1234, Springer-Verlag (1986) 285–293.
- [Zel1] Zeilberger, D. *Proof of the alternating sign matrix conjecture.* Electronic Journal of Combinatorics, 3, (1996), R13.
- [Zel2] Zeilberger, D. *Proof of the refined alternating sign matrix conjecture.* New York Journal of Mathematics, 2, (1996), 59–68.

# Sous-ensembles rationnels de $\mathbf{Z}^k$ , théorème de Popoviciu et valuation des polyèdres

Srećko Brlek

LACIM, Département de Mathématiques et Informatique, UQAM  
Montréal, P.Q. H3C 3P8, Canada

Christophe Reutenauer

UFR Mathématiques, Université Louis Pasteur  
7, rue René Descartes, 67084 Strasbourg, France

June 1, 2000

## Abstract

We associate to each rational subset on a free abelian group, and more generally to each (generalized) rational series on this group, a multivariate rational function. The kernel of this mapping is precisely described, thereby extending Popoviciu's theorem. Another application is a valuation of rational polyhedra.

## Résumé

Nous associons à toute partie rationnelle d'un groupe abélien libre, et plus généralement à toute série rationnelle (en un sens généralisé) sur ce groupe, une fraction rationnelle en plusieurs variables. Le noyau de cette fonction est caractérisé, ce qui étend le théorème de Popoviciu. Une autre application est une valuation des polyèdres rationnels.

Un sous-ensemble  $S$  d'un monoïde  $M$  est *rationnel* s'il est obtenu à partir des sous-ensembles finis par les opérations rationnelles, à savoir l'union, le produit et l'étoile (= sous-monoïde engendré, noté  $*$ ). La classe des sous-ensembles *rationnels non-ambigus* de  $M$  est la plus petite classe  $\mathcal{E}$  de sous-ensembles de  $M$  contenant les sous-ensembles finis et telle que [5]:

- si  $S_1$  et  $S_2 \in \mathcal{E}$  et  $S_1 \cap S_2 = \emptyset$ , alors  $S_1 \cup S_2 \in \mathcal{E}$ ;
- si  $S_1$  et  $S_2 \in \mathcal{E}$  et si leur produit est non-ambigu (i.e  $x_1x_2 = y_1y_2$  avec  $x_i, y_i \in S_i$  implique  $x_i = y_i$ ), alors  $S_1S_2 \in \mathcal{E}$ ;
- si  $S \in \mathcal{E}$  et si  $S$  est une base d'un sous-monoïde libre de  $M$  alors  $S^* \in \mathcal{E}$ .

Par définition tout sous-ensemble rationnel de  $M$  est décrit par une *expression rationnelle*, qui est une formule bien formée qui utilise les éléments de  $M$  et les opérations rationnelles. De manière analogue un sous-ensemble rationnel non-ambigu possède une expression rationnelle non-ambigüe, qui est une expression rationnelle "sans multiplicités". Tout sous-ensemble rationnel non-ambigu de  $M$  est clairement rationnel . De plus, si  $M$  est un monoïde libre, alors tout sous-ensemble rationnel est aussi non-ambigu (conséquence directe du théorème de Kleene [4]). L'hypothèse que  $M$  soit libre est essentielle ([5], p. 174); néanmoins, Eilenberg et Schützenberger ont montré que dans le

cas où  $M$  est commutatif, tout sous-ensemble rationnel de  $M$  est aussi non-ambigu ([5] Th. IV).

Nous considérons le cas  $M = \mathbf{Z}^k$  où le théorème d'Eilenberg et Schützenberger est valide. Ceci nous permet d'associer une fonction rationnelle à chaque sous-ensemble rationnel de  $\mathbf{Z}^k$  de la manière suivante. Soit  $\mathbf{F}$  un corps et  $x_1, \dots, x_k$  des variables commutatives. On identifie  $\mathbf{Z}^k$  avec le groupe multiplicatif des monômes de Laurent en les variables  $x_1, \dots, x_k$ :

$$\mathbf{Z}^k = \{x_1^{i_1} x_2^{i_2} \dots x_k^{i_k} \mid i_j \in \mathbf{Z}\}$$

Maintenant, à chaque expression rationnelle  $E$  sur  $M$ , on associe l'expression  $f(x) = f(x_1, \dots, x_k)$  dans le corps  $\mathbf{F}(x_1, \dots, x_k)$  en remplaçant  $\cup$  par  $+$ ,  $S^*$  par  $(1 - S)^{-1}$  et le produit par le produit.

Par exemple l'expression rationnelle  $x^* \cup yz$  est remplacée par  $(1 - x)^{-1} + yz$  et l'expression  $x^*(y^{-1})^* \cup x^{-1}(x^{-1})^*$  par  $(1 - x)^{-1}(1 - y^{-1})^{-1} + x^{-1}(1 - x^{-1})^{-1}$ .

**Théorème 1** *Si  $E$  est une expression rationnelle non-ambigüe, alors  $f(x)$  est un élément bien défini de  $\mathbf{F}(x_1, \dots, x_k)$  qui dépend uniquement du sous-ensemble  $S$  de  $\mathbf{Z}^k$  représenté par  $E$ .*

Le fait que  $f$  soit bien définie n'est pas trop difficile à montrer car l'étoile non-ambigüe s'applique uniquement aux singletons, dans le cas commutatif. Nous établissons un résultat plus général qui, en contrepartie, implique l'unicité. Dans le cas où  $S$  est un sous-ensemble rationnel de  $\mathbf{N}^k$ , la fonction  $f$  n'est autre que la fonction représentée par la série formelle  $\sum_{m \in S} m$ , laquelle est rationnelle, comme il découle directement du théorème d'Eilenberg et Schützenberger.

L'application  $S \mapsto f$  est loin d'être injective comme on peut le voir dans l'exemple suivant:  $S = \{x^n \mid n \in \mathbf{Z}\}$  possède l'expression non-ambigüe  $x^* \cup x^{-1}(x^{-1})^*$ . Alors

$$f(x) = \frac{1}{1-x} + \frac{x^{-1}}{1-x^{-1}} = \frac{1}{1-x} + \frac{1}{x-1} = 0.$$

Nous considérons maintenant l'espace vectoriel  $V$  des séries formelles  $\sum_{m \in M} \alpha_m m$ , à coefficients  $\alpha_m \in \mathbf{F}$ . Nous définissons sur  $V$  les opérations (partiellement définies) produit et étoile comme suit. Soit  $S = \sum_{m \in M} \alpha_m m$ , et  $T = \sum_{m \in M} \beta_m m$ . Alors, les opérations

$$\begin{aligned} ST &= \sum_{m \in M} \gamma_m m, \quad \gamma_m = \sum_{xy=m} \alpha_x \beta_y; \\ S^* &= \sum_{m \in M} \delta_m, \quad \delta_m = \sum_{i \geq 0, x_1 \dots x_i = m} \alpha_{x_1} \dots \alpha_{x_i}; \end{aligned}$$

sont *définies* (ou *convergentes*) lorsque chaque coefficient  $\gamma_m$  et  $\delta_m$  est obtenu comme somme finie.

Le sous-espace vectoriel  $\mathcal{S}$  des *séries rationnelles* (en un sens généralisé) est le sous-espace contenant les polynômes (séries finies) et vérifiant

- si  $S, T \in \mathcal{S}$  et si le produit  $ST$  est défini alors  $ST \in \mathcal{S}$ ;
- si  $S \in \mathcal{S}$  et si  $S^*$  est défini alors  $S^* \in \mathcal{S}$ .

Les séries rationnelles possèdent une caractérisation matricielle. En effet, on considère les matrices carrées sur l'algèbre  $\mathbf{F}[M]$  où les opérations produit et étoile sont définies de manière similaire:  $A^* = \sum_{n \in \mathbf{N}} A^n$  est défini si pour tout  $m \in M$  il n'y a qu'un

nombre fini de puissances  $n \in \mathbf{N}$  telles que  $m$  apparaisse dans  $A^n$  avec un coefficient non-nul.

Le résultat suivant est bien connu des spécialistes (cf. [8]).

**Théorème 2** *Une série  $S$  est rationnelle si et seulement s'il existe une matrice  $A$ , une matrice ligne  $\lambda$ , et une matrice colonne  $\gamma$  sur  $\mathbf{F}$  telles que  $A^*$  soit défini et que  $S = \lambda A^* \gamma$ .*

Finalement, toute série rationnelle  $S$  possède une expression rationnelle  $E$  à laquelle on associe l'expression  $f(x_1, \dots, x_k)$  dans le corps  $\mathbf{F}(x) = \mathbf{F}(x_1, \dots, x_k)$  en remplaçant les opérations rationnelles comme précédemment. On obtient alors le résultat suivant qui implique le théorème 1.

**Théorème 3** *La fonction  $f(x_1, \dots, x_k)$  est un élément bien défini de  $\mathbf{F}(x)$  ne dépendant que de  $S$ . Si  $S$  est représentée par  $(\lambda, A, \gamma)$  alors  $1 - A$  (qui est une matrice de polynômes de Laurent) est inversible sur  $\mathbf{F}(x)$  et  $f(x_1, \dots, x_k) = \lambda(1 - A)^{-1} \gamma$ .*

On remarque que si  $A$  est une partie rationnelle de  $\mathbf{Z}^k$ , alors sa série caractéristique  $\sum_{m \in A} m$  est rationnelle; le théorème 3 implique donc le théorème 1.

Si l'expression rationnelle pour  $S$  n'implique que des éléments de  $\mathbf{F}[[x]]$ , alors  $S$  est rationnelle au sens classique dans  $\mathbf{F}[[x_1, \dots, x_k]]$ , et alors  $f(x)$  est la fonction rationnelle représentée par cette série.

Le noyau de l'application  $\Phi : S \mapsto f$  est complètement décrit par le résultat suivant.

**Théorème 4**  *$S \in \ker(\Phi)$  si et seulement si  $PS = 0$  pour un certain polynôme de Laurent  $P$  non nul.*

Ce résultat implique le théorème de Popoviciu ([9], Th. 4.2.3).

**Corollaire 5** *Soit  $d \in \mathbf{N}$  et  $(a_n)_{n \in \mathbf{Z}}$  une suite satisfaisant une récurrence linéaire propre: pour tout  $n$  dans  $\mathbf{Z}$ ,*

$$a_{n+d} = \alpha_1 a_{n+d-1} + \alpha_2 a_{n+d-2} + \dots + \alpha_d a_n$$

où  $\alpha_i \in \mathbf{F}$  et  $\alpha_d \neq 0$ . Alors

$$\sum_{n < 0} a_n x^n \quad \text{et} \quad \sum_{n \geq 0} a_n x^n$$

sont des fonctions rationnelles en  $x$  et  $x^{-1}$  respectivement, et leur somme dans  $\mathbf{F}(x)$  est nulle.

Un exemple typique de cette situation est l'exemple présenté après le théorème 1.

Un polyèdre dans  $\mathbf{R}^k$  est l'intersection d'un nombre fini de demi-espaces fermés. Il est *rationnel* si ces demi-espaces sont définis par des coordonnées rationnelles. L'*algèbre des polyèdres*  $\mathcal{P}$  est le  $\mathbf{R}$ -espace vectoriel engendré par les fonctions caractéristiques des polyèdres rationnels. Une *valuation* des polyèdres est une application linéaire définie sur  $\mathcal{P}$ .

Nous utilisons les résultats précédents pour construire une certaine valuation sur les polyèdres. A chaque polyèdre rationnel  $P$  nous associons la série  $S_P = \sum_{m \in P \cap \mathbf{Z}^k} m$ , en identifiant monômes de Laurent et vecteurs dans  $\mathbf{Z}^k$ . Comme les ensembles rationnels de  $\mathbf{Z}^k$  sont fermés sous l'intersection et la projection ([5]), et comme un polyèdre rationnel peut être décrit par des formes linéaires ayant des coordonnées entières,  $P \cap \mathbf{Z}^k$  est une partie rationnelle de  $\mathbf{Z}^k$ . Donc  $S_P \in \mathcal{S}$ . La fonction  $P \mapsto S_P$  s'étend naturellement en une application linéaire  $\mathcal{P} \rightarrow \mathcal{S}$ . Nous obtenons ainsi une valuation  $\mathfrak{V} : \mathcal{P} \rightarrow \mathbf{R}(x_1, \dots, x_k)$  définie par  $P \mapsto \Phi(S_P)$ .

**Théorème 6** Notons  $[P]$  la fonction caractéristique d'un polyèdre  $P$ .

- (1) Si  $P_1, \dots, P_n$  sont des polyèdres rationnels et si  $\alpha_1, \dots, \alpha_n$  sont des coefficients réels tels que  $\alpha_1[P_1] + \dots + \alpha_n[P_n] = 0$ , alors

$$\alpha_1\mathfrak{V}[P_1] + \dots + \alpha_n\mathfrak{V}[P_n] = 0.$$

- (2) Si  $P'$  est le translaté de  $P$  par  $v \in \mathbf{Z}^k$ , alors

$$\mathfrak{V}[P'] = x^v\mathfrak{V}[P],$$

où  $x^v = x^{i_1} \dots x^{i_k}$  si  $v = (i_1, \dots, i_k)$ .

- (3) Si  $P$  contient un sous-espace de dimension 1 dans  $\mathbf{R}^k$ , alors  $\mathfrak{V}[P] = 0$ .

Nous conjecturons que la valuation ci-dessus est la même que la valuation  $\mathfrak{F}$  de J. Lawrence [7] et A. Khovanskii, A. Pukhlikov [6] (voir aussi [1], Th. 3.1). On montre aussi dans [1] que  $\mathfrak{F}$  a la propriété suivante: si pour  $(x_1, \dots, x_k) \in \mathbf{C}^k$ , la série  $\sum_{v \in P} x^v$  converge absolument, alors sa limite est égale à  $\mathfrak{F}([P])(x_1, \dots, x_k)$ .

Si  $\mathbf{F}$  est un corps avec une valeur absolue, complet, ce qui précède montre que la conjecture suivante est certainement vraie: si  $S = \sum_{v \in \mathbf{Z}^k} \alpha_v x^v$  converge absolument pour un  $k$ -uplet  $(x_1, \dots, x_k) \in \mathbf{F}^k$ , alors sa limite est égale à  $\mathfrak{V}(S)(x_1, \dots, x_k)$ .

On peut imaginer une généralisation non commutative des résultats ci-dessus. Par exemple, en remplaçant le groupe abélien libre  $\mathbf{Z}^k$  par le groupe libre  $G$  et le corps  $\mathbf{F}(x_1, \dots, x_k)$  par le *corps libre* [3]. Dans [2] nous donnons le contre-exemple suivant, qui montre que cette généralisation n'est pas possible:  $G$  est une partie rationnelle non-ambiguë de lui-même, comme il est bien connu en théorie des automates; nous en donnons une expression rationnelle non-ambiguë, laquelle peut être interprétée dans le corps libre, et nous montrons qu'elle n'y est pas définie, c'est-à-dire qu'on inverse 0 à un moment donné.

## References

- [1] A. Barvinok and J. E. Pommersheim, An Algorithmic Theory of Lattice Points in Polyhedra, *Preprint*, 1999.
- [2] S. Brlek and C. Reutenauer, On a valuation of rational subsets of  $\mathbf{Z}^k$ , à paraître.
- [3] P.M. Cohn, *Skew Fields*, Cambridge University Press, 1996.
- [4] S. Eilenberg, *Automata, Languages and Machines*, Academic Press, San Diego, Cal. 1974.
- [5] S. Eilenberg and M.P. Schützenberger, Rational Sets in Commutative monoids, *J. Algebra*, **13**, 173–101, 1969.
- [6] A. G. Khovanskii and A.V. Pukhlikov, The Riemann-Roch theorem for integrals and sums of quasipolynomials on virtual polytopes, (Russian) *Algebra i Analiz* **4**, no. 4, 188–21 (1992) translation in *St. Petersburg Math. J.* **4**, no. 4, 789–812, 1993.
- [7] J. Lawrence, Rational-function-valued valuations on polyhedra, in: *Discrete and Computational Geometry (New Brunswick, NJ, 1989/1990)*, 199–208, DIMACS Ser. Discrete Mathematics and Theoretical Computer Science, **6**, Amer. Math. Soc., Providence, RI, 1991.

- [8] J. Sakarovitch, Kleene's theorem revisited. *Trends, techniques, and problems in theoretical computer science (Smolenice, 1986)*, 39–50, Lecture Notes in Comput. Sci., **281**, Springer, Berlin-New York, 1987.
- [9] R. P. Stanley, *Enumerative Combinatorics, Vol I*, Wadsworth & Brooks/Cole, Monterey, Cal., 1986.

Les auteurs remercient Ira Gessel pour les intéressantes discussions sur le théorème de Popoviciu et les théorèmes de dualité en combinatoire, lesquelles ont motivé ce travail.

# Flow Polytopes

Richard Stanley  
MIT

## Abstract

If  $G$  is an acyclic directed graph with source vertex  $s$  and sink vertex  $t$ , then the flow polytope  $P(G)$  of  $G$  is the convex polytope of all nonnegative real flows of size 1 from  $s$  to  $t$  through the edges of  $G$ . We will give a survey of properties of this polytope, including the computation of its vertices, volume, Ehrhart polynomial, an explicit triangulation, etc., and its connection with Kostant's partition function and the Chan-Robbins conjecture. This lecture is based on joint work with Alexander Postnikov.

## Résumé

Si  $G$  est un graphe orienté acyclique avec des sommets source  $s$  et puits  $t$ , alors le polytope  $P(G)$  de flots de  $G$  est le polytope convexe de tous les flots réels non-négatifs de valeur 1, de source  $s$  et de puits  $t$ , sur les arcs de  $G$ . Nous présentons les principales propriétés de ce polytope, incluant le calcul des sommets, du volume, du polynôme d'Ehrhart, une triangulation explicite, etc., ainsi que ses rapports avec la fonction de partition de Kostant et avec la conjecture de Chan-Robbins. Cet exposé est basé sur un travail conjoint avec Alexandre Postnikov.

# La beauté des mathématiques combinatoires

Xavier G. Viennot  
CNRS et LaBRI  
Université Bordeaux I

## Résumé

La combinatoire connaît actuellement une renaissance remarquable et est devenue un domaine très actif de la recherche contemporaine avec des interactions fructueuses avec d'autres parties des mathématiques ainsi que l'informatique et la physique théorique. Je donnerai un « pot-pourri » de quelques petites merveilles de cette combinatoire, appelée aussi combinatoire énumérative, algébrique ou encore bijective. Ces morceaux choisis, allant de l'algèbre à l'analyse classique, passant par la physique, sont accessibles à un large public. Une certaine beauté se dégage à partir de formules simples et profondes associant calculs et constructions combinatoires. Quelques coïncidences surprenantes peuvent parfois faire penser à une combinatoire magique.

## Abstract

Combinatorics is experiencing remarkable growth as a research area, with fruitful interactions with other parts of mathematics, theoretical physics and computer science. I will present a potpourri of selected marvels in the area called enumerative, algebraic or bijective combinatorics. These selected pieces, ranging from algebra to classical analysis and physics, are accessible to a large public. A certain beauty emerges from simple and deep formulas associated with combinatorial computations and constructions. Some surprising coincidences may suggest the existence of a combinatorial magic.

# $4^2$ proofs of $n^{n-2}$

Doron Zeilberger  
Temple University

## **Abstract**

Cayley, Borchardt, Sylvester, Prüfer, Göbel, Foata-Fuchs, Moon, T.L. Hill, JOYAL,  
G. Labelle, ..., Ekhad, Majewicz.

## **Résumé**

Cayley, Borchardt, Sylvester, Prüfer, Göbel, Foata-Fuchs, Moon, T.L. Hill, JOYAL,  
G. Labelle, ..., Ekhad, Majewicz.

# Exemples de suites de complexité inférieure à $2n$

Ali ABERKANE

Institut de Mathématiques de Luminy, Marseille  
 aberkane@iml.univ-mrs.fr

## Abstract

In 1940, MORSE and HEDLUND studied Sturmian words (words of complexity exactly  $n + 1$ ). In 1994, G. ROTE presented a study on the words of complexity  $2n$ . We introduce here a class of words of complexity between  $n + 1$  and  $2n$ , and we prove that if the complexity is lower than  $\frac{4}{3}n + 1$ , then  $s(n) = p(n + 1) - p(n) \leq 2$  and that we can always build words of complexity very near to  $n + 1$ .

## Résumé

En 1940, MORSE et HEDLUND ont étudié les suites sturmianes (les suites de complexité exactement  $n + 1$ ). En 1994, G. ROTE a présenté une étude sur les suites de complexité  $2n$ . Nous introduisons ici une classe de suites de complexité comprise entre  $n + 1$  et  $2n$ , et nous démontrons que si la complexité est inférieure à  $\frac{4}{3}n + 1$  alors  $s(n) = p(n + 1) - p(n) \leq 2$  et qu'on peut toujours construire des suites de complexité très proche de  $n + 1$ .

## 1 Introduction

Dans ce travail, on va étudier l'existence d'une classe de suites infinies de complexité inférieure à  $2n$  et essayer de donner quelques exemples sur ce genre de suites. Dans un premier temps, on s'intéresse à regarder l'évolution des graphes de RAUZY des suites notées  $u_s$  (voir la notation dans le chapitre de définitions) et à calculer le nombre de graphes consécutifs ayant plus d'un facteur spécial à droite. On démontre ainsi que si  $u$  est une suite de type  $u_s$  avec  $a \leq \frac{4}{3}$  alors ces derniers graphes auront une évolution presque similaire et on déduit que  $s(n) = p(n + 1) - p(n) \leq 2$  pour tout  $n \in \mathbb{N}$ . Dans un deuxième temps, on démontre qu'on peut construire des suites de très faible complexité, en donnant une preuve constructive où on calcule la borne inférieure et supérieure de  $\frac{p(n)}{n}$  ( $p(n)$  est la complexité de ces suites), et on donne un exemple de point fixe de substitutions de type  $u_s$ .

## 2 Définitions et notations

### 2.1 Définitions

1. Une suite finie sur un alphabet  $\mathcal{A}$  est appelée un **mot**.
2. La **longueur** d'un mot est le nombre de lettres qui composent ce mot.
3. Un mot  $w$  de longueur  $n$  est un facteur d'une suite  $u$  si et seulement s'il existe  $n_0 \in \mathbb{N}$  tel que  $w = u(n_0)u(n_0 + 1)\cdots u(n_0 + n - 1)$ .

4. Soit  $u$  une suite. On note  $\mathcal{L}_n(u)$  l'ensemble des facteurs de longueur  $n$  de la suite  $u$  et  $p_u(n)$  le cardinal de  $\mathcal{L}_n(u)$ .

La fonction définie de la façon suivante :

$$\begin{array}{ccc} \mathbb{N} & \longrightarrow & \mathbb{N} \\ n & \longmapsto & p_u(n) \end{array}$$

est appelée fonction de **complexité** de la suite  $u$ .

5. Soit  $u$  une suite sur un alphabet  $\mathcal{A}$ .

Son graphe de RAUZY d'ordre  $n$  ou graphe des mots de longueur  $n$ , noté  $\Gamma_n$ , est le graphe orienté tel que :

- Ses sommets sont les mots de longueur  $n$  de  $u$ .
- Il existe un arc du mot  $w$  vers le mot  $v$  s'il existe  $a$  et  $b$ , éléments de  $\mathcal{A}$  vérifiant  $wa = bv$  et  $wa$  facteur de longueur  $n + 1$  de  $u$ .

On appelle  $a$  l'**étiquette** de cet arc.

6. Soit  $u$  une suite sur un alphabet  $\mathcal{A}$  et  $\Gamma_n$  son graphe de RAUZY d'ordre  $n$ .

Soit  $\mathcal{B} = x_1x_2 \dots x_n, x_2 \dots x_n a_1, \dots, x_{k+1} \dots x_n a_1 \dots a_k$  une branche orientée du graphe  $\Gamma_n$ .

**Le mot étiquetant** de la branche  $\mathcal{B}$  est le mot  $a_1a_2 \dots a_k$ .

**La longueur** de la branche  $\mathcal{B}$  est  $|a_1a_2 \dots a_k| = k$ .

7. Soit  $u$  une suite sur un alphabet  $\mathcal{A}$ ,  $w$  un facteur de  $u$  et  $x$  une lettre de  $\mathcal{A}$ .  
 $x$  est une **extension gauche** de  $w$  si  $xw$  appartient à  $\mathcal{L}(u)$ .

$w$  est un facteur **spécial à gauche** s'il accepte plusieurs extensions gauches.

$x$  est une **extension droite** de  $w$  si  $wx$  appartient à  $\mathcal{L}(u)$ .

$w$  est un facteur **spécial à droite** s'il accepte plusieurs extensions droites.

$w$  est un facteur **bispécial** s'il est spécial à la fois à gauche et à droite.

8. Un graphe de mots est dit **sturmien** s'il contient un seul facteur spécial à droite et un seul facteur spécial à gauche.

## 2.2 Notations

1. On note  $G_s(x, y)$  le graphe ci-dessous où  $x$  et  $y$  sont les mots étiquetant les deux boucles. On suppose que  $|x| \geq |y|$ .

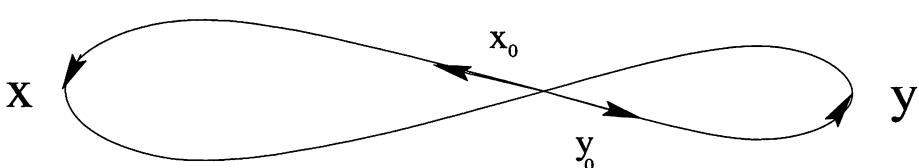


FIG. 1 – Le graphe  $G_s(x, y)$

Le mot  $x$  (resp.  $y$ ) est le mot étiquetant la boucle commençant par l'étiquette  $x_0$  (resp.  $y_0$ ).

On remarque qu'il y a  $|xy| - 1$  sommets,  $|xy|$  arêtes et un unique facteur bispécial.

2. Appelons suite de type  $u_s$  toute suite  $u$  qui satisfait :

- $n + 1 \leq p(n) \leq an + 1$  où  $1 < a < 2$  ( $u$  est donc une suite binaire).
- $\forall k, \exists n, p(n) \geq n + k$  (pour exclure les suites sturmien et quasi-sturmien).
- $u$  est récurrente.

### 3 Les graphes de mots des suites $u_s$

**Lemme 1** Soit  $u$  une suite de type  $u_s$ , il existe une infinité de  $n \in \mathbb{N}$  tels que  $\Gamma_n$  est de la forme  $G_s(x, y)$

C'est à dire qu'il y a une infinité de graphes de genre  $G_s$  dans une suite  $u_s$  (le lemme se démontre par l'absurde). Dans ce qui suit, on va étudier l'évolution des graphes de mots entre deux graphes consécutifs de genre  $G_s$ .

Soit  $\Gamma_n = G_s(x, y)$  un graphe de mots de type  $G_s$  d'une suite infinie  $u$ , le facteur bispécial de ce graphe va subir un éclatement qui va donner un des graphes suivants :

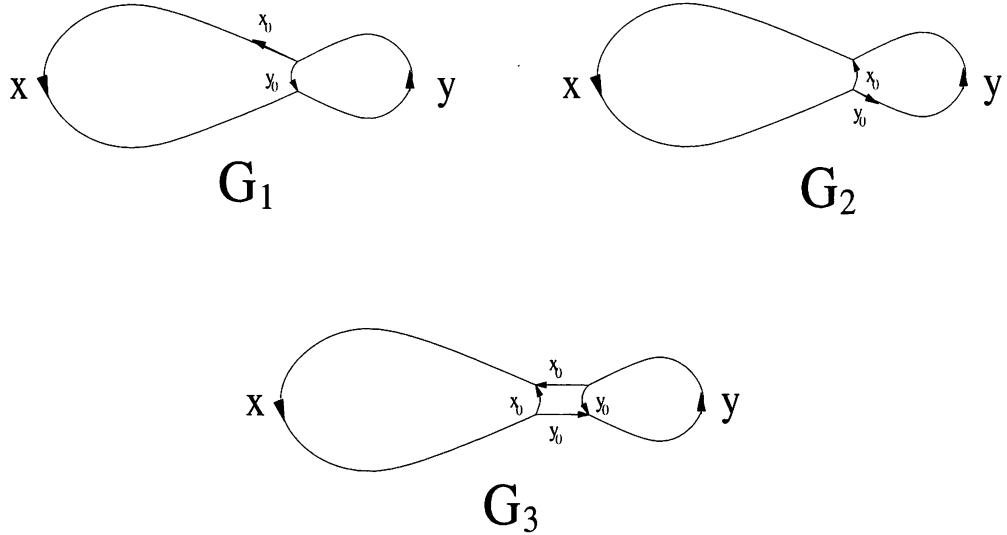


FIG. 2 –  $\Gamma_{n+1}$

- Dans le cas de  $G_1$  et  $G_2$ , ce nouveau graphe est sturmien, et il évolue jusqu'à  $\Gamma_{n+|y|} = G_s(xy, y)$  ou  $\Gamma_{n+|x|} = G_s(yx, x)$ .

**Notation** Notons  $O_{1,x}$  (respectivement  $O_{1,y}$ ) l'évolution d'un graphe sturmien  $G_s(x, y)$  en faisant un tour sur la boucle  $x$  (respectivement  $y$ ).

On peut noter ainsi :

$$\begin{aligned} O_{1,y}(G_s(x, y)) &= G_s(xy, y) \\ \text{et} \\ O_{1,x}(G_s(x, y)) &= G_s(yx, x) \end{aligned}$$

- Dans le cas de  $G_3$ , le graphe  $\Gamma_n$  évolue de la façon suivante :

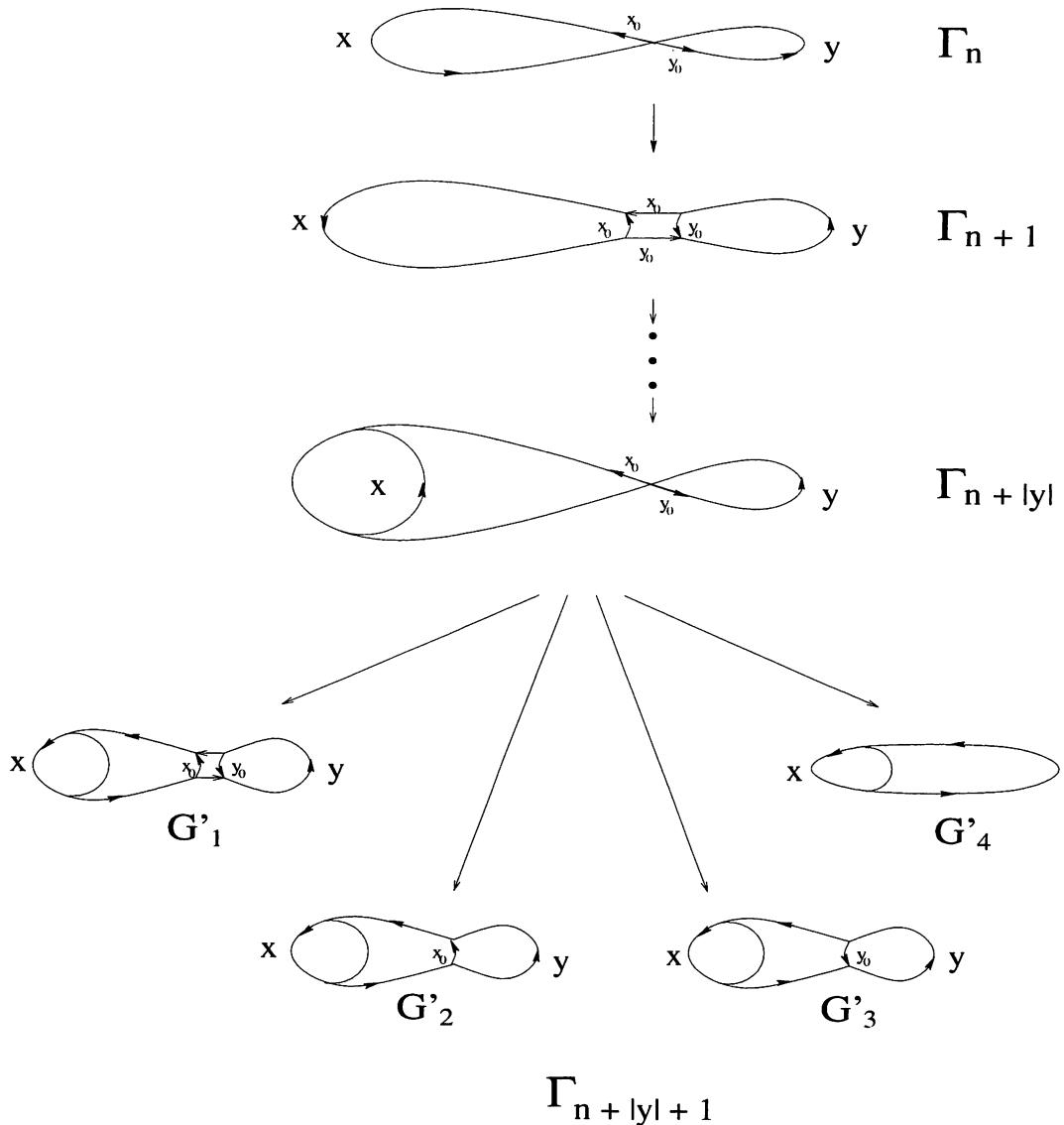
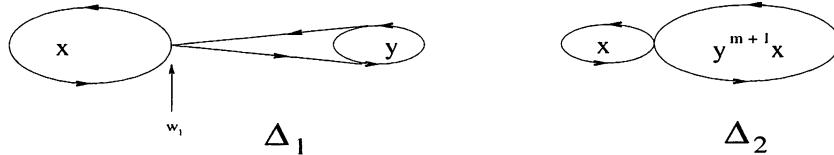


FIG. 3 – L'évolution du graphe  $G_3$

étudions les quatre cas :

- Dans le cas de  $G'_1$  et  $G'_2$ , l'évolution va devenir de plus en plus compliquée, ce qui va augmenter le nombre de cas qu'on doit étudier. De préférence, on va éviter ces deux cas pendant toute l'évolution.
  - Dans le cas de  $G'_4$ , on tombe sur une évolution d'un graphe sturmien.
  - Dans le cas de  $G'_3$ , l'évolution va continuer pour faire un tour sur la boucle  $x$  en donnant à la fin un des deux graphes  $\Delta_1$  et  $\Delta_2$  (à partir de  $G'_3$ , on suppose que l'évolution ne donne pas un graphe de type  $G'_1$  ou  $G'_2$ ) :
- Dans le graphe  $\Delta_1$ , le facteur bispécial  $w_1$  va éclater et l'évolution continue et

FIG. 4 – L'évolution du graphe  $G'_3$ 

devient de plus en plus incontrôlable, alors on doit éviter (de préférence) de tomber sur un graphe comme ceci ( $\Delta_1$ ), pour se ramener toujours à un graphe comme  $\Delta_2$ . L'idée qu'on va utiliser dans cet article est de mettre quelques conditions sur la complexité pour forcer les graphes à suivre le chemin et l'évolution qu'on désire. Par exemple en empêchant la complexité d'être trop proche de  $2n$ , ce qui limite le nombre de graphes consécutifs ayant plus d'un facteur spécial à droite, et interdit donc les évolutions très compliquées.

Remarquons que si l'évolution nous donne le graphe  $\Delta_2$ , alors la durée pendant laquelle des graphes consécutifs ont plus d'un facteur spécial à droite est majorée par  $|x| - 1$ . Dans le cas de  $\Delta_1$ , elle est au contraire minorée par  $|x|$ .

**Notation** Notons l'évolution d'un graphe  $G_s(x, y)$  en faisant  $m$  tours sur la petite boucle  $y$  et en s'arrêtant après un temps inférieur à  $|x| - 1$  par  $O_{m+1}$ .

On a donc  $O_{m+1}(G_s(x, y)) = G_s(y^{m+1}x, x)$ , d'où on remarque que  $O_1(G_s(x, y)) = G_s(yx, x)$  : on retrouve l'évolution  $O_{1,x}$  définie ci-dessus.

**Lemme 2** Soit  $u$  une suite de type  $u_s$ .

Si  $a \leq \frac{4}{3}$  alors  $s(n) = p(n+1) - p(n) \leq 2$ .

La preuve de ce lemme se base sur les propositions suivantes :

**Proposition 1** Soit  $u$  une suite de type  $u_s$ . Soit  $T$  la durée d'une évolution de graphes consécutifs ayant plus d'un facteur spécial à droite, commençant par un graphe  $G_s(x, y)$ .

Si  $a \leq \frac{4}{3}$  alors  $T \leq |x| - 1$

**Proposition 2** Soit  $u$  une suite telle que  $p(n) \leq an + 1$ ,  $1 < a < 2$

Si  $s(n) \geq 2 \quad \forall n \in [n_1, n_2[$  alors  $T = n_2 - n_1 \leq \frac{a-1}{2-a}n_1$ .

On peut déduire du lemme 2 et de l'étude des évolutions qui précède le théorème suivant :

**Théorème 1** Soit  $u$  une suite de type  $u_s$  avec  $a \leq \frac{4}{3}$ , et  $(\Gamma_{n_i})$  la suite de ses graphes de mots de genre  $G_s$ . Alors l'évolution entre  $\Gamma_{n_i}$  et  $\Gamma_{n_{i+1}}$ , pour tout  $i \geq 1$ , est de l'un des types  $O_{1,y}$  ou  $O_m$ ,  $m \geq 1$ . On a donc, en posant  $\Gamma_{n_i} = G_s(x_i, y_i)$  :

$$\begin{cases} x_{i+1} = x_i y_i \\ y_{i+1} = y_i \end{cases} \quad \text{ou} \quad \begin{cases} x_{i+1} = y_i^{m+1} x_i \\ y_{i+1} = x_i \end{cases}$$

où  $m \geq 0$ .

En contrôlant l'évolution des graphes  $\Gamma_{n_i}$ , on peut donc, grâce à ce résultat, construire des suites de type  $u_s$  dont la fonction de complexité a les propriétés désirées. Ainsi, on peut construire des suites de très faible complexité :

**Théorème 2** Pour tout nombre réel  $a$ ,  $a > 1$ , il existe une suite  $u$  telle que :

$$1 < \underline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} < \overline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} \leq a$$

**Preuve** La preuve de ce théorème est purement constructive. On fait subir au graphe des mots l'évolution  $O_2$ , puis  $l$  fois l'évolution  $O_{1,y}$ , puis à nouveau  $O_2$ ,  $l$  fois  $O_{1,y}$ , etc.

Soit le graphe  $\Gamma_{n_i}$  défini comme ceci :  $\Gamma_{n_i} = G_s(x_i, y_i) = (O_{1,y}^l \cdot O_2)^i(\Gamma_{n_0})$  pour  $l \in \mathbb{N}$  fixé et  $\Gamma_{n_0} = G_s(10, 0)$ .

**Notation** Notons  $u_{s2}$  toute suite dont les graphes de mots subissent l'évolution  $(O_{1,y}^l \cdot O_2)^\infty(\Gamma_{n_0})$ . La fonction de complexité d'une telle suite ne dépend que de  $l$ .

Cherchons les limites inférieure et supérieure de  $\frac{p(n)}{n}$  des suites  $u_{s2}$ . Le graphe  $\Gamma_{n_i}$  vérifie :

$$\begin{cases} x_{i+1} &= y_i^2 x_i^{l+1} \\ y_{i+1} &= x_i \end{cases}$$

Donc

$$\begin{pmatrix} |x_i| \\ |y_i| \end{pmatrix} = \begin{pmatrix} l+1 & 2 \\ 1 & 0 \end{pmatrix}^i \begin{pmatrix} |x_0| \\ |y_0| \end{pmatrix}$$

D'où

$$\begin{cases} |x_i| &= \alpha \left(\frac{l+1+\sqrt{\Delta}}{2}\right)^i + \beta \left(\frac{l+1-\sqrt{\Delta}}{2}\right)^i \\ |y_i| &= \alpha' \left(\frac{l+1+\sqrt{\Delta}}{2}\right)^i + \beta' \left(\frac{l+1-\sqrt{\Delta}}{2}\right)^i \end{cases}$$

Avec  $\lambda_1 = \left(\frac{l+1+\sqrt{\Delta}}{2}\right)$ ,  $\lambda_2 = \left(\frac{l+1-\sqrt{\Delta}}{2}\right)$  les deux valeurs propres de la matrice ( $\Delta = l^2 + 2l + 9$  est le discriminant de la matrice) et  $\alpha = 1 + \frac{l+3}{\sqrt{\Delta}}$ ,  $\beta = 1 - \frac{l+3}{\sqrt{\Delta}}$ ,  $\alpha' = \frac{1}{2} + \frac{3-l}{2\sqrt{\Delta}}$ ,  $\beta' = \frac{1}{2} + \frac{l-3}{2\sqrt{\Delta}}$ .

La durée des graphes consécutifs ayant deux facteurs spéciaux à droite (respectivement un seul facteur spécial à droite) entre deux graphes  $\Gamma_{n_i}$  et  $\Gamma_{n_{i+1}}$  est  $|y_i|$  (respectivement  $(l+1)|x_i| - |y_i|$ ). On pose  $n'_i = n_i + |y_i|$ .

Le schéma ci-dessous nous donne la complexité des suites  $u_{s2}$  :

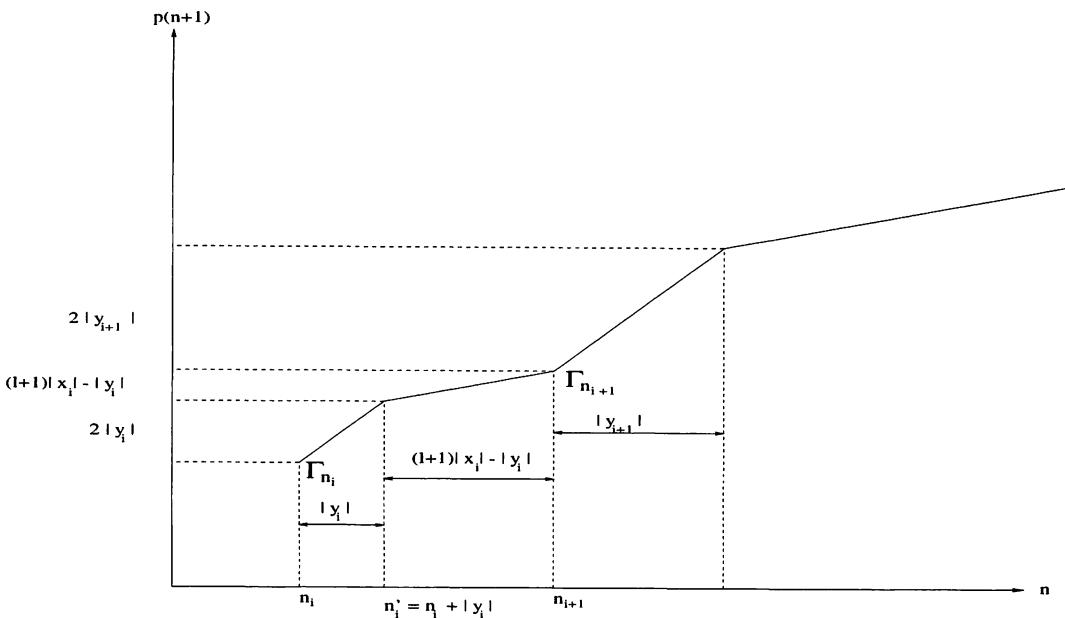


FIG. 5 – La complexité des suites  $u_{s2}$

$$\begin{aligned}
\overline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} &= \overline{\lim}_{i \rightarrow \infty} \frac{p(n'_i + 1)}{n'_i + 1} = \overline{\lim}_{i \rightarrow \infty} \frac{\sum_{j=0}^{i-1} (p(n'_{j+1} + 1) - p(n'_j + 1))}{\sum_{j=0}^{i-1} (n'_{j+1} - n'_j)} \\
&= \lim_{j \rightarrow \infty} \frac{(l+1)|x_j| - |y_j| + 2|y_{j+1}|}{(l+1)|x_j| - |y_j| + |y_{j+1}|} \\
&= 1 + \frac{\alpha}{(l+2)\alpha - \alpha'} = 1 + \frac{5l + \sqrt{\Delta} + 9}{6l^2 + 22l + 18}. \\
\overline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} &= \overline{\lim}_{i \rightarrow \infty} \frac{p(n_i + 1)}{n_i + 1} = \overline{\lim}_{i \rightarrow \infty} \frac{\sum_{j=0}^{i-1} (p(n_{j+1} + 1) - p(n_j + 1))}{\sum_{j=0}^{i-1} (n_{j+1} - n_j)} \\
&= \lim_{j \rightarrow \infty} \frac{(l+1)|x_j| + |y_j|}{(l+1)|x_j|} \\
&= 1 + \frac{\alpha'}{(l+1)\alpha} = 1 + \frac{2}{(l+1)(\sqrt{\Delta} + l + 1)}.
\end{aligned}$$

On remarque que  $\overline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} - 1 \sim \frac{1}{l}$  et  $\underline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} - 1 \sim \frac{1}{l^2}$ .

Donc, on peut déduire que si  $l$  est très grand alors la complexité  $p(n)$  est très proche de  $n + 1$ . Alors pour trouver une suite  $u$  telle que  $1 < \underline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} < \overline{\lim}_{n \rightarrow \infty} \frac{p(n)}{n} \leq a$ , il suffit de prendre une suite  $u_{s2}$  avec un  $l$  suffisamment grand pour que  $1 + \frac{\alpha}{(l+2)\alpha - \alpha'} \leq a$ .

On donne ci-dessous un exemple plus concret d'une substitution extraite de l'évolution précédente.

En définissant le morphisme  $\delta$  par  $\delta(0) = 10$  et  $\delta(1) = 00(10)^l$ , on trouve que  $y_{i+1} = \delta(y_i)$  et  $x_{i+1} = \delta(x_i)$ ,  $\forall n \in \mathbb{N}$ . On peut démontrer par récurrence que  $y_i$  est un préfixe de  $y_{i+2}$ . Définissons maintenant la suite  $v$  comme ceci :  $v = \lim_{i \rightarrow \infty} y_{2i}$ . Cette limite existe pour les deux raisons suivantes :

$$\left\{
\begin{array}{l}
\lim_{i \rightarrow \infty} |y_{2i}| = +\infty \\
\forall i \in \mathbb{N}, y_{2i} \text{ est un préfixe de } y_{2i+2} \text{ (preuve par récurrence)}
\end{array}
\right.$$

On a  $y_{2i+2} = \delta^2(y_{2i})$ ,  $x_{2i+2} = \delta^2(x_{2i})$ ,  $\forall n \in \mathbb{N}$  et  $v = \lim_{i \rightarrow \infty} y_{2i}$ . Ainsi  $v = (\delta^2)^w(0)$ . Donc la suite  $v$  est un exemple de suite de type  $u_{s2}$  qui peut être obtenue comme un point fixe d'une substitution.

**Remarque** Pour  $l = 0$ , on trouve l'exemple présenté par G. RAUZY dans [6].

## Références

- [1] P. ALESSANDRI, Codages de rotations et basses complexités, Université Aix-Marseille II, Thèse, 1996.
- [2] P. ARNOUX, G. RAUZY, Représentation géométrique des suites de complexité  $2n + 1$ , Bull. Soc. Math. France **119** (1991), 199-215.
- [3] J. CASSAIGNE, Complexité et facteurs spéciaux, Bull. Belg. Math. Soc. **4**(1997), 67-88.
- [4] A. DESIDERI-BRACCO, Vers une construction générale pour les suites de complexité  $2n$ , mémoire de DEA sous la direction de J. CASSAIGNE (1997).

- [5] M. MORSE, G. A. HEDLUND, Symbolic dynamics II : Sturmian trajectories, Amer. J. Math. **62** (1940), 1-42.
- [6] G. RAUZY, Suites à termes dans un alphabet fini, Séminaire de Théorie des Nombres de Bordeaux **25** (1982-1983).
- [7] G. ROTE, Sequences with subword complexity  $2n$ , Journal of number theory **46**, 196-213(1994).
- [8] M.-L. SANTINI, échanges de trois intervalles, Thèse, Université Aix-Marseille II (1994).

# Enumeration of multi-colored rooted maps

Didier Arquès \*, Anne Micheli \*

## Abstract

We present a study of  $n$ -colored rooted maps in orientable and locally orientable surfaces. As far as we know, no work on these maps has yet been published. We give a system of  $n$  functional equations verified by  $n$ -colored orientable rooted maps regardless of genus and with respect to edges and vertices. We exhibit the solution of this system as a vector where each component has a continued fraction form and we deduce a new equation generalizing the Dyck equation for rooted planar trees. Similar results are shown for  $n$ -colored rooted maps in locally orientable surfaces.

## Résumé

Nous présentons une étude sur les cartes pointées  $n$ -coloriées sur des surfaces orientables et localement orientables. A notre connaissance, aucun travail sur ces cartes n'a encore été publié. Nous donnons un système de  $n$  équations fonctionnelles vérifiées par les cartes pointées  $n$ -coloriées orientables comptées indépendamment du genre et en fonction des arêtes et sommets. Nous montrons que la solution de ce système est un vecteur dont chaque composante s'écrit sous forme d'une fraction continue et déduisons une nouvelle équation généralisant l'équation de Dyck pour les arbres planaires pointés. Des résultats similaires sont obtenus pour les cartes pointées  $n$ -coloriées sur des surfaces localement orientables.

## 1 Introduction

The enumerative study of maps starts in 1962 with W. Tutte [10, 11], who counts the number of rooted planar maps with  $n$  edges. In 1975, R. Cori [6] describes planar maps as combinatorics objects. In 1987, D. Arquès [1] determines functional relations verified by generating functions of rooted maps on the torus and obtain closed formulae to count these maps by vertices and faces. Many works follow on maps of greater genus, orientable or not, as for example the works of E. Bender and E. Canfield and also D. Arquès and A. Giorgetti [5, 4]. Some works [10, 12] deal also with hypermaps, equivalent to 2-colored maps.

The study of rooted maps counted regardless of genus is not much extended. This study start with T. Walsh and A. Lehman [13] who give a recursive relation on the number of rooted maps counted by edges. In 1990, D. Jackson and T. Visentin [8] use an algebraic approach in order to obtain a closed formula for the generating functions of rooted orientable maps counted by edges and vertices and extend this approach to enumerate hypermaps [9]. The case of these maps in locally orientable surfaces is treated

---

\*{arques,micheli}@univ-mlv.fr, Équipe de synthèse d'images, Institut Gaspard Monge, Unité C-NRS UPRES-A 8049, Université de Marne la Vallée, 5 Boulevard Descartes, Champs sur Marne, 77454 Marne la Vallée Cedex 2, France

by I. Goulden and D. Jackson [7]. More recently D. Arquès and J.-F. Béraud [2, 3], give the functional equation verified by the generating function of rooted maps counted by edges and vertices, and exhibit the function in continued fraction form.

We present a topological study (with exhibition of a functional topological equation) of  $n$ -colored rooted maps counted regardless of genus. After recalling some basic notions on maps, we prove that the generating function of  $n$ -colored orientable rooted maps is a solution of a set of partial differential equations. Then we solve this set of equations to obtain the generating function of  $n$ -colored orientable rooted maps with root vertex of color  $i$ ,  $1 \leq i \leq n$ , in continued fraction form. This continued fraction brings to the fore a generalized Dyck equation for these maps. As these results extend themselves to  $n$ -colored rooted maps in locally orientable surfaces, we next give the set of functional equations verified by these maps and the solution with a continued fraction form.

## 2 Definitions

We recall some definitions used afterwards (for further details, see for example [6, 2]).

A topological map  $C$  in an orientable surface  $\Sigma$  of  $\mathbb{R}^3$  is a partition of  $\Sigma$  in three finite sets of cells :

1. The set of vertices of  $C$ , which is a finite set of dots ;
2. The set of edges of  $C$ , which is a finite set of open Jordan arcs, pairwise disjoint, whose extremities are vertices ;
3. The set of faces of  $C$ . Each face is simply connected and its border is the union of vertices and edges.

The genus of the map  $C$  is the genus of  $\Sigma$ . A cell is incident to another cell if one is contained in the boundary of the other. An isthmus is an edge incident on both sides to the same face. We call half-edge an oriented edge of the map.

We call  $B$  the set of half-edges of the map. To each half-edge is associated its initial vertex, its final vertex and its underlying edge.  $\alpha$  (resp.  $\sigma$ ) is the permutation in  $B$  associating to each half-edge  $b$  its opposite half-edge (resp. the first half-edge met when turning round the initial vertex of  $b$  in the positive way of the surface). The cycles of  $\alpha$  (resp.  $\sigma$ ) represent the edges (resp. the vertices) of the map. The cycles of  $\sigma \circ \alpha$  are the oriented borders of the faces of the map.  $(B, \sigma, \alpha)$  constitute the combinatoric definition of the topological orientable map associated  $C$ .

A map is said rooted if a half-edge  $b$  is distinguished. The half-edge  $b$  is called the root half-edge of the map, and its initial vertex is the root vertex. By convention, the planar map reduced to an unique vertex is said to be rooted.

Two orientable maps of the same genus are isomorphic if there is a homeomorphism of the surfaces, preserving its orientation, mapping vertices, edges and faces of one map onto vertices, edges and faces respectively of the other map. An isomorphic class of orientable rooted maps of genus  $g$  will simply be called orientable rooted map.

**Definition 1** *A  $n$ -colored orientable rooted map ( $n > 1$ ) is a rooted map, where a maximum of  $n$  colors are used to color the vertices and such as each edge be incident to two vertices of different colors.*

The property “ $n$ -colored” is compatible with the equivalence relation whose classes are the rooted maps.

Let  $\mathcal{I}_n$  be the set  $\{1, \dots, n\}$ . Let  $c_i, i \in \mathcal{I}_n$ , be the variable whose exponent represents the number of half-edges with initial vertex of color  $i$ . Let  $y$  be the variable whose exponent represents the number of vertices of the map. Let  $M_{n,i}, i \in \mathcal{I}_n$ , be the generating function of  $n$ -colored orientable rooted maps whose color of the root vertex is  $i$ , counted by vertices and half-edges whose initial vertex is of color  $j \in \mathcal{I}_n$ .

### 3 Functional equation

We present in this section a system of partial differential equations and show that the vector whose components are the generating functions  $M_{n,i}, 1 \leq i \leq n$ , is the unique solution of this system.

**Theorem 1**  $\forall i \in \mathcal{I}_n, M_{n,i} = M_{n,i}(y, c_1, \dots, c_n)$  is solution of the following partial differential equation :  $M_{n,i} = y + c_i M_{n,i} \sum_{j=1, j \neq i}^n c_j M_{n,j} + c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial M_{n,i}}{\partial c_j}$ .  $(D_i)$

Furthermore  $(M_{n,i})_{1 \leq i \leq n}$  is the unique solution of the system  $\{(D_i), 1 \leq i \leq n\}$ .

**Proof of theorem 1** 1. The set of  $n$ -colored orientable rooted maps, with root vertex of color  $i$  is composed of :

- the map reduced to the root vertex of color  $i$ ,
- the set of  $n$ -colored orientable rooted maps, with root vertex of color  $i$ , whose edge supporting the root half-edge is an isthmus whose deletion disconnects the map into two maps,
- the set of  $n$ -colored orientable rooted maps, with root vertex of color  $i$ , whose edge supporting the root half-edge is not an isthmus whose deletion disconnects the map.

The generating functions of these sets are respectively :

- $y$  (straightforward : a vertex, no half-edge).
- $c_i M_{n,i} \sum_{j=1, j \neq i}^n c_j M_{n,j}$ . The deletion of the root isthmus  $b$  gives two  $n$ -colored orientable rooted maps, one whose root half-edge is  $\sigma(b)$  and root vertex is of color  $i$ , the other whose root half-edge is  $(\sigma \circ \alpha)(b)$  and root vertex is of a color  $j \in \mathcal{I}_n, j \neq i$ . The contribution of the isthmus is  $c_i c_j$ .
- $c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial M_{n,i}}{\partial c_j}$ . The deletion of the edge supporting the root half-edge  $b$  doesn't disconnect the map. To enumerate these maps, one counts the number of ways to add an edge supporting the new root half-edge  $b$  to any  $n$ -colored orientable rooted map, of root half-edge  $e_i$ , of root vertex of color  $i$ , as follows :
  - $\sigma(b) = e_i$ ,
  - the final vertex of  $b$  be any vertex of color  $j, j \neq i$ .

For a given  $n$ -colored orientable rooted map, the number of ways to add the edge supporting the root half-edge  $b$  as described, such as the final vertex of  $b$  be of color  $j, j \neq i$ , is the number of vertices of color  $j$ . Thus applying the operator  $c_j \frac{\partial}{\partial c_j}$  to the generating function  $M_{n,i}$  gives us the generating function of  $n$ -colored orientable rooted maps, with root vertex of color  $i$ , of final vertex

of the root half-edge of color  $j$ ,  $j \neq i$ , whose edge supporting the root half-edge is not an isthmus whose deletion disconnects the map. The contribution of the edge supporting the root half-edge  $b$  is  $c_i c_j$ .

## 2. uniqueness

$\mathcal{L} = \mathbb{Z}[y][[c_1, \dots, c_n]]$ , the algebra of the formal series in the commutative variables  $c_1, \dots, c_n$ , with coefficients in  $\mathbb{Z}[y]$ , supplied with the distance defined for two series  $S_1$  et  $S_2$  as follows :  $d(S_1, S_2) = 2^{-\omega(S_1 - S_2)}$  where

$\omega(S_1) = \min\{\alpha_1 + \dots + \alpha_n / \text{the coefficient of } c_1^{\alpha_1} \dots c_n^{\alpha_n} \text{ in } S_1 \text{ is not equal to zero}\}$ , is a complete metric space. Then  $\mathcal{L}^n$  supplied with the product distance defined for two elements  $S = (S_1, \dots, S_n)$  and  $T = (T_1, \dots, T_n)$  as  $p(S, T) = \sup_{1 \leq i \leq n} d(S_i, T_i)$

is a complete metric space. One can easily show that the system composed of the right members of equations  $(D_i), 1 \leq i \leq n$ , define a contracting operator on  $(\mathcal{L}^n, p)$  and has an unique fixed point  $\diamond$

## 4 Determination of the generating function of $n$ -colored orientable rooted maps

The theorem 2 presents a generalization of the system of partial differential equations seen in theorem 1. This generalization defines a new family of formal series, which are related to each other as shown in proposition 1. These relations lead to the solution of the system of equations  $\{(D_i), 1 \leq i \leq n\}$  (see theorem 3).

Let  $u_i, i \in \mathcal{I}_n$ , be the variable whose exponent represents the number of vertices of color  $i$ . Let  $M_i(u), i \in \mathcal{I}_n$ , be the generating function of  $n$ -colored orientable rooted maps whose color of the root vertex is  $i$ , counted by vertices of color  $j \in \mathcal{I}_n$  and half-edges whose initial vertex is of color  $j \in \mathcal{I}_n$ .

**Theorem 2**  $\forall i \in \mathcal{I}_n, M_i(u) = M_i(u)(c_1, \dots, c_n)$  is solution of the following partial differential equation :  $M_i(u) = u_i + c_i M_i(u) \sum_{j=1, j \neq i}^n c_j M_j(u) + c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial M_i(u)}{\partial c_j}$ .  $(E_{i,u})$

Furthermore  $(M_i(u))_{1 \leq i \leq n}$  is the unique solution of the system  $\{(E_{i,u}), 1 \leq i \leq n\}$ .

**Proof of theorem 2** The generating functions of the three sets given in the proof of theorem 1 are obtained in a similar way in this case, except for the generating function of the map reduced to the root vertex of color  $i$  which is  $u_i$ .

The system composed of the right members of  $(E_{i,u}), 1 \leq i \leq n$ , define a contracting operator on the complete space  $(\mathcal{L}^n, p)$  and has an unique fixed point  $\diamond$

**Remark 1** If  $u = (y, \dots, y)$ , theorem 1 implies  $M_i(u) = M_{n,i}$ .

**Definition 2**  $\forall i \in \mathcal{I}_n$ , let  $U(i, u) = (U_1(i, u), \dots, U_n(i, u))$  be the vector such that  $\forall j \in \mathcal{I}_n, U_j(i, u) = u_j + \delta_{ij}$ , where  $\delta_{ij}$  represents the Kronecker symbol.

**Proposition 1**  $\forall i \in \mathcal{I}_n$ , the  $M_i(u)$  are related to each other by the following formula :

$$M_i(u) = \frac{u_i}{1 - c_i \sum_{j=1, j \neq i}^n c_j M_j(U(i, u))}. \quad (1)$$

The proof of this proposition is shown in section 6.

**Remark 2** Observe that in  $M_j(U(i, u))$ , the exponent of  $u_i$  does not count any more the number of vertices of color  $i$ , but the number of vertices of a sub-set of these vertices.

We deduce from this proposition :

**Theorem 3** *The generating function  $M_{n,1}$  can be written in continued fraction form :*

$$\frac{1}{c_1} \frac{a_1^{(0)}}{\frac{1}{c_1} - \sum_{j_1=2}^n \frac{a_{j_1}^{(0,1)}}{\frac{1}{c_{j_1}} - \sum_{j_2=1, j_2 \neq j_1}^n \frac{a_{j_2}^{(0,1,j_1)}}{\frac{1}{c_{j_2}} - \sum_{j_3=1, j_3 \neq j_2}^n \frac{a_{j_3}^{(0,1,j_1,j_2)}}{\frac{1}{c_{j_3}} - \dots}}},$$

$$\text{in which } a^{(0)} = (a_1^{(0)}, \dots, a_n^{(0)}) = (y, \dots, y)$$

$$a^{(0,1)} = (a_1^{(0,1)}, \dots, a_n^{(0,1)}) = (y+1, y, \dots, y)$$

$$a^{(0,1,j_1, \dots, j_k)} = (a_1^{(0,1,j_1, \dots, j_k)}, \dots, a_n^{(0,1,j_1, \dots, j_k)}) = a^{(0,1,j_1, \dots, j_{k-1})} + (\delta_{ij_k}, \dots, \delta_{ij_k})$$

**Proof of theorem 3** Using (1) repeatedly to infinity, one obtains :

$$\begin{aligned} M_{n,1} &= \frac{y}{c_1 \left( \frac{1}{c_1} - \sum_{j=2}^n c_j M_j \left( U(1, a^{(0)}) \right) \right)} \\ &= \frac{1}{c_1} \frac{y}{\frac{1}{c_1} - \sum_{j_1=2}^n \frac{y}{\frac{1}{c_{j_1}} - \sum_{j_2=1, j_2 \neq j_1}^n c_{j_2} M_{j_2} \left( U(j_1, a^{(0,1)}) \right)}} \text{ as } a^{(0,1)} = U(1, a^{(0)}) \\ &= \frac{1}{c_1} \frac{y}{\frac{1}{c_1} - \sum_{j_1=2}^n \frac{y}{\frac{1}{c_{j_1}} - \sum_{j_2=1, j_2 \neq j_1}^n \frac{y}{\frac{1}{c_{j_2}} - \sum_{j_3=1, j_3 \neq j_2}^n c_{j_3} M_{j_3} \left( U(j_2, a^{(0,1,j_1)}) \right)}} = \dots \end{aligned}$$

From  $a^{(0,1,j_1, \dots, j_k)} = U(j_k, a^{(0,1,j_1, \dots, j_{k-1})})$ , we obtain the equations verified by the  $a_i^{(0,1,j_1, \dots, j_k)}$  ◊

From proposition 1, one obtains :

**Corollary 1** *The generating function of  $n$ -colored orientable rooted maps with root vertex of color  $i$ , is the solution of the following generalized Dyck equation :*

$$M_{n,i} = y + c_i M_{n,i} \sum_{j=1, j \neq i}^n c_j M_j(u)$$

in which  $u = (u_j)_{1 \leq j \leq n} = (y + \delta_{ij})_{1 \leq j \leq n}$ .

## 5 Application to 3-colored orientable rooted maps

The results obtained are applied to 3-colored orientable rooted maps.

**Theorem 4**  $\forall i, j, k \in \{1, 2, 3\}$  with  $i, j, k$  all distinct, we have :

$$y + (c_i c_j M_{3,j} + c_i c_k M_{3,k} - 1) M_{3,i} + c_i c_j^2 \frac{\partial M_{3,i}}{\partial c_j} + c_i c_k^2 \frac{\partial M_{3,i}}{\partial c_k} = 0. \quad (2)$$

**Corollary 2** The generating function  $M_{3,1}$  can be written in continued fraction form

$$\frac{y}{c_1 \left( \frac{1}{c_1 - \frac{1}{c_2 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+1}{c_1 - \frac{y+2}{c_2 - \frac{y}{c_1 - \dots}}}}}} \right.} \\ \left. - \frac{y}{\frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+1}{c_1 - \frac{y+2}{c_3 - \frac{y}{c_1 - \dots}}}}}} \right.} \\ \left. - \frac{y}{c_2 \left( \frac{1}{c_1 - \frac{1}{c_2 - \frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+1}{c_3 - \frac{y+2}{c_1 - \dots}}}}}} \right.} \right.} \\ \left. \left. - \frac{y}{\frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+1}{c_3 - \frac{y+2}{c_1 - \dots}}}}}} \right) \right)$$

## 6 Proof of proposition 1

### 6.1 Two preliminary lemmas

**Lemma 1**  $\forall j, k \in \mathcal{I}_n$ ,

$$c_j \frac{\partial M_k(u)}{\partial c_j} - c_k \frac{\partial M_j(u)}{\partial c_k} = 0. \quad (3)$$

**Proof of lemma 1** The set  $C_{j \rightarrow k}$  of  $n$ -colored orientable rooted maps, with root vertex of color  $j$ , of final vertex of the root half-edge of color  $k$ , whose edge supporting the root half-edge is not an isthmus whose deletion disconnects the map, is in one-to-one correspondence with  $C_{k \rightarrow j}$ . Furthermore  $c_j c_k^2 \frac{\partial M_j(u)}{\partial c_k}$  is the generating function of  $C_{j \rightarrow k}$  and  $c_k c_j^2 \frac{\partial M_k(u)}{\partial c_j}$  is the generating function of  $C_{k \rightarrow j}$ , both generating functions being counted by vertices of color  $i, i \in \mathcal{I}_n$ , and half-edges whose initial vertex is of color  $i, i \in \mathcal{I}_n$  (see proof of theorem 2)  $\diamond$

**Lemma 2**  $\forall i, j \in \mathcal{I}_n$ ,

$$M_j(U(i, u)) = M_j(u) + \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} + \delta_{ij}. \quad (4)$$

**Proof of lemma 2** Let us show that (4) is true for any given  $i \in \mathcal{I}_n$ .

Let us define the functions  $\Lambda_{i,j}(u) = M_j(u) + \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} + \delta_{ij}, 1 \leq j \leq n$ . It will be shown that the vector  $(\Lambda_{i,j}(u))_{1 \leq j \leq n}$  is a solution of the system of equations  $\{E_{(j, U(i, u))}, 1 \leq j \leq n\}$ . As  $\{E_{(j, U(i, u))}, 1 \leq j \leq n\}$  has an unique solution (see theorem 2),  $(\Lambda_{i,j}(u))_{1 \leq j \leq n}$  is a solution of the system of equations  $\{E_{(j, U(i, u))}, 1 \leq j \leq n\}$ , implies  $M_j(U(i, u)) = \Lambda_{i,j}(u), \forall 1 \leq j \leq n$ .

$$\begin{aligned}
& \Lambda_{i,j}(u) - u_j - \delta_{ij} - c_j \Lambda_{i,j}(u) \sum_{k=1, k \neq j}^n c_k \Lambda_{i,k}(u) - c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial \Lambda_{i,j}(u)}{\partial c_k} \\
& = \left( M_j(u) - u_j - c_j M_j(u) \sum_{k=1, k \neq j}^n c_k M_k(u) - c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial M_j(u)}{\partial c_k} \right) \\
& \quad + \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} - c_j M_j(u) \sum_{k=1, k \neq j}^n c_k \left( \frac{c_k}{M_i(u)} \frac{\partial M_i(u)}{\partial c_k} + \delta_{ik} \right) \\
& \quad - c_j \left( \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} + \delta_{ij} \right) \sum_{k=1, k \neq j}^n c_k \left( M_k(u) + \frac{c_k}{M_i(u)} \frac{\partial M_i(u)}{\partial c_k} + \delta_{ik} \right) \\
& \quad - c_j^2 \sum_{k=1, k \neq j}^n c_k^2 \left( \frac{1}{M_i(u)} \frac{\partial^2 M_i(u)}{\partial c_k \partial c_j} - \frac{1}{M_i(u)^2} \frac{\partial M_i(u)}{\partial c_k} \frac{\partial M_i(u)}{\partial c_j} \right)
\end{aligned}$$

As from  $(E_{j,u})$ ,  $M_j(u) - u_j - c_j M_j(u) \sum_{k=1, k \neq j}^n c_k M_k(u) - c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial M_j(u)}{\partial c_k} = 0$ , it gives :

$$\begin{aligned}
& = \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} - c_j M_j(u) \sum_{k=1, k \neq j}^n c_k \left( \frac{c_k}{M_i(u)} \frac{\partial M_i(u)}{\partial c_k} + \delta_{ik} \right) \\
& \quad - \frac{c_j^2}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} \sum_{k=1, k \neq j}^n c_k (M_k(u) + \delta_{ik}) \\
& \quad - \delta_{ij} c_j \sum_{k=1, k \neq j}^n c_k \left( M_k(u) + \frac{c_k}{M_i(u)} \frac{\partial M_i(u)}{\partial c_k} + \delta_{ik} \right) - c_j^2 \sum_{k=1, k \neq j}^n \frac{c_k^2}{M_i(u)} \frac{\partial^2 M_i(u)}{\partial c_k \partial c_j}
\end{aligned}$$

Applying (3) to this equation :

$$\begin{aligned}
& = \frac{c_i}{M_i(u)} \left( \frac{\partial M_j(u)}{\partial c_i} - c_j M_j(u) \sum_{k=1, k \neq j}^n c_k \frac{\partial M_k(u)}{\partial c_i} - c_j \frac{\partial M_j(u)}{\partial c_i} \sum_{k=1, k \neq j}^n c_k M_k(u) \right. \\
& \quad \left. - c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial^2 M_j(u)}{\partial c_k \partial c_i} - \sum_{k=1, k \neq j}^n \delta_{ik} \left( c_j M_j(u) M_i(u) + 2c_i c_j \frac{\partial M_j(u)}{\partial c_i} \right) \right. \\
& \quad \left. - \delta_{ij} \left( M_j(u) \sum_{k=1, k \neq j}^n c_k M_k(u) + \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial M_j(u)}{\partial c_k} \right) \right)
\end{aligned}$$

Differentiating  $M_j(u)$  with respect to  $c_i$  in  $(E_{j,u})$  :

$$\begin{aligned}
\frac{\partial M_j(u)}{\partial c_i} & = \delta_{ij} M_j(u) \sum_{k=1, k \neq j}^n c_k M_k(u) + c_j \frac{\partial M_j(u)}{\partial c_i} \sum_{k=1, k \neq j}^n c_k M_k(u) \\
& \quad + c_j M_j(u) \sum_{k=1, k \neq j}^n c_k \frac{\partial M_k(u)}{\partial c_i} + \sum_{k=1, k \neq j}^n \delta_{ik} c_j M_i(u) M_j(u) \\
& \quad + \delta_{ij} \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial M_j(u)}{\partial c_k} + \sum_{k=1, k \neq j}^n 2\delta_{ik} c_i c_j \frac{\partial M_j(u)}{\partial c_i} + c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial^2 M_j(u)}{\partial c_i \partial c_k}
\end{aligned}$$

Therefore  $\Lambda_{i,j}(u) - u_j - \delta_{ij} - c_j \Lambda_{i,j}(u) \sum_{k=1, k \neq j}^n c_k \Lambda_{i,k}(u) - c_j \sum_{k=1, k \neq j}^n c_k^2 \frac{\partial \Lambda_{i,j}(u)}{\partial c_k} = 0 \diamond$

## 6.2 End of proof of proposition 1

$$(4) \text{ implies } \sum_{j=1, j \neq i}^n c_i c_j \left( M_j(u) - M_j(U(i, u)) + \frac{c_j}{M_i(u)} \frac{\partial M_i(u)}{\partial c_j} \right) = 0.$$

Multiplying by  $M_i(u)$  then adding to each member  $M_i(u)$ , one obtains :

$$M_i(u) \left( 1 - c_i \sum_{j=1, j \neq i}^n c_j M_j(U(i, u)) \right) + c_i M_i(u) \sum_{j=1, j \neq i}^n c_j M_j(u) + c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial M_i(u)}{\partial c_j} = M_i(u).$$

$$\text{It follows from } (E_{i,u}) \text{ that : } M_i(u) \left( 1 - c_i \sum_{j=1, j \neq i}^n c_j M_j(U(i, u)) \right) = u_i \diamond$$

## 7 $n$ -colored rooted maps in locally orientable surfaces

As the results on  $n$ -colored orientable rooted maps extend themselves rather easily to the locally orientable case, this section presents some results on  $n$ -colored rooted maps in locally orientable surfaces. After recalling the definition of a map in a locally orientable surface, the set of functional equations of the generating functions of  $n$ -colored rooted maps in locally orientable surfaces is given, which leads to the solution of this set of equations, as a vector whose components are continued fractions. Then these results are applied to 3-colored rooted maps in locally orientable surfaces.

### 7.1 Definitions

We recall some definitions that differ from those given in the orientable case (section 2).

A map in a locally orientable surface of  $\mathbb{R}^3$  will be called a locally orientable map.

Each edge of a locally orientable map has two ends and two sides, and therefore four half-edges, each one corresponding to a side and an end.

Two locally orientable maps of the same genus are isomorphic if there is a homeomorphism of the surfaces, preserving their roots, mapping vertices, edges and faces of one map onto vertices, edges and faces respectively of the other map. An isomorphic class of locally orientable rooted maps of genus  $g$  will simply be called locally orientable rooted map.

Let  $c_i$ ,  $i \in \mathcal{I}_n$ , be the variable whose exponent represents the number of half-edges with initial vertex of color  $i$ . Let  $y$  be the variable whose exponent represents the number of vertices of the map. Let  $\widetilde{M}_{n,i}$  be the generating function of  $n$ -colored locally orientable rooted maps whose color of the root vertex is  $i$ , counted by vertices and half-edges whose initial vertex is of color  $j \in \mathcal{I}_n$ .

### 7.2 Functional equation

**Theorem 5**  $\forall i \in \mathcal{I}_n$ ,  $\widetilde{M}_{n,i} = \widetilde{M}_{n,i}(y, c_1, \dots, c_n)$  is solution of the following partial differential equation :  $\widetilde{M}_{n,i} = y + c_i \widetilde{M}_{n,i} \sum_{j=1, j \neq i}^n c_j \widetilde{M}_{n,j} + 2c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial \widetilde{M}_{n,i}}{\partial c_j}$ .  $(\tilde{D}_i)$

Furthermore  $(\widetilde{M}_{n,i})_{1 \leq i \leq n}$  is the unique solution of the system  $\{(\tilde{D}_i), 1 \leq i \leq n\}$ .

**Proof of theorem 5** The proof is the same as in the orientable case, except for the term corresponding to  $n$ -colored locally orientable rooted maps whose removal of the edge supporting the root half-edge  $b$  does not disconnect the map. The contribution of the edge supporting the root half-edge  $b$  is  $2c_i c_j$ , as there is two ways to choose the side of the half-edge  $\diamond$

### 7.3 Enumeration of $n$ -colored locally orientable rooted maps

Let  $u_i$ ,  $i \in \mathcal{I}_n$  be the variable whose exponent represents the number of vertices of color  $i$ . Let  $\widetilde{M}_i(u)$  be the generating function of  $n$ -colored locally orientable rooted maps whose color of the root vertex is  $i$ , counted by vertices of color  $j \in \mathcal{I}_n$  and half-edges whose initial vertex is of color  $j \in \mathcal{I}_n$ .

**Theorem 6**  $\forall i \in \mathcal{I}_n$ ,  $\widetilde{M}_i(u) = \widetilde{M}_i(u)(c_1, \dots, c_n)$  is solution of the following partial differential equation :  $\widetilde{M}_i(u) = y + c_i \widetilde{M}_i(u) \sum_{j=1, j \neq i}^n c_j \widetilde{M}_j(u) + 2c_i \sum_{j=1, j \neq i}^n c_j^2 \frac{\partial \widetilde{M}_i(u)}{\partial c_j}$ .  $(\widetilde{E}_{i,u})$

Furthermore  $(\widetilde{M}_i(u))_{1 \leq i \leq n}$  is the unique solution of the system  $\{(\widetilde{E}_{i,u}), 1 \leq i \leq n\}$ .

**Remark 3** If  $u = (y, \dots, y)$ , theorem 5 implies  $\widetilde{M}_i(u) = \widetilde{M}_{n,i}$ .

**Definition 3**  $\forall i \in \mathcal{I}_n$ , let  $V(i, u) = (V_1(i, u), \dots, V_n(i, u))$  be a vector such as  $\forall j \in \mathcal{I}_n$ ,  $V_j(i, u) = u_j + 2\delta_{ij}$ .

**Proposition 2**  $\forall i \in \mathcal{I}_n$ , the  $\widetilde{M}_i(u)$ , are related to each other by the following formula :

$$\widetilde{M}_i(u) = \frac{u_i}{1 - c_i \sum_{j=1, j \neq i}^n c_j \widetilde{M}_j(V(i, u))}.$$

**Proof of proposition 2** The proof of this proposition is similar to the proof of proposition 1. Therefore we only give here the lemmas needed for the proof :

**Lemma 3**  $\forall j, k \in \mathcal{I}_n$ ,  $c_j \frac{\partial \widetilde{M}_k(u)}{\partial c_j} - c_k \frac{\partial \widetilde{M}_j(u)}{\partial c_k} = 0$ .

**Lemma 4**  $\forall i, j \in \mathcal{I}_n$ ,  $\widetilde{M}_j(V(i, u)) = \widetilde{M}_j(u) + 2 \frac{c_j}{\widetilde{M}_i(u)} \frac{\partial \widetilde{M}_i(u)}{\partial c_j} + 2\delta_{ij}$   $\diamond$

We deduce from this proposition :

**Theorem 7** The generating function  $\widetilde{M}_{n,1}$  can be written in continued fraction form :

$$\frac{1}{c_1} \frac{a_1^{(0)}}{\frac{1}{c_1} - \sum_{j_1=2}^n \frac{a_{j_1}^{(0,1)}}{\frac{1}{c_{j_1}} - \sum_{j_2=1, j_2 \neq j_1}^n \frac{a_{j_2}^{(0,1,j_1)}}{\frac{1}{c_{j_2}} - \dots}}},$$

in which  $a^{(0)} = (a_1^{(0)}, \dots, a_n^{(0)}) = (y, \dots, y)$

$a^{(0,1)} = (a_1^{(0,1)}, \dots, a_n^{(0,1)}) = (y + 2, y, \dots, y)$

$a^{(0,1,j_1, \dots, j_k)} = (a_1^{(0,1,j_1, \dots, j_k)}, \dots, a_n^{(0,1,j_1, \dots, j_k)}) = a^{(0,1,j_1, \dots, j_{k-1})} + 2(\delta_{ijk}, \dots, \delta_{ijk})$ .

From proposition 2, one obtains :

**Corollary 3** *The generating function of  $n$ -colored locally orientable rooted maps with root vertex of color  $i$ , is the solution of the following generalized Dyck equation :*

$$\widetilde{M}_{n,i} = y + c_i \widetilde{M}_{n,i} \sum_{j=1, j \neq i}^n c_j \widetilde{M}_j(u)$$

in which  $u = (u_j)_{1 \leq j \leq n} = (y + 2\delta_{ij})_{1 \leq j \leq n}$ .

## 7.4 Application to 3-colored locally orientable rooted maps

**Theorem 8**  $\forall i, j, k \in \{1, 2, 3\}$  with  $i, j, k$  all distinct, we have :

$$y + (c_i c_j \widetilde{M}_{3,j} + c_i c_k \widetilde{M}_{3,k} - 1) \widetilde{M}_{3,i} + 2c_i c_j^2 \frac{\partial \widetilde{M}_{3,i}}{\partial c_j} + 2c_i c_k^2 \frac{\partial \widetilde{M}_{3,i}}{\partial c_k} = 0.$$

**Corollary 4** *The generating function  $\widetilde{M}_{3,1}$  can be written in continued fraction form*

$$\begin{aligned} & \frac{y}{c_1 \left( \frac{1}{c_1 - \frac{1}{c_2 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+2}{c_1 - \frac{y}{c_3 - \dots}}}}}} \right.} \\ & \quad \left. \frac{y}{\frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+2}{c_3 - \dots}}}}} \right) \\ & - \frac{y}{\frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+2}{c_1 - \frac{y+2}{c_3 - \dots}}}}} \left( \frac{y}{\frac{1}{c_2 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+2}{c_3 - \dots}}}}} \right.} \\ & \quad \left. \left. \frac{y}{\frac{1}{c_3 - \frac{1}{c_1 - \frac{1}{c_2 - \frac{y+2}{c_1 - \dots}}}}} \right) \right) \end{aligned}$$

## 8 Conclusion

This work is a generalization of a work realized by D. Arquès and J.F. Béraud (see [2, 3]) on the enumeration with a continued fraction, of the generating function of rooted maps regardless of genus, with respect to vertices and edges.

We first wrote and proved the set of functional equation for the generating function of  $n$ -colored rooted maps in orientable and locally orientable surfaces, regardless of genus, with respect to vertices and half-edges whose initial vertex is of color  $i$ ,  $i \in \mathcal{I}_n$ .

To solve this set of equations, we exhibited a new set of equations for the generating function of  $n$ -colored rooted maps in orientable and locally orientable surfaces, regardless of genus, with respect to vertices of color  $i$ ,  $i \in \mathcal{I}_n$  and half-edges whose initial vertex is of color  $i$ ,  $i \in \mathcal{I}_n$ .

Then it has been shown that the solution of the partial differential equation verified by the generating function of  $n$ -colored rooted maps whose root vertex is of color  $i$ , in orientable and locally orientable surfaces, can be written with a continued fraction form. This enumeration with a continued fraction form leads to a generalization of the classical Dyck equation for rooted planar trees. It should be interesting to interpret topologically this generalized Dyck equation.

An interesting, but not trivial, extension of this work would be to give and solve the functional equation for  $n$ -colored rooted maps for a given genus.

## References

- [1] D. Arquès, Relations fonctionnelles et dénombrement des cartes pointées sur le tore, *J. Combin. Theory Ser. B* 43 (1987) 253-274.
- [2] D. Arquès and J.-F. Béraud, Rooted maps on orientable surfaces, Riccati's equation and continued fractions, *Discrete Math.* 215 (2000) 1-12.
- [3] D. Arquès and J.-F. Béraud, Rooted maps and hypermaps on surfaces, *Proceedings of FPSAC'99*, Barcelona, pp. 18-29, 1999.
- [4] D. Arquès and A. Giorgetti, Enumération des cartes pointées de genre quelconque en fonction des nombres de sommets et de faces, *J. Combin. Theory Ser. B* 77 (1999) 1-24.
- [5] E.A. Bender and E.R. Canfield, The number of rooted maps on an orientable surface, *J. Combin. Theory Ser. B*, 53 (1991), 293-299.
- [6] R. Cori, Un code pour les graphes planaires et ses applications, *Astérisque* 27 (1975).
- [7] I.P. Goulden and D.M. Jackson, Maps in locally orientable surfaces, the double coset algebra, and zonal polynomial, *Can. J. Math.* 48 (3) (1996) 569-584.
- [8] D.M. Jackson and T.I. Visentin, A character theoretic approach to embeddings of rooted maps in an orientable surface of given genus, *Trans. Amer. Math. Soc.* 322 (1990) 343-363.
- [9] D.M. Jackson and T.I. Visentin, Character theory and rooted maps in an orientable surface of given genus: face-colored maps, *Trans. Amer. Math. Soc.* 322 (1990) 365-376.
- [10] W.T. Tutte, A census of planar maps, *Canad. J. Math.* 15 (1963) 249-271.
- [11] W.T. Tutte, On the enumeration of planar maps, *Bull. Amer. Math. Soc.* 74 (1968) 64-74.
- [12] T.R.S Walsh, Hypermaps versus bipartite maps, *J. Combin. Theory Ser. B* 18 (1975) 155-163.
- [13] T.R.S Walsh and A.B. Lehman, Counting rooted maps by genus. I, *J. Combin. Theory Ser. B* 13 (1972) 192-218.

# On certain spaces of lattice diagram polynomials

Jean-Christophe Aval

Laboratoire A2X, Université Bordeaux 1

351 cours de la Libération, F-33405 Talence cedex

e-mail : [aval@math.u-bordeaux.fr](mailto:aval@math.u-bordeaux.fr)

## Abstract

The aim of this work is to study some lattice diagram polynomials  $\Delta_D(X, Y)$  as defined in [4] and to extend results of [3]. We recall that  $M_D$  denotes the space spanned by all partial derivatives of  $\Delta_D$ . In this paper, we want to study the space  $M_{i,j}^k(X, Y)$  which is the sum of  $M_D$  spaces where the lattice diagrams  $D$  are obtained by removing  $k$  cells from a given partition, these cells being in the “shadow” of a given cell  $(i, j)$  of the Ferrers diagram. We obtain an upper bound for the dimension of the resulting space  $M_{i,j}^k(X, Y)$ , that we conjecture to be optimal. These upper bounds allow us to construct explicit bases for the subspace  $M_{i,j}^k(X)$  consisting of elements of 0  $Y$ -degree.

## Résumé

Le but de ce travail est l'étude de certains polynômes  $\Delta_D(X, Y)$  associés à des diagrammes du réseau carré, introduits dans [4], et d'étendre des résultats obtenus dans [3]. Rappelons que  $M_D$  désigne l'espace engendré par toutes les dérivées partielles de  $\Delta_D$ . Dans cet article, nous voulons étudier l'espace  $M_{i,j}^k(X, Y)$  qui est défini comme la somme des espaces  $M_D$  où  $D$  décrit l'ensemble des diagrammes obtenus en enlevant  $k$  cases à une partition donnée, toutes ces cases étant dans l’“ombre” de  $(i, j)$  dans le diagramme de Ferrers. Nous obtenons une majoration pour la dimension de cet espace que nous conjecturons comme optimale. Cette majoration est exacte quand on se restreint au seul alphabet  $X$ , cas pour lequel nous construisons une base explicite.

## 1 Introduction

**Definition 1** *A lattice diagram is a finite subset of  $\mathbb{N} \times \mathbb{N}$ . For  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_k > 0$ , we say that  $\mu = (\mu_1, \mu_2, \dots, \mu_k)$  is a partition of  $n$  if  $n = \mu_1 + \dots + \mu_k$ . We associate to a partition  $\mu$  its Ferrers diagram  $\{(i, j) : 0 \leq i \leq k-1, 0 \leq j \leq \mu_{i+1} - 1\}$  and we use the symbol  $\mu$  for both the partition and its Ferrers diagram.*

Most definitions and conventions we use are similar to [4]. For example, given the

partition  $(4, 2, 1)$ , its partition diagram is

	2,0			
1,0		1,1		
0,0	0,1	0,2	0,3	

It consists of the lattice cells  $\{(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (2, 0)\}$ .

**Definition 2** Given a lattice diagram  $D = \{(p_1, q_1), (p_2, q_2), \dots, (p_n, q_n)\}$  we define the lattice determinant

$$\Delta_D(X; Y) = \det \|x_i^{p_j} y_i^{q_j}\|_{i,j=1}^n,$$

where  $X = X_n = \{x_1, x_2, \dots, x_n\}$  and  $Y = Y_n = \{y_1, y_2, \dots, y_n\}$ .

The polynomial  $\Delta_D(X; Y)$  is bihomogeneous of degree  $|p| = p_1 + \dots + p_n$  in  $X$  and of degree  $|q| = q_1 + \dots + q_n$  in  $Y$ . To insure that this definition associates a unique polynomial to  $D$  we require that the list of lattice cells be given with respect to the following “pseudo-lexicographic” order:  $\{(0, 0), (1, 0), (2, 0), (0, 1), (1, 1), (0, 2), (0, 3)\}$ .

For a polynomial  $P(X; Y)$ , the vector space spanned by all the partial derivatives of  $P$  of all orders is denoted  $\mathcal{L}_\partial[P]$ . A permutation  $\sigma \in S_n$  acts diagonally on a polynomial  $P(X; Y)$  as follows:  $\sigma P(X; Y) = P(x_{\sigma_1}, x_{\sigma_2}, \dots, x_{\sigma_n}; y_{\sigma_1}, y_{\sigma_2}, \dots, y_{\sigma_n})$ . Under this action,  $\Delta_D(X; Y)$  is clearly an alternant. It follows that for any lattice diagram  $D$  with  $n$  cells, the vector space  $M_D = \mathcal{L}_\partial[\Delta_D(X; Y)]$  is an  $S_n$ -module. Since  $\Delta_D(X; Y)$  is bihomogeneous, this module affords a natural bigrading.

The most general problem exposed in [4] and [5] concerns the space  $M_D$ . The main question is to decide whether this space is  $S_n$ -isomorphic to a sum of left regular representations or not. In the particular case where  $D$  corresponds to a partition  $\mu$  the question leads to the “ $n!$  conjecture” which asserts that the space  $M_\mu$  is a single copy of the left regular representation. Many efforts to prove this conjecture were only sufficient to obtain it in some special cases (see [1], [2], [7], [8] for example). In [5], the case where all the lattice cells of  $D$  lies on a single axis is solved.

The next class of lattice diagrams that is of interest is obtained by removing a single cell from a partition diagram. Its interest comes in part from the fact that it gives a possible recursive approach for the  $n!$  conjecture. If  $\mu$  is a partition of  $n+1$ , we denote by  $\mu/ij$  the lattice diagram obtained by removing the cell  $(i, j)$  from the Ferrers diagram of  $\mu$ . We refer to the cell  $(i, j)$  as the hole of  $\mu/ij$ . It is conjectured in [4] that the number of copies of the left regular representations in  $M_{\mu/ij}$  is equal to the cardinality (which we denote by  $s$ ) of the  $(i, j)$ -shadow, that is the cardinality of  $\{(i', j') \in \mu : i' \geq i, j' \geq j\}$ .

A study of the subspace  $M_{\mu/ij}(X)$  of  $M_{\mu/ij}$  consisting of elements of 0  $Y$ -degree can be found in [3], in which the corresponding “four term recursion” is proved by using the construction of explicit bases.

Here we study the following problem. Let  $\mu$  be a partition of  $n+k$ . This partition is fixed and does not appear in the following notations.

**Definition 3** Let  $M_{i,j}^k$  denote the following sum of vector spaces

$$M_{i,j}^k(X, Y) = \sum_{(a_1, b_1), \dots, (a_k, b_k)} M_{\mu/\{(a_1, b_1), \dots, (a_k, b_k)\}},$$

where the sum is over all the  $k$ -tuples of cells in the shadow of  $(i, j)$ .

Because of the “shift” operators (see [4], Proposition I.3 or section 2 in this paper) we have  $M_{\mu/ij} = M_{i,j}^1$ . Hence this space  $M_{i,j}^k$  is a possible generalization of  $M_{\mu/ij}$  if we want to make  $k$  holes in the Ferrers diagram. That is precisely this space that we want to study. In fact we obtain an upper bound for the dimension of this object, that we conjecture to be optimal.

In the second section we introduce some “shift” operators which are useful to move the holes and the cells in the diagrams. The third section is devoted to the proof of an upper bound for the dimension of  $M_{i,j}^k$  that is conjectured to be optimal. In the fourth section we study  $M_{i,j}^k(X)$ , the subspace of  $M_{i,j}^k(X, Y)$  consisting of elements of 0  $Y$ -degree, for which we obtain explicit bases.

## 2 The “shift” operators

For the sake of simplicity, we only state the following propositions for  $X$ -shifts. Of course similar results also hold for  $Y$ -shifts.

**Proposition 1** *Let  $L$  be a lattice diagram. Then for any integer  $k \geq 1$  we have*

$$p_k(\partial X)\Delta_L(X, Y) = \sum_{i=1}^n \epsilon(p_k(i; L))\Delta_{p_k(i; L)}(X, Y)$$

where  $p_k(i, L)$  is obtained by replacing the  $i$ -th biexponent  $(p_i, q_i)$  by  $(p_i - k, q_i)$  and the coefficient  $\epsilon(p_k(i; L))$  is different from zero only if the resulting diagram consists of  $n$  distinct cells in the positive quadrant. Its sign is the sign of the permutation that reorders the obtained biexponents in increasing pseudo-lexicographic order.

*Proof.* This is a particular case of Proposition I.1 in [4].

**Remark 1** The diagram  $p_k(i, L)$  is the diagram obtained by pushing down the  $i$ -th cell of  $L$ : its biexponent  $(p_i, q_i)$  is replaced by  $(p_i - k, q_i)$  which corresponds to  $k$  steps down. The other biexponents are unchanged. This duality between the subtractions on the set of biexponents and movements of cells of the diagram will be extensively employed all along this article, explicitly or implicitly.

**Proposition 2** *Let  $L$  be a lattice diagram. Then for any integer  $k \geq 1$  we have*

$$e_k(\partial X)\Delta_L(X, Y) = \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} \epsilon(e_k(i_1, \dots, i_k; L))\Delta_{e_k(i_1, \dots, i_k; L)}(X, Y)$$

where the diagram  $e_k(i_1, \dots, i_k; L)$  is obtained by replacing the biexponents  $(p_{i_1}, q_{i_1}), \dots, (p_{i_k}, q_{i_k})$  by  $(p_{i_1} - 1, q_{i_1}), \dots, (p_{i_k} - 1, q_{i_k})$  and where the coefficient  $\epsilon(e_k(i_1, \dots, i_k; L))$  is a nonnegative integer which is different from zero only when the resulting diagram consists of  $n$  distinct cells in the positive quadrant.

*Proof.* The proof is almost the same as for the previous proposition.

For a lattice diagram  $L$ , we denote by  $\bar{L}$  its complement in the positive quadrant (it is an infinite subset). Again we order  $\bar{L} = \{(\bar{p}_1, \bar{q}_1), (\bar{p}_2, \bar{q}_2), \dots\}$  using the pseudo-lexicographic order.

### Proposition 3

$$h_k(\partial X)\Delta_L(X; Y) = \sum_{1 \leq i_1 < i_2 < \dots < i_k} \epsilon(h_k(i_1, \dots, i_k, L))\Delta_{h_k(i_1, \dots, i_k, L)}(X; Y)$$

where  $h_k(i_1, \dots, i_k, L)$  is the lattice diagram with the following complement diagram. Replace the biexponents  $(\bar{p}_{i_1}, \bar{q}_{i_1}), \dots, (\bar{p}_{i_k}, \bar{q}_{i_k})$  of the complement  $\bar{L}$  with  $(\bar{p}_{i_1} + 1, \bar{q}_{i_1}), \dots, (\bar{p}_{i_k} + 1, \bar{q}_{i_k})$ . The coefficient  $\epsilon(h_k(i_1, \dots, i_k, L))$  is a nonnegative integer, different from zero only if the resulting complement consists of distinct cells in the positive quadrant.

*Proof.* We shall prove this proposition by induction on  $k$ . If  $k = 1$ , then  $h_1 = e_1$  and the result is true since moving down a cell is equivalent to moving up a hole. Assume the result is true up to  $k - 1$ . Then we use the fact that  $h_k = h_{k-1}e_1 - h_{k-2}e_2 + \dots + (-1)^k h_1 e_{k-1} + (-1)^{k+1} e_k$ . Indeed  $h_{k-1}$  pushes up  $(k - 1)$  different holes. And  $e_1$  moves down one cell, that is pushes up a hole. But if this hole has already been pushed up by  $h_{k-1}$ , we can view this situation as  $k - 2$  holes which have made one “step” and one which has made two steps. But this situation is killed by the term  $-h_{k-2}e_2$ . Thus by looking successively at the terms in the previous formula for  $h_k$ , we get what we want.

**Remark 2** One efficient application of the previous propositions appears in the case of two holes. They imply that for any couple of holes  $(h_1, h_2)$  in the shadow of  $(i, j)$  then  $\Delta_{\mu/\{h_1, h_2\}} \in M_{\mu/\{(i, j), (i, j+1)\}} + M_{\mu/\{(i, j), (i+1, j)\}}$  thus

$$M_{i,j}^2 = M_{\mu/\{(i, j), (i, j+1)\}} + M_{\mu/\{(i, j), (i+1, j)\}}.$$

The question of the generalization of the previous result when  $k \geq 3$  appears spontaneously. Is it sufficient to take only the diagrams such that the holes form a partition of origin  $(i, j)$ ? The answer is negative. For example it is easy to check that when  $\mu = (3, 2)$

$$\Delta_{\mu/\{(0,0), (1,0), (0,2)\}} \notin M_{\mu/\{(0,0), (1,0), (0,1)\}} + M_{\mu/\{(0,0), (0,1), (0,2)\}}.$$

### 3 The upper bound

We define the annulator ideal of a vector subspace  $M$  of  $\mathbb{Q}[X_n, Y_n]$  as the following ideal:  $I_M = \{P \in \mathbb{Q}[X_n, Y_n] : \forall Q \in M, P(\partial)Q = 0\}$ . If  $M = \mathcal{L}_\partial[P]$  then we denote its annulator ideal simply by  $I_P$ . In the case of  $M_{i,j}^k$  we denote  $I_{M_{i,j}^k}$  by  $I_{i,j}^k$ .

We recall the following important result ([7], Proposition 1.1):  $M = I_M^\perp$ , where the scalar product is defined by  $(P, Q) = L_0(P(\partial)Q)$  and where  $L_0$  is the linear form that associates to a polynomial its term of degree 0.

#### 3.1 About ideals

We want here to prove the following

##### Proposition 4

$$I_{i,j}^k = \bigcap_{(a_1, b_1), \dots, (a_k, b_k)} I_{\partial x_{n+1}^{a_1} \partial y_{n+1}^{b_1} \cdots \partial x_{n+k}^{a_k} \partial y_{n+k}^{b_k} \Delta_\mu} \cap \mathbb{Q}[X_n, Y_n] \stackrel{\text{def}}{=} \mathcal{I},$$

where the intersection is over the  $k$ -tuples of different cells in the shadow of  $(i, j)$  that we assume to be ordered in lexicographic order.

*Proof.* By expanding  $\Delta_\mu$  with respect to the last  $k$  columns, we obtain:

$$\begin{aligned}\Delta_\mu(X_{n+k}, Y_{n+k}) &= \sum_{(a_1, b_1), \dots, (a_k, b_k)} \pm \Delta_{\{(a_1, b_1), \dots, (a_k, b_k)\}}(\bar{X}_n, \bar{Y}_n) \\ &\quad \times \Delta_{\mu/\{(a_1, b_1), \dots, (a_k, b_k)\}}(X_n, Y_n),\end{aligned}$$

where  $\bar{X}_n = \{x_{n+1}, \dots, x_{n+k}\}$  and  $\bar{Y}_n = \{y_{n+1}, \dots, y_{n+k}\}$ . Thus for example:

$$(*) \quad \partial(x_{n+1}^{a_1} y_{n+1}^{b_1} \cdots x_{n+k}^{a_k} y_{n+k}^{b_k}) \Delta_\mu(X_{n+k}, Y_{n+k}) = c \Delta_{\mu/\{(a_1, b_1), \dots, (a_k, b_k)\}}(X_n, Y_n) + C$$

where  $c$  is a rational constant (different from 0) and  $C$  a linear combination with coefficients in  $\mathbb{Q}[x_{n+1}, y_{n+1}, \dots, x_{n+k}, y_{n+k}]$  of polynomials  $\Delta_{\mu/\{(a'_1, b'_1), \dots, (a'_k, b'_k)\}}(X_n, Y_n)$ , with  $(a'_j, b'_j)$  in the shadow of  $(a_j, b_j)$ . Observe that in this case any polynomial that kills  $\Delta_{\mu/\{(a_1, b_1), \dots, (a_k, b_k)\}}(X_n, Y_n)$  also kills every terms in  $C$ .

Hence we get what we want because:

- $\mathcal{I} \subset I_{i,j}^k$  follows from  $(*)$  by looking at the constant term of this equality in  $\mathbb{Q}[x_{n+1}, y_{n+1}, \dots, x_{n+k}, y_{n+k}]$ ;
- $I_{i,j}^k \subset \mathcal{I}$  follows directly from  $(*)$ .

### 3.2 Orbits

The reasoning is inspired from [4], Theorem 4.2.

Let  $\mu$  be a partition of  $n+k$ ,  $h = \mu_1$  and  $l = \mu'_1$ . We consider two sets  $\alpha = (\alpha_1, \dots, \alpha_h)$  and  $\beta = (\beta_1, \dots, \beta_l)$  of distinct complex numbers. To any injective tableau  $T$  of shape  $\mu$  with entries  $1, \dots, n+k$ , we associate a point  $(a(T), b(T))$  in  $\mathbb{C}^{2(n+k)}$  by the classical process, ie:  $a_i(T) = \alpha_{r_i(T)}$ ,  $b_i(T) = \beta_{c_i(T)}$  where  $r_i(T)$  (resp.  $c_i(T)$ ) is the number of the row (resp. column) of  $T$  where the entry  $i$  lies in  $T$ . We define  $\rho$  as the orbit of  $(a, b)$  when  $T$  varies over the  $(n+k)!$  injective tableaux of shape  $\mu$ . We introduce  $J_\rho$  the ideal of polynomials that are zero all over the orbit, and  $I = \text{gr } J_\rho$  and  $H = I^\perp$  (we recall that  $\text{gr}$  is an operator that associate to a polynomial its term of maximum degree and that the  $\text{gr}$  of an ideal is generated by the  $\text{gr}$  of its elements). It is now a classical result (cf. [7], Theorem 1.1) that  $I \subset I_{\Delta_\mu}$ .

We now look at another set in  $\mathbb{C}^{2n}$ . We consider the set of tableaux  $T$  of shape  $\mu$  with  $n$  entries and  $k$  white cells such that the  $k$  white cells are in the shadow of  $(i, j)$  (we denote this set of tableaux by  $\mathcal{T}_{i,j}^k$ ). By the same process as described above, we define an set  $\rho^k$  in  $\mathbb{C}^{2n}$ . Since the cardinality of  $\mathcal{T}_{i,j}^k$  is  $\binom{s}{k} n!$  the set  $\rho^k$  has this cardinality. We introduce  $J_{\rho^k}$  the ideal of polynomials that are zero all over  $\rho^k$ , and  $I^k = \text{gr } J_{\rho^k}$  and  $H^k = (I^k)^\perp$ . We have of course  $\dim H^k = \binom{s}{k} n!$ .

We want to prove that  $M_{i,j}^k \subset H^k$  and by [7], Proposition 1.1, it is equivalent to prove that  $I^k \subset \mathcal{I}$ .

### 3.3 Inclusion

We want here to obtain the next proposition:

**Proposition 5** *We have the inclusion:*

$$I^k \subset \mathcal{I}.$$

*Proof.* Let  $P$  be a polynomial in  $J_{\rho^k}$ . Let us consider

$$\begin{aligned} Q(X_{n+k}, Y_{n+k}) &= P(X_n, Y_n) \prod_{i'=1}^i (x_{n+1} - \alpha_{i'}) \cdots \prod_{i'=1}^i (x_{n+k} - \alpha_{i'}) \\ &\quad \times \prod_{j'=1}^j (y_{n+1} - \beta_{j'}) \cdots \prod_{j'=1}^j (y_{n+k} - \beta_{j'}). \end{aligned}$$

We want to check that this polynomial is an element of  $J_\rho$ . We take an element  $(\alpha, \beta)$  of  $\rho$ . If its projection on  $\mathbb{C}^{2n}$  is in  $\rho^k$  then  $Q(\alpha, \beta) = 0$  because of  $P$ . If not it must have at least one entry between  $n+1$  and  $n+k$  in the first  $i$  rows or the first  $j$  columns and we have still  $Q(\alpha, \beta) = 0$ . Thus  $\text{gr}(P) \in I_{\partial x_{n+1}^i \partial y_{n+1}^j \cdots \partial x_{n+k}^i \partial y_{n+k}^j \Delta_\mu}$ .

For any set of  $k$  cells  $\{(a_1, b_1), \dots, (a_k, b_k)\}$  in the shadow of  $(i, j)$ , we observe that  $\forall h, 1 \leq h \leq k, a_h \geq i$  and  $b_h \geq j$ . Hence  $\text{gr}(P)$  is in  $\mathcal{I}$ , which was to be proved.

### 3.4 Conclusion

The main result is now a consequence of all what precedes:

**Theorem 1** *If  $\mu$  is a partition of  $n+k$  and  $s$  the cardinal of the shadow of the cell  $(i, j)$ , then we have:*

$$\dim M_{i,j}^k \leq \binom{s}{k} n!.$$

**Remark 3** Recall the proof of Theorem 1.1 of [7]: the previous reasoning implies that if equality holds in Theorem 1, then  $M_{i,j}^k$  decomposes as  $\binom{s}{k}$  times the left regular representation.

Numerical examples and the fact that the construction described in the previous subsection affords the “good” upper bound in the case of one set of variables (see the next section) support the following conjecture, which was first stated by F. Bergeron.

**Conjecture 1** *With the notations of the previous theorem:*

$$\dim M_{i,j}^k = \binom{s}{k} n!.$$

**Remark 4** When  $k = 1$ , this conjecture reduces to Conjecture I.2 of [4] and when  $s = k$  or  $k = 0$  to the  $n!$  conjecture.

## 4 Case of one set of variables

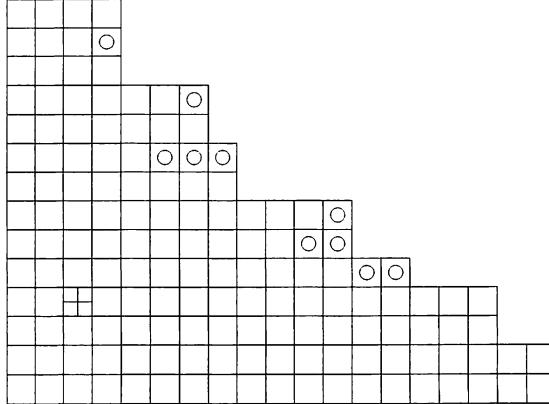
The goal of this section is to obtain an explicit basis for  $M_{i,j}^k(X)$ , ie the subspace of  $M_{i,j}^k(X, Y)$  consisting of elements of 0  $Y$ -degree.

### 4.1 Construction

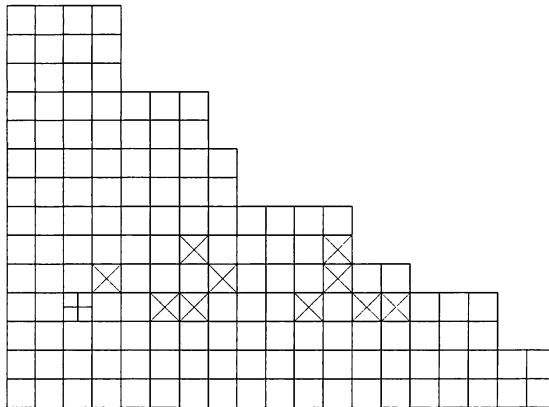
We recall that in [2] is constructed a basis for  $M_\mu^0$  made of monomial derivatives  $M_S(\partial) \Delta_\mu$  of  $\Delta_\mu$ , where the objects  $S$  varies over a set  $\mathcal{S}(\mu)$  which depends on  $\mu$ . The corresponding construction can be found in [2]. It is not necessary to know the details of the

construction, just recall that we have a basis for  $M_\mu^0$  of the form:  $\{M_S(\partial)\Delta_\mu, S \in \mathcal{S}(\mu)\}$ . The cardinality of  $\mathcal{S}(\mu)$  is equal to  $n!/\mu!$  where  $\mu! = \mu_1!\mu_2!\cdots\mu_k!$ . This cardinality is the number of injective, row-increasing tableaux of shape  $\mu$ .

Then we choose in the Ferrers diagram  $\mu$   $k$  cells which are simultaneously in the shadow of  $(i, j)$  and on the right edges of  $\mu$ . We denote by  $\mathcal{F}_\mu^k$  the set of these objects. In the next figure, the chosen cells are cells with a circle and in the cell  $(i, j)$  appears a + sign. In this example  $n = 142$  and  $k = 10$ .



Then to an object  $F$  in  $\mathcal{F}_\mu^k$  we associate  $\mu_F$  the partition of  $n$  obtained by pushing up the circled cells and by removing the corresponding cells. We also define a partition  $\mu_F^k$  with  $k$  holes as follows. We look at the circled cells from the right to the left and from bottom to top. For a circled cell in column  $j' \geq j$  such that there are  $l$  possible places where this circle could have been placed in the same column and under this one, we put a hole in the cell  $(i + l, j')$ . The following figure illustrates this construction for the previous  $F$  (the holes are as usual cells with crosses:  $\times$ ).



The object of the end of the article is to prove the following theorem.

**Theorem 2** *With the previous notations*

$$B_{i,j}^k(X) = \{M_S(\partial)\Delta_{\mu_F^k}; S \in \mathcal{S}(\mu_F), F \in \mathcal{F}_\mu^k\}$$

*is a basis for  $M_{i,j}^k(X)$ .*

In the next paragraphs, the proofs are omitted or briefly described, because either too long or a bit technical.

## 4.2 Upper bound

**Definition 4** We denote by  $\mathcal{T}_{i,j}^k$  the set of injective, row-increasing tableaux of shape  $\mu$  with entries  $1, \dots, n$  (whence each of these tableaux has  $k$  white cells) such that the  $k$  white cells are in the shadow of  $(i, j)$ .

**Proposition 6** We have this upper bound for the dimension of  $M_{i,j}^k(X)$ :

$$\dim M_{i,j}^k(X) \leq \#\mathcal{T}_{i,j}^k.$$

## 4.3 Cardinality

We claim that:

**Proposition 7** We have the following equality

$$\#B_{i,j}^k(X) = \#\mathcal{T}_{i,j}^k.$$

## 4.4 Independence

We want here to conclude by proving that

**Proposition 8** The family  $B_{i,j}^k(X)$  is linearly independent. Thus in particular equality holds in Proposition 6.

*Proof.* Assume that we have a non-trivial dependence relation.

We define the depth of a hole to be the number of cells (different from holes) that are above this hole. We look at the  $k$ -tuples of the depths of the  $k$  holes of the  $\mu_F^k$ :  $(d_1 \leq d_2 \leq \dots \leq d_k)$ . The crux of the proof is the following result:

**Lemma 1** The  $k$ -tuples  $(d_1, d_2, \dots, d_k)$  are all distinct.

Then if we have a non-trivial dependence relation, we consider the greatest  $k$ -tuple of depths with respect to the lexicographic order that appear in this relation (relative to an object  $F^0$ ):  $(d_1^0, d_2^0, \dots, d_k^0)$ . We then apply the differential operator  $h_k(\partial)^{d_1^0} \cdot h_k(\partial)^{d_2^0 - d_1^0} \dots h_1(\partial)^{d_k^0 - d_{k-1}^0}$  to the dependence relation. It kills all the terms but those which come from the single object  $F^0$ . These terms give in fact terms in  $B = \{M_S(\partial). \Delta_{\mu_{F^0}}; S \in \mathcal{S}(\mu_{F^0})\}$ , which are independent since  $B$  is a basis of  $M_{\mu_{F^0}}^0$ .

**Acknowledgement.** The author would like to thank François Bergeron for indicating him this problem and for numerous valuable suggestions.

## References

- [1] E. Allen, *The decomposition of a bigraded left regular representation of the diagonal action of  $S_n$* , J. Comb. Theory A, **71** (1995), 97-111.
- [2] J.-C. Aval, *Monomial bases related to the  $n!$  conjecture*, to appear in Discrete Mathematics.

- [3] J.-C. Aval, F. Bergeron and N. Bergeron, *On certain spaces of quasi-harmonic polynomials*, in preparation.
- [4] F. Bergeron, N. Bergeron, A. Garsia, M. Haiman and G. Tesler, *Lattice Diagram Polynomials and Extended Pieri Rules*, Adv. Math., **142** (1999), 244-334.
- [5] F. Bergeron, A. Garsia and G. Tesler, *Multiple Left Regular Representations Associated with Alternants of the Symmetric Group*, in preparation.
- [6] N. Bergeron and A. Garsia, *On certain spaces of harmonic polynomials*, Contemp. Math., **138** (1992), 51-86.
- [7] A. Garsia and M. Haiman, *Some natural bigraded  $S_n$ -modules and  $q, t$ -Kostka coefficients*, Elec. J. Combin. **3**, No 2 (1996) [Paper 24, 60 pp.].
- [8] A. Garsia and M. Haiman, *Orbit Harmonics and Graded Representations*, to appear in Les éditions du Lacim.

# Inequalities for Eulerian posets via doubling

Margaret M. Bayer\* and Gábor Hetyei†

Department of Mathematics  
University of Kansas  
Lawrence KS 66045-2142

## Abstract

We show a simple and efficient way to generate inequalities holding for the flag  $f$ -vector of an Eulerian poset from the inequalities holding for the flag vector of an arbitrary graded poset. The method allows us to generate all facet inequalities of the cone of flag  $f$ -vectors of Eulerian posets up to rank 7. It also gives a complete description of the cone of flag  $f$ -vectors of “locally wide” posets, which are a generalization to Eulerian posets.

## Résumé

Nous montrons une méthode simple et efficace pour engendrer des inégalités linéaires valables pour le vecteur  $f$  drapeau d'un ensemble partiellement ordonné eulérien, à partir des inégalités valables pour le vecteur  $f$  drapeau d'un ensemble partiellement ordonné quelconque. Cette méthode nous permet d'engendrer toutes les inégalités correspondant à une face maximale du cône des vecteurs  $f$  drapeau des ensembles partiellement ordonnés “localement larges”, ce qui sont une généralisation des ensembles partiellement ordonnés eulériens.

## 1 Introduction

The study of Eulerian partially ordered sets (posets) originated with Stanley ([11]). Examples of Eulerian posets are the posets of faces of regular CW spheres. These include face lattices of convex polytopes, the Bruhat order on finite Coxeter groups, and the lattices of regions of oriented matroids. (See [7] and [8].)

The flag  $f$ -vector (or simply flag vector) of a poset is a standard parameter counting chains in the partially ordered set by ranks. In the last twenty years there has grown a body of work on numerical conditions on flag vectors of posets and complexes, especially those arising in geometric contexts. Early contributions are from Stanley on balanced Cohen-Macaulay complexes ([10]) and Bayer and Billera on the linear equations on flag vectors of Eulerian posets ([1]). A major recent contribution is the determination of the closed cone of flag vectors of all graded posets by Billera and Hetyei ([4]). Results on flag vectors and other invariants of Eulerian posets and special classes of them are surveyed in [12].

Currently we are working on describing the closed cone of flag vectors of Eulerian partially ordered sets. This problem was posed explicitly in [6]. In a recent paper [4]

---

\*This research was supported by University of Kansas General Research allocation #3552.

†On leave from the Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences. Partially supported by Hungarian National Foundation for Scientific Research grant no. F 032325.

we have described this cone up to rank 7. The extreme rays of the cone were obtained from sequences of half-Eulerian posets via replacing each element in the poset with a pair of elements. We call this operation *horizontal doubling*. (Half-Eulerian posets are defined by their horizontal double being Eulerian.) We made the conjecture that the same construction will yield the entire cone of flag vectors of Eulerian posets in arbitrary rank.

This conjecture is still open, but now we have gone an important step further. Half-Eulerian posets up to rank 7 have the remarkable property that every linear inequality holding for their flag vectors is equivalent modulo the half-Eulerian relations to an inequality holding for all graded posets. If this was true in arbitrary rank then the following theorem would imply the conjecture.

**Theorem 1.1** *If an inequality of the form  $\sum_{S \subseteq [1,n]} a_S f_S(P) \geq 0$  holds for every graded poset of rank  $n+1$ , then  $\sum_{S \subseteq [1,n]} a_S 2^{n-|S|} f_S(P) \geq 0$  holds for every Eulerian poset of the same rank.*

An outline of the proof of this result is given in section 5. All we use about Eulerian posets is that the *width* of every interval of rank at least 2 is 2 or more. The cone of flag  $f$ -vectors for this class of graded posets may be explicitly described with a little addition to Theorem 1.1. Analogous results may be stated for partially ordered sets whose intervals of rank at least 2 have width at least  $k$ , where  $k$  is an arbitrary constant.

## 2 Preliminaries

### 2.1 Graded posets

A *graded poset*  $P$  is a finite partially ordered set with a unique minimum element  $\hat{0}$ , a unique maximum element  $\hat{1}$ , and a *rank function*  $\rho : P \rightarrow \mathbf{N}$  satisfying  $\rho(\hat{0}) = 0$ , and  $\rho(y) - \rho(x) = 1$  whenever  $y \in P$  covers  $x \in P$ . The *rank*  $\rho(P)$  of a graded poset  $P$  is the rank of its maximum element. Given a graded poset  $P$  of rank  $n+1$  and a subset  $S$  of  $\{1, 2, \dots, n\}$  (which we abbreviate as  $[1, n]$ ), define the  *$S$ -rank-selected subposet of  $P$*  to be the poset

$$P_S := \{x \in P : \rho(x) \in S\} \cup \{\hat{0}, \hat{1}\}.$$

Denote by  $f_S(P)$  the number of maximal chains of  $P_S$ . Equivalently,  $f_S(P)$  is the number of chains  $x_1 < \dots < x_{|S|}$  in  $P$  such that  $\{\rho(x_1), \dots, \rho(x_{|S|})\} = S$ . The vector  $(f_S(P) : S \subseteq [1, n])$  is called the *flag  $f$ -vector* of  $P$ . Whenever it does not cause confusion, we write  $f_{s_1 \dots s_k}$  rather than  $f_{\{s_1, \dots, s_k\}}$ ; in particular,  $f_{\{m\}}$  is always denoted  $f_m$ .

### 2.2 Eulerian posets

The *Möbius function* of a graded poset  $P$  is defined recursively for any subinterval of  $P$  by the formula

$$\mu([x, y]) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \leq z < y} \mu([x, z]) & \text{otherwise.} \end{cases}$$

A graded poset  $P$  is *Eulerian* if the Möbius function of every interval  $[x, y]$  is given by  $\mu([x, y]) = (-1)^{\rho(x, y)}$ . (Here  $\rho(x, y) = \rho([x, y]) = \rho(y) - \rho(x)$ .)

The first characterization of all linear equalities holding for the flag vectors of all Eulerian posets was given by Bayer and Billera in [1].

**Theorem 2.1 (Bayer and Billera)** *Every linear equality holding for the flag vector of all Eulerian posets of rank  $n + 1$  is a consequence of the equalities*

$$((-1)^{i-1} + (-1)^{k+1}) f_S + \sum_{j=i}^k (-1)^j f_{S \cup \{j\}} = 0$$

for  $S \subseteq [1, n]$  and  $[i, k]$  a maximal interval of  $[1, n] \setminus S$ .

### 2.3 Locally wide posets

Given a graph  $G$  with two distinguished vertices  $u$  and  $v$ , the *width* of  $G$  is the size of the smallest cutset separating  $u$  and  $v$ . Applying this notion to the Hasse-diagram of a graded poset  $P$  of rank at least 2 and its distinguished vertices  $\hat{0}$  and  $\hat{1}$ , the width of a graded poset is the size of the smallest maximal antichain. We call a graded poset *wide* if its width is at least two, and we call it *locally wide* if every interval  $[u, v]$  of rank at least two has width at least two. It is easy to show that a graded poset is locally wide if every interval  $[u, v]$  of rank 2 has at least four elements, and that the class of locally wide posets is closed under rank selection. Since every interval of rank two in an Eulerian poset has exactly four elements, Eulerian posets form a subclass of locally wide posets.

## 3 Flag vectors of arbitrary graded posets

An *interval system* on  $[1, n]$  is any set of subintervals of  $[1, n]$  that form an antichain (that is, no interval is contained in another). We say that a set  $S \subseteq [1, n]$  *blocks* the interval system  $\mathcal{I}$  if it has a nonempty intersection with every  $I \in \mathcal{I}$ . The family of all subsets of  $[1, n]$  blocking  $\mathcal{I}$  is denoted by  $\mathbf{B}_{[1, n]}(\mathcal{I})$ . The main result of [4] is the following.

**Theorem 3.1** *An expression  $\sum_{S \subseteq [1, n]} a_S f_S(P)$  is nonnegative on all graded posets  $P$  of rank  $n + 1$  if and only if we have*

$$\sum_{S \in \mathbf{B}_{[1, n]}(\mathcal{I})} a_S \geq 0 \quad \text{for every interval system } \mathcal{I} \text{ on } [1, n]. \quad (1)$$

The proof of the necessity of the condition (1) involves constructing for every interval system  $\mathcal{I}$  on  $\{1, 2, \dots, n\}$  a family of posets  $\{P(n, \mathcal{I}, N) : N \in \mathbb{N}\}$  of rank  $n + 1$  such that we have

$$\lim_{N \rightarrow \infty} \frac{1}{f_{[1, n]}(P(n, \mathcal{I}, N))} \sum_{S \subseteq [1, n]} a_S f_S(P(n, \mathcal{I}, N)) = \sum_{S \in \mathbf{B}_{[1, n]}(\mathcal{I})} a_S.$$

The difficult part of the proof of Theorem 1.1 necessitates revisiting the proof of the other implication of Theorem 3.1.

Let  $P$  be an arbitrary graded poset, and assume that we have drawn its Hasse-diagram in the plane. Given an interval  $[p, q]$  of  $P$ , let  $\phi(p, q)$  denote the leftmost atom in  $[p, q]$ . (If  $q$  covers  $p$  then we set  $\phi(p, q) := q$ .) The operation  $\phi$  has the following crucial property:

$$\text{If } p \in [x, y] \subseteq [x, z] \text{ and } p = \phi([x, z]) \text{ then } p = \phi([x, y]). \quad (2)$$

For every  $S \subseteq [1, n]$  and  $i \in [1, n]$  we define  $M_S(i)$  to be the smallest  $j \in [i, n+1]$  such that  $j \in S \cup \{n+1\}$ . Consider the set of maximal chains

$$F_S := \{\hat{0} = p_0 < p_1 < \cdots < p_n < p_{n+1} = \hat{1} : \forall i \in [1, n] (p_i = \phi([p_{i-1}, p_{M_S(i)}]))\}.$$

It is easy to verify that  $F_S$  contains exactly  $f_S(P)$  elements. Moreover, there is a way of associating a family of intervals  $\mathcal{I}_C$  to every maximal chain  $C := \{\hat{0} = p_0 < p_1 < \cdots < p_n < p_{n+1} = \hat{1}\}$  such that  $C$  belongs to  $F_S$  if and only if  $S$  blocks  $\mathcal{I}_C$ . (This family is not necessarily an antichain, but  $S$  blocks a family of intervals if and only if it blocks the antichain of the minimal intervals in this family.) The fact that one may find such a family of intervals is a direct consequence of property (2).

## 4 Half-Eulerian posets and their Eulerian horizontal doubles

The *horizontal double*  $D(P)$  of a graded poset  $P$  is obtained by replacing every  $x \in P \setminus \{\hat{0}, \hat{1}\}$  with two elements  $x_1, x_2$  such that  $\hat{0}$  and  $\hat{1}$  remain the minimum and maximum elements of the partially ordered set, and  $x_i < y_j$  if and only if  $x < y$  in  $P$ . (In the Hasse diagram of  $P$ , every edge is replaced by  $\bowtie$ .) The flag vectors of  $P$  and its horizontal double are connected by the formula  $f_S(DP) = 2^{|S|} f_S(P)$ .

**Definition 1** A *half-Eulerian poset* is a graded partially ordered set whose horizontal double is Eulerian.

Applying the definition of Eulerian property to the horizontal double of a poset we get

**Proposition 4.1** A graded partially ordered set  $P$  is half-Eulerian if and only if for every interval  $[x, y]$  of  $P$ ,

$$\sum_{i=1}^{\rho(x,y)-1} (-1)^{i-1} f_i([x, y]) = (1 + (-1)^{\rho(x,y)})/2.$$

The analogue of the Bayer-Billera theorem holds for half-Eulerian posets, and it is easy to show that the vector space generated by the flag  $f$ -vectors of half-Eulerian posets of given rank has the same dimension as its Eulerian analogue. In other words, the vector space automorphism

$$D_n : (f_S : S \subseteq [1, n]) \mapsto (2^{|S|} f_S : S \subseteq [1, n])$$

maps the subspace generated by the flag vectors of half-Eulerian posets of rank  $n+1$  onto the subspace generated by the flag vectors of Eulerian posets. The question naturally arises, whether the closed cone generated by the flag vectors of Eulerian posets of given rank is also sent by  $D_n$  onto the closed cone of flag  $f$ -vectors of Eulerian posets.

In [3] we have constructed the extreme rays of the cone of the flag vectors of half-Eulerian posets up to rank 7 as limits of normalized flag vectors of sequences of graded posets. We obtained the facets of the cones using PORTA [9]. (The documentation for our use of PORTA is available online at [2].) They turn out to represent inequalities holding for the flag vector of an arbitrary poset (and that's how we saw that the cone of flag vectors we generated is the entire cone).

**Conjecture 4.2** Every linear inequality holding for the flag vectors of all half-Eulerian posets of rank  $n+1$  is equivalent modulo the half-Eulerian analogue of the Bayer-Billera relations to a linear inequality that holds for the flag vector of an arbitrary poset.

Our paper [3] verifies Conjecture 4.2 holds for  $n \leq 6$ . Straightforward substitution into the definition of  $D_n$  shows the following.

**Proposition 4.3** If

$$\sum_{S \subseteq [1,n]} a_S f_S(P) \geq 0$$

is a facet inequality of the closed cone of half-Eulerian posets of rank  $n+1$  then

$$\sum_{S \subseteq [1,n]} a_S 2^{n-|S|} f_S(Q) \geq 0$$

is a facet inequality of the closed cone generated by the doubles of half-Eulerian posets of rank  $n+1$ .

Hence, Theorem 1.1 implies that for  $n \leq 6$  the closed cone of flag vectors of Eulerian posets of rank  $n+1$  equals the closed cone of flag vectors of doubles of half-Eulerian posets of the same rank. (In [3] we had to verify the validity of the Eulerian facet inequalities individually.) Moreover, we have the following.

**Corollary 4.4** If Conjecture 4.2 holds for rank  $n+1$  then every linear inequality holding for the flag vector of every Eulerian poset of rank  $n+1$  is the consequence of the following set of equations and inequalities:

- (i) the Bayer-Billera relations, and
- (ii) every inequality of the form  $\sum_{S \subseteq [1,n]} a_S 2^{n-|S|} f_S(Q) \geq 0$  where  $\sum_{S \subseteq [1,n]} a_S f_S(P) \geq 0$  is an inequality holding for the flag vector of every graded poset  $P$  of rank  $n+1$ .

Evidently, it is sufficient to list the facet-inducing inequalities under (ii).

## 5 The main theorem

Theorem 1.1 is a consequence of the following slightly stronger statement.

**Theorem 5.1** The inequality  $\sum_{S \subseteq [1,n]} a_S f_S(P) \geq 0$  holds for every graded poset  $P$  of rank  $n+1$  if and only if  $\sum_{S \subseteq [1,n]} a_S 2^{n-|S|} f_S(Q) \geq 0$  holds for every locally wide poset  $Q$  of the same rank.

The “if” part is a straightforward consequence of the fact that the horizontal double of an arbitrary graded poset is locally wide. For the proof of the other implication, let us assume that  $\sum_{S \subseteq [1,n]} a_S f_S(P) \geq 0$  holds for every graded poset  $P$  of rank  $n+1$ . We may use a modified version of the proof of Theorem 3.1. As in that proof we start out by drawing an arbitrary locally wide graded poset  $Q$  of rank  $n+1$  into the plane.

Given an interval  $[p, q]$  of  $Q$ , let  $\phi(p, q)$  denote now the set of the leftmost and second leftmost atoms atom in  $[p, q]$ . (If  $q$  covers  $p$  then we set  $\phi(p, q) := \{q\}$ .) In analogy with (2), the operation  $\phi$  now satisfies the following

$$\text{If } p \in [x, y] \subseteq [x, z] \text{ and } p \in \phi([x, z]) \text{ then } p \in \phi([x, y]). \quad (3)$$

We proceed by modifying the definition of the sets  $F_S$ , and let

$$F_S := \{\hat{0} = p_0 < p_1 < \cdots < p_n < p_{n+1} = \hat{1} : \forall i \in [1, n] (p_i \in \phi([p_{i-1}, p_{M_S(i)}]))\}.$$

It may be verified that  $F_S$  has now  $2^{n-|S|} f_S(Q)$  elements. Furthermore, property (3) may be used to prove that one may associate a family of intervals  $\mathcal{I}_C$  to an arbitrary maximal chain

$$C := \{\hat{0} = p_0 < p_1 < \cdots < p_n < p_{n+1} = \hat{1}\}$$

such that  $C$  belongs to  $F_S$  if and only if  $S$  blocks  $\mathcal{I}_C$ . Given a system of intervals  $\mathcal{I}$  let us denote by  $f_{\mathcal{I}}$  the number of those maximal chains  $C$  of  $Q$  for which the antichain of minimal intervals of  $\mathcal{I}_C$  is  $\mathcal{I}$ . (Note that  $f_{\mathcal{I}}$  depends not only on  $Q$  but also on the way we drew it in the plane!) We have

$$\begin{aligned} \sum_{S \subseteq [1, n]} a_S 2^{n-|S|} f_S(Q) &= \sum_{S \subseteq [1, n]} a_S |F_S| = \sum_{S \subseteq [1, n]} a_S \sum_{S \in \mathbf{B}_{[1, n]}(\mathcal{I})} f_{\mathcal{I}} \\ &= \sum_{\mathcal{I}} f_{\mathcal{I}} \sum_{S \in \mathbf{B}_{[1, n]}(\mathcal{I})} a_S. \end{aligned}$$

By Theorem 3.1 the sums  $\sum_{S \in \mathbf{B}_{[1, n]}(\mathcal{I})} a_S$  are all nonnegative, and so we have

$$\sum_{S \subseteq [1, n]} a_S 2^{n-|S|} f_S(Q) \geq 0.$$

## 6 Generalizations and $q$ -Eulerian posets

Theorem 5.1 may be generalized in many ways. If we restrict ourselves to those graded posets of rank  $n+1$ , for which every interval of rank 2 has width at least  $k$ , an inequality of the form  $\sum_{S \subseteq [1, n]} a_S k^{n-|S|} f_S(Q) \geq 0$  holds for all such posets  $Q$  if and only if  $\sum_{S \subseteq [1, n]} a_S f_S(P) \geq 0$  holds for all graded posets  $P$  of rank  $n+1$ . We can go even further and prescribe the width of an interval  $[u, v]$  of rank 2 depending on the rank of  $u$ .

**Definition 2** Let  $\underline{k} := (k_1, \dots, k_n)$  be a vector of positive integers. We say that a graded poset  $P$  of rank  $n+1$  is locally  $\underline{k}$ -wide if the width of every interval  $[u, v] \subseteq P$  with  $u$  of rank  $i-1$  and  $v$  of rank  $i+1$  is at least  $k_i$ .

Theorem 5.1 has the following generalization.

**Theorem 6.1** Let  $\underline{k} := (k_1, \dots, k_n)$  be a vector of positive integers. An inequality of the form

$$\sum_{S \subseteq [1, n]} a_S \prod_{t \in [1, n] \setminus S} k_t f_S(Q) \geq 0$$

holds for all locally  $\underline{k}$ -wide posets  $Q$  of rank  $n+1$  if and only if

$$\sum_{S \subseteq [1, n]} a_S f_S(P) \geq 0$$

holds for all graded posets  $P$  of rank  $n+1$ .

The proof is the same as that of Theorem 5.1 except that  $\phi(u, v)$  will denote the set of first  $k_i$  atoms, if the rank of  $u$  is  $i - 1$ .

The interest of this generalization is the following. Let  $\underline{q} := (q_1, \dots, q_n)$  be a sequence of real numbers. We may define a  $\underline{q}$ -analogue of the Möbius function on the class of intervals of graded posets of rank  $n + 1$  by

$$\mu_{\underline{q}}([x, y]) := \begin{cases} 1 & \text{if } x = y, \\ -\mu_{\underline{q}}([x, x]) - \sum_{x < z < y} \frac{1}{q_{\rho(z)}} \mu_{\underline{q}}([x, z]) & \text{otherwise.} \end{cases}$$

We call a rank  $n + 1$  graded poset  $\underline{q}$ -Eulerian, if every interval  $[x, y]$  in it satisfies  $\mu_{\underline{q}}([x, y]) = (-1)^{\rho(x, y)}$ .

Given a sequence of positive integers  $\underline{k} = (k_1, \dots, k_n)$  and a rank  $n + 1$  half-Eulerian poset  $P$ , we may construct a  $(\frac{k_1}{2}, \dots, \frac{k_n}{2})$ -Eulerian poset by applying the following generalization of the horizontal doubling operation. Let  $D_{\underline{k}}(P)$  be the graded poset obtained by replacing each  $x \in P \setminus \{\hat{0}, \hat{1}\}$  of rank  $s$  with  $k_s$  copies  $x_1, \dots, x_{k_s}$  and by setting  $x_i < y_j$  if and only if  $x < y$  in  $P$ . On the other hand, every interval  $[x, y]$  of a  $\underline{q}$ -Eulerian poset satisfying  $\rho(x) = i - 1$  and  $\rho(y) = i + 1$  must have width  $2q_i$ , showing that  $\underline{q}$ -Eulerian posets exist exactly for those vectors  $\underline{q}$  in which each  $q_i$  is the half of a positive integer. For such vectors  $\underline{q}$  one may show that the  $\underline{q}$ -analogue of the Bayer-Billera theorem holds, and that the vector space of flag vectors of  $\underline{q}$ -Eulerian posets is spanned by the flag vectors of posets of the form  $D_{\underline{q}}(\bar{P})$  where  $\bar{P}$  is an arbitrary half-Eulerian poset.

Hence Conjecture 4.2, together with Theorem 6.1, implies at once the description of the closed cone of flag vectors of  $\underline{q}$ -Eulerian posets for every vector  $\underline{q}$  consisting of halves of positive integers.

## References

- [1] M. M. Bayer and L. J. Billera, Generalized Dehn–Sommerville relations for polytopes, spheres and Eulerian partially ordered sets, *Invent. Math.* **79** (1985), 143–157.
- [2] M. M. Bayer and G. Hetyei, The cone of flag vectors of Eulerian posets up to rank 7, <http://www.math.ukans.edu/~bayer/Eulerian> (April 2000).
- [3] M. M. Bayer and G. Hetyei, Flag vectors of Eulerian partially ordered sets, preprint 1999.
- [4] L. J. Billera and G. Hetyei, Linear inequalities for flags in graded partially ordered sets, *J. Combinatorial Theory, Ser. A* **89** (2000), 77–104.
- [5] L. J. Billera and G. Hetyei, Decompositions of partially ordered sets, preprint.
- [6] L. J. Billera and N. Liu, Noncommutative enumeration in graded posets, to appear in *J. Algebraic Combin.*
- [7] A. Björner, Posets, regular CW complexes and Bruhat order, *European J. Combin.* **5** (1984), 7–16.
- [8] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, G. Ziegler, “Oriented Matroids,” Cambridge University Press, Cambridge, 1993.
- [9] T. Christof and A. Loebel, PORTA—A Polyhedron Representation Transformation Algorithm, elib.zib.de, 1997

- [10] R. P. Stanley, Balanced Cohen-Macaulay complexes, *Trans. Amer. Math. Soc.*, **249** (1979), 139–157.
- [11] R. P. Stanley, Some aspects of groups acting on finite posets, *J. Combinatorial Theory, Ser. A* **32** (1982), 132–161.
- [12] R. P. Stanley, A survey of Eulerian posets, in: “Polytopes: Abstract, Convex, and Computational,” T. Bisztriczky, P. McMullen, R. Schneider, A. I. Weiss, eds., NATO ASI Series C, vol. 440, Kluwer Academic Publishers, 1994, pages 301–333.

# Fast Implementations of Automata Computations

Anne Bergeron\* and Sylvie Hamel

## Abstract

In [M99], G. Myers describes a bit-vector algorithm to compute the edit distance between strings. The algorithm converts an input sequence to an output sequence in a parallel way, using bit operations readily available in processors.

In this paper, we generalize the technique, and characterize a class of automata for which there exists equivalent parallel, or *vector*, algorithms. As an application, we extend Myers result to arbitrary weighted edit distances, which are currently used to explore the vast data-bases generated by genetic sequencing.

## Résumé

G. Myers a récemment décrit un algorithme basé sur les vecteurs de bits pour le calcul de la distance d'édition entre deux chaînes [M99]. Cet algorithme convertit une chaîne d'entrée de taille  $n$  en une chaîne de sortie de taille  $n$  d'une façon parallèle, en utilisant des opérations standard sur les vecteurs de bits.

Dans cet article, nous généralisons cette technique et nous donnons une caractérisation d'une classe d'automates pour lesquels il existe un algorithme parallèle équivalent. En guise d'application, nous donnons une extension du résultat de Myers au calcul de distances valuées entre deux chaînes. Ce type de distance est couramment employé pour l'exploration des immenses banques de données générées par les projets de séquençage génétique.

## 1 Introduction

Finite automaton are powerful devices for computing on sequences of characters. Among the finest examples, very elegant linear algorithms have been developed for the string matching problem [A90]. Automata are also widely used in fields such as metric lexical analysis [CSY99] or bio-computing, where *approximate string matching* is at the core of most algorithms that deal with genetic sequences [G97]. In these fields, the huge amount of data to be processed – sometimes billions of characters – calls for algorithms that are better than linear.

One way to accelerate the computations is to exploit the parallelism of vector operations, especially bit-vector operations. For example, in [BYG92] and [HM99], bit-vectors are used to code the set of states of a non-deterministic automaton. In this paper, as in [M99], we want to accelerate computations done with deterministic automata, and we use vectors to represent sequences of events or sequences of states.

Given a deterministic finite automaton, and an input sequence  $x_1 \dots x_m$ , we are interested in the output sequence  $y_1 \dots y_m$  of *visited states*. Since executing one transition is usually considered to be a constant time operation, the output sequence can be obtained in  $\mathcal{O}(m)$  time.

---

\*LACIM, Université du Québec à Montréal, C.P. 8888 Succursale Centre-Ville, Montréal, Québec, Canada, H3C 3P8. E-mail:anne@lacim.uqam.ca

In order to improve the efficiency of such algorithms, we have to find quicker ways to obtain  $y_1 \dots y_m$  from  $x_1 \dots x_m$ . The best possible algorithm would do it in constant time:

$$\begin{array}{c} x_1 x_2 \dots x_m \\ \downarrow \\ y_1 y_2 \dots y_m \end{array}$$

Clearly, this can seldom be done when  $m$  is unbounded. The next best alternative would be to obtain  $y_1 \dots y_m$  with a bounded number of *vector* operations on  $x_1 \dots x_m$ :

$$\begin{array}{cccc} x_1 & x_2 & \dots & x_m \\ \downarrow & \downarrow & & \downarrow \\ y_1 & y_2 & \dots & y_m \end{array}$$

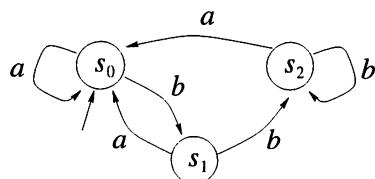
Indeed, vector operations can be implemented in parallel, in dedicated circuits, or using high-speed bit-wise operations available in processors. The drawback is that vector operations are applied component by component, meaning that the only computations that one could hope to solve with *pure* vector operations are those where the value of  $y_i$  depends only on the value of  $x_i$ , and its close neighbors.

On the bright side, some bit operations widely available in processors do have a *memory* of past events. Using these, it is possible to parallelize complex computations done with automata.

## 2 The Basics of Vector Algorithms

### 2.1 A First Example

As a simple example, consider the following automaton. On input sequence *babba*, it will generate the output  $s_1 s_0 s_1 s_2 s_0$  – we omit the leading initial state.



Let's associate to a string  $x_1 \dots x_m$ , the *characteristic vector* of a letter  $l$ , denoted by the bold letter  $\mathbf{l}$ , and defined by:

$$\mathbf{l}_i = \begin{cases} 1 & \text{if } x_i = l \\ 0 & \text{otherwise} \end{cases}$$

Characteristic vectors are sequences of bits, and we will operate on them with the standard bit operations: bit-wise logical operators, left and right shifts, binary addition, etc. We can obtain, for example, the characteristic vector of a set  $S$  of letters by computing the disjunction of the characteristic vectors of the letters in  $S$ .

In our example, we have the following *direct* computation of the characteristic vectors  $s_0$ ,  $s_1$  and  $s_2$  of the output sequence  $y_1 \dots y_m$ , in terms of the characteristic vectors  $a$  and  $b$  of the input sequence.

$$s_0 = a \quad (1)$$

$$s_1 = (\uparrow_1 s_0) \wedge b \quad (2)$$

$$s_2 = \neg(s_0 \vee s_1) \quad (3)$$

where  $\uparrow_b l$  stands for a right shift of the vector  $l$  with the value  $b$  filled in in the first position.

These three equations are used in the following way. Suppose, for example, that the input sequence given to the automaton is *babba*. The characteristic vectors of the letters  $a$  and  $b$  are thus:

$$\begin{aligned} a &= 01001 \\ b &= 10110 \end{aligned}$$

Equation (1) states that the output is  $s_0$  if and only if the input is  $a$ , thus:

$$s_0 = 01001$$

The output state is  $s_1$  when the input letter is  $b$ , and the preceding state is  $s_0$ . This can be computed, as in equation (2), by shifting vector  $s_0$  to the right and taking the bit-wise conjunction with vector  $b$ .

$$\begin{aligned} s_1 &= 10100 \wedge 10110 \\ &= 10100 \end{aligned}$$

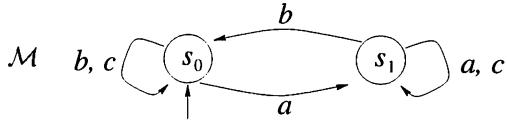
In all other cases, the output state is  $s_2$ , which can be expressed, as in equation (3), by the bit-wise negation of the disjunction of characteristic vectors  $s_0$  and  $s_1$ :

$$\begin{aligned} s_2 &= \neg(01001 \vee 10100) \\ &= 00010 \end{aligned}$$

If we assume that vector operations are done in parallel then, regardless of the length of the input sequence, the characteristic vectors of the output can be computed with 4 operations! This example is simple, since the output state depends on at most two input letters, but it gives the flavor of the technique. In general, the output states will depend on arbitrarily "far" events but, in some interesting cases, it will still be possible to reduce the computation to direct bit-vector operations.

## 2.2 Remembering Past Events

The simplest example of a computation that is influenced by past events is given by the following automaton  $\mathcal{M}$ .

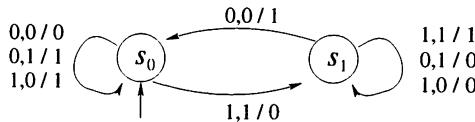


In this case, whether input  $c$  yields state  $s_0$  or  $s_1$  depends on events that can be arbitrarily far, as shown by the two input sequences  $ac^n$  and  $bc^n$ . Fortunately, we have the following formulas that compute  $s_0$  and  $s_1$  using binary addition with carry propagation – performed from left to right on the bit-vectors:

**The Addition Lemma:** The characteristic vectors of states  $s_0$  and  $s_1$  of  $\mathcal{M}$  are

$$\begin{aligned} s_0 &= b \vee [c \wedge (\neg b + \neg(b \vee c))] \\ s_1 &= a \vee [c \wedge \neg(a + (a \vee c))]. \end{aligned}$$

*Proof:* As noted in [M99], automaton  $\mathcal{M}$  is similar to the classical bit addition Moore automaton:



In this automaton, state  $s_1$  means that the *carry bit* is set to 1. This state is reached when both bits are 1, or when the two bits are different, and their sum is 0. Thus, if two vectors  $x_1$  and  $x_2$  are added, the characteristic vector of state  $s_1$  is

$$(x_1 \wedge x_2) \vee [((\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)) \wedge \neg(x_1 + x_2)].$$

Using the characteristic vectors associated with events  $a$ ,  $b$  and  $c$  of the automaton  $\mathcal{M}$ , define  $x_1 = a$  and  $x_2 = a \vee c$ . Then:

$$\begin{aligned} a &= x_1 \wedge x_2 \\ b &= \neg x_1 \wedge \neg x_2 \\ c &= \neg x_1 \wedge x_2 \end{aligned}$$

With these identities, automaton  $\mathcal{M}$  becomes a sub-graph of the bit addition automaton. Since  $x_1 \wedge \neg x_2$  is always false, we get, by substitution, the formula  $s_1 = a \vee [c \wedge \neg(a + (a \vee c))]$ . The formula for  $s_0$  is derived in a similar way with  $x_1 = \neg b$  and  $x_2 = \neg(b \vee c)$ . ■

## 2.3 Solving More Complex Automata

In this section, we establish a sufficient condition for the existence of a vector algorithm for an automaton.

Consider a finite *complete* automaton  $\mathcal{A}$  on alphabet of events  $\Sigma$ , with states  $Q$  and transition function  $Q \times \Sigma \xrightarrow{T} Q$ . We say that a state  $s$  is *decidable* if, for all events  $x \in \Sigma$  that do not loop on  $s$ , either all states reach  $s$  with  $x$ , or none does.

Formally:

$$\begin{aligned} \forall s' \in Q \quad T(s', x) = s & \quad \text{or} \\ \forall s' \neq s \in Q \quad T(s', x) \neq s. \end{aligned}$$

A decidable state can be *removed* from an automaton in the following sense. Let  $E_s$  be the set of events for which  $T(s', x) = s$  for all  $s'$ , and  $\mathcal{A} \setminus \{s\}$  be the automaton obtained from  $\mathcal{A}$  by removing  $s$ , and all its pending arrows. Then if  $s$  is decidable,  $\mathcal{A} \setminus \{s\}$  is still a complete automaton on the alphabet  $\Sigma \setminus E_s$ , since  $T(s', y) \neq s$  if  $y$  is not in  $E_s$ .

**Definition 2.1** An automaton  $\mathcal{A}$  is *decidable* if it has one state, or if it has one decidable state  $s$ , and  $\mathcal{A} \setminus \{s\}$  is decidable.

**Theorem 2.1** If an automaton  $\mathcal{A}$  is decidable, then there exists a vector algorithm for  $\mathcal{A}$ .

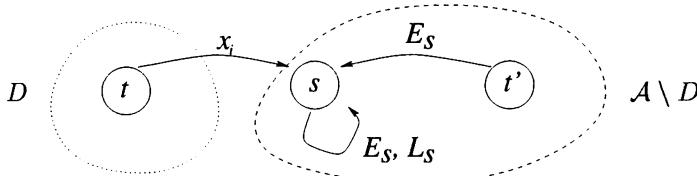
*Sketch of Proof.* When a state  $s$  is decidable, the Addition Lemma can be used to compute the characteristic vector of  $s$ . Indeed, let  $\mathbf{E}_s$  be the characteristic vector of the set  $E_s$ , and  $\mathbf{L}_s$ , the characteristic vector of the events that loop on  $s$  but are not in  $E_s$ . Then:

$$s = \begin{cases} \mathbf{E}_s \vee [\mathbf{L}_s \wedge (\neg \mathbf{E}_s + \neg(\mathbf{E}_s \vee \mathbf{L}_s))] & \text{or} \\ \mathbf{E}_s \vee [\mathbf{L}_s \wedge \neg(\mathbf{E}_s + (\mathbf{E}_s \vee \mathbf{L}_s))] \end{cases} \quad (4)$$

according to whether  $s$  is initial or not.

Suppose that we have computed the characteristic vectors of a subset  $D$  of decidable states, and let  $\mathbf{K}$  be the disjunction of all those known characteristic vectors.

If  $s$  is a decidable state in  $\mathcal{A} \setminus D$ , we will show that computing the characteristic vector of  $s$  is essentially a *local* decision, and can be done in a vectorial way.



The value of  $s_i$  will be equal to 1 in three circumstances. First, if the preceding state is known, that is, when  $K_{i-1} = 1$ , then the value of  $s_i = 1$  can be decided with the transition table of  $\mathcal{A}$ . If the preceding state is unknown, then it belongs to  $\mathcal{A} \setminus D$ , and  $s_i = 1$  if the input  $x_i$  is in  $E_s$ . Let  $\mathbf{N}$  be the characteristic vector resulting from these two possibilities.

Vector  $\mathbf{N}$  covers, at least, all the cases when  $s$  is reached from a different state. In order to account for looping events in  $L_s$ , we apply Equation 4 with  $\mathbf{E}_s = \mathbf{N}$ . ■

Theorem 2.1 proves the existence of a vector algorithm, but does not give an efficient way to construct one. In the following sections, we will study in details a non trivial application.

### 3 Approximate String Matching

A common way to formalize the notion of *distance* between two strings is the *edit distance*, based on the number of operations required to transform one string into another. Three basic operations are permitted on individual characters: insertion, deletion and replacement.

For example, in order to transform the string **COMPUTER** into the string **SOLUTION**, we can apply to the first string the sequence of edit operations *RMDRMMIRR*, where *R* denotes a replacement, *D* a deletion, *I* an insertion and *M* a match.

Such a transformation is usually displayed as an *alignment* of the two strings, where matched or replaced letters are on top of each other, and insertions and deletions are denoted by a properly placed dashes. With our example, we get the alignment:

```
C O M P U T - E R
S O - L U T I O N
```

The *edit distance* between two strings is defined as the minimum number of edit operations – excluding matches – needed to transform one string into another.

A crucial generalization of the edit distance for applications in biology is the *weighted edit distance*. It comes from the observation – in biological sequences – that replacements are not equally likely. Assigning different costs to different edit operations allows the construction of alignments that are meaningful from an evolutionary point of view.

Let  $c$  be the cost associated to an insertion or a deletion, and  $\delta(a, b)$  be the cost of replacing  $a$  by  $b$ . We define the cost of a sequence of edit operations to be the sum of the costs of the operations involved. Since a replacement can be achieved by a deletion followed by an insertion, the replacement cost  $\delta(a, b)$  should be less than  $2c$ .

**Definition 3.1** *The weighted edit distance  $\delta(A, B)$  between two string  $A$  and  $B$  is the minimal cost to transform  $A$  into  $B$ .*

In the sequel, we will focus on the following problem. Given a query sequence  $P = p_1 \dots p_m$ , and a text  $T = t_1 \dots t_n$ , we want to find the *approximate occurrences* of  $P$  in  $T$ . Formally, the problem is to find all positions  $j$  in  $T$  such that, for a given threshold  $t \geq 0$ , we have  $\min_g \delta(P, T[g, j]) \leq t$ . Typically,  $P$  will be relatively short – a few hundred characters –, while  $T$  can be quite large.

The classic solution [S80] is obtained by computing the matrix  $C[0..m, 0..n]$  with the recurrence relation:

$$C[i, j] = \min \begin{cases} C[i - 1, j - 1] + \delta(p_i, t_j) \\ C[i, j - 1] + c \\ C[i - 1, j] + c \end{cases} \quad (5)$$

and initial conditions  $C[0, j] = 0$  and  $C[i, 0] = ic$ .

The successive values of  $C[m, j]$  give the desired distances and can be compared to the threshold  $t$ . For example, suppose one wants to compute the approximate occurrences of *TATA* in the text *ACGTAATAGC...* with the usual edit distance, that is  $c = 1$  and  $\delta(a, b) = 1$  if  $a \neq b$ . The computation is done with the help of a grid whose cells hold the values of  $C[i, j]$ .

The following table gives a snapshot of the evolving computation:

	<i>A</i>	<i>C</i>	<i>G</i>	<i>T</i>	<i>A</i>	<i>A</i>	<i>T</i>	<i>A</i>	<i>G</i>	<i>C</i>	...
<i>T</i>	0	0	0	0	0	0	0	0	0	0	...
<i>T</i>	1	1	1	1	0	1	1	0	1	1	...
<i>A</i>	2	1	2	2	1	0	1	1	0	1	...
<i>T</i>	3	2	2	3	2	1	1	1	1	1	...
Line <i>m</i>	<i>A</i>	4	3	3	3	2	1	2	1	2	...

↑      ↑

In this table, we can see that, in two occasions, the query string is at one unit of distance from substrings in the text – the substring *TAA*, at position 6, and the substrings *ATA*, *AATA*, and *TAATA*, at position 8.

The whole computation can be carried out column by column, requiring  $\mathcal{O}(nm)$  time and  $\mathcal{O}(m)$  space. In order to do better, we first state a useful lemma that bounds the absolute value of differences between horizontal and vertical values in the matrix:

**Lemma 1**  $|C[i, j] - C[i - 1, j]| \leq c$  and  $|C[i, j] - C[i, j - 1]| \leq c$ . ■

Since the horizontal and vertical differences are bounded, we can code this computation as an automaton, which will turn out to be decidable.

### 3.1 Computing Distances With an Automaton

With the notations of Equation 5, define:

$$\begin{aligned}\Delta v_{i,j} &= c - (C[i, j] - C[i - 1, j]) \\ \Delta h_{i,j} &= C[i, j] - C[i, j - 1] + c\end{aligned}$$

From Lemma 1,  $\Delta v_{i,j}$  and  $\Delta h_{i,j}$  are in the interval  $[0..2c]$ , and if the successive values of  $\Delta h_{m,j}$  are known, then the value of the score,  $C[m, j]$ , can be computed by the recurrence,  $C[m, j] = C[m, j - 1] + \Delta h_{m,j} - c$ , and initial condition  $C[m, 0] = mc$ .

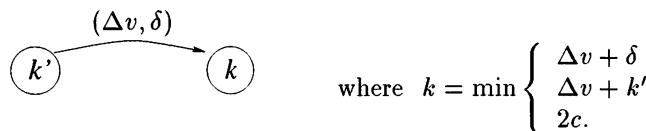
With elementary arithmetic manipulations, Equation 5 translates as:

$$\Delta h_{i,j} = \min \left\{ \frac{\Delta v_{i,j-1} + \delta(p_i, t_j)}{2c}, \frac{\Delta v_{i,j-1} + \Delta h_{i-1,j}}{2c} \right\} \quad (6)$$

with initial conditions  $\Delta h_{0,j} = c$  and  $\Delta v_{i,0} = 0$ . We can thus define an automaton  $\mathcal{B}$  that will compute the sequence  $\Delta h_j = \Delta h_{1,j} \dots \Delta h_{m,j}$  given the sequence of pairs:

$$(\Delta v_{j-1}, \delta(p, t_j)) = ((\Delta v_{1,j-1}, \delta(p_1, t_j)) \dots (\Delta v_{m,j-1}, \delta(p_m, t_j))).$$

The states of  $\mathcal{B}$  are  $\{0, \dots, 2c\}$ , with initial state  $c$ , and the transition function of  $\mathcal{B}$  is given by following diagram, for an event  $(\Delta v, \delta)$  in the cartesian product  $[0..2c] \times [0..2c - 1]$ .



**Theorem 3.1** *Automaton  $\mathcal{B}$  is decidable.*

*Sketch of proof.* We will show that, for  $k \in [0..2c]$ ,  $k$  is decidable in  $\mathcal{B}_{k-1} = \mathcal{B} \setminus \{0, \dots, k-1\}$ , with the set of events  $(\Delta v, \delta)$  such that  $\Delta v + \delta = k$ .

First note that the only remaining events in  $\mathcal{B}_{k-1}$  are those that satisfy  $\Delta v + \delta \geq k$ . If  $\Delta v + \delta = k$ , then  $\min(\Delta v + \delta, \Delta v + k', 2c) = k$ , since  $k' \geq k$  for states in  $\mathcal{B}_{k-1}$ . If both  $\Delta v + \delta > k$  and  $k' > k$ , then the minimum is certainly greater than  $k$ . ■

Theorem 3.1 can then be used to produce a corresponding vector algorithm. Note that, in this case, looping events are easily detected, since if  $k = k' < 2c$ , then either  $\Delta v = 0$  or  $\Delta v + \delta = k$ .

In order to complete the presentation of a vector algorithm for the computation of  $\Delta v_j$ , we use the relation  $\Delta v_{i,j} = \Delta h_{i-1,j} + \Delta v_{i,j-1} - \Delta h_{ij}$  which leads to the vector equation:

$$\Delta v_j = \uparrow_c \Delta h_j + \Delta v_{j-1} - \Delta h_j.$$

## 4 A Bit-Vector Algorithm

This section contains brief notes on how to implement a *bit*-vector algorithm for the approximate string matching problem. The actual implementation will be discussed in a subsequent paper.

The first problem is to represent vectors of integers that are not necessarily 0 or 1, but in the bounded interval  $[0..2c]$ . This can be done easily with an  $l \times m$  bit-matrix, where  $l = \log(2c) + 1$ . There are well known algorithms for all the basic operations – assignment, comparison and arithmetic – on these bit-matrices. These operations are identified with arrows, and bold constants stand for constant vectors.

Assuming that  $\Delta v$  is known, the next letter of the text is read, and  $\delta(p, t_j)$  is looked up in a pre-computed table. Three vectors, corresponding respectively to the three cases of the proof of Theorem 2.1, are initialized in the following way:

$Sum_1 \leftarrow \Delta v$	% $Sum_1$ will hold the values of $\Delta h_{i-1} + \Delta v_i$ .
$Sum_2 \leftarrow \Delta v \vec{+} \delta$	% $Sum_2$ is used to test if $(\Delta v, \delta) \in E_k$ .
$Loop \leftarrow (\Delta v \equiv 0)$	% $Loop$ contains looping events not in $E_k$ .

The characteristic vector  $N$  of state  $k$ , from 0 to  $2c - 1$ , is computed with the following equations, keeping track of the known states  $K$ .

$N \leftarrow (\uparrow_b K \wedge (Sum_1 \equiv k)) \vee (Sum_2 \equiv k)$	% $b = (k \geq c)$ .
$N \leftarrow N \vee [Loop \wedge \neg(N + (N \vee Loop))]$	
$N \leftarrow N \wedge \neg K$	
$K \leftarrow K \vee N$	
$Sum_1 \leftarrow Sum_1 \vec{+} \neg(\uparrow_b K)$	

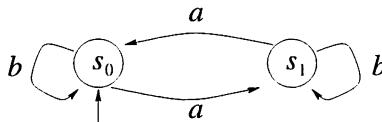
The values of the vector  $\Delta h$  can then be set to  $k$  using the mask  $N$ . According to Theorem 2.1, when  $k = c$ , the initial state, the computation of  $N$  should use an alternative formula. But, in this case, the first bit of  $N$  is properly set by the first instruction since  $Sum_1$  contains the value  $c$  if  $\Delta v_i = 0$ .

Finally the algorithm computes the characteristic vector of state  $2c$ , the score, and the new value of  $\Delta v$ .

## 5 Further Developments

As a first remark, we want to underline the fact that testing decidability for an automaton is a simple procedure that can be easily automated. If an automaton is decidable, it should also be possible to obtain automatically a corresponding vector algorithm, though not necessarily optimal. Indeed, in order to produce an efficient algorithm, Section 4 relied on the fact that the transition table of the automaton had many “arithmetic” symmetries. Is it possible to optimize the general algorithm?

Another interesting avenue is to broaden the class of automata that can be parallelized. Clearly, not all automata are decidable, the simplest counter-example being:



In this case, if one looks at an event  $x$  as a function from  $x : Q \rightarrow Q$ , then both  $a$  and  $b$  are permutations. For an automaton to be decidable, it must have at least one *constant* event. One way to generalize the notion of decidability would be to extend it to constant *composition* of events, hinting at the possibility that properties of the syntactic monoid (see [S94] and [P97]) are related to the existence of vector algorithms.

## References

- [A90] A. Aho, *Algorithms for Finding Patterns in Strings*, in Handbook of Theoretical Computer Science, Vol. A, Elsevier 1990, 255-300.
- [BYG92] R. A. Baeza-Yates and G. H. Gonnet, *A New Approach to Text Searching*, Communications of the ACM, 35, 1992, 74-82.
- [CSY99] C.S. Calude, K. Salomaa, S. Yu, *Metric Lexical Analysis*, WIA99 Proceedings, Potsdam, July 1999, (to appear).
- [G97] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, 1997.
- [HM99] J. Holub and B. Melichar, *Implementation of Nondeterministic Finite Automata for Approximate Pattern Matching*, in Automata Implementation, LNCS 1660, Springer-Verlag, 1999.
- [M99] E. Myers, *A Fast Bit-Vector Algorithm for Approximate String Matching Based on Dynamic Programming*, J. ACM, 46-3, 1999, 395-415.
- [P97] J.-E. Pin, *Syntactic Semigroups*, in Handbook of Formal Languages, Vol. 1, Springer, 1997, 679-738.
- [S80] P. H. Sellers, *The Theory and Computation of Evolutionary Distances*, J. of Algorithms, 1, 1980, 359-373.
- [S94] H. Straubing, Finite Automata, Formal Logic and Circuit Complexity, Birkhauser, Boston, 1994.

# Pattern frequency sequences and internal zeros

Miklós Bóna \*      Bruce E. Sagan †      Vincent R. Vatter ‡

## Abstract

Let  $q$  be a pattern and let  $S_{n,q}(c)$  be the number of  $n$ -permutations having exactly  $c$  copies of  $q$ . We investigate when the sequence  $(S_{n,q}(c))_{c \geq 0}$  has internal zeros. If  $q$  is a monotone pattern it turns out that, except for  $q = 12$  or  $21$ , the sequence always has internal zeros. For the pattern  $132$ , the only possible places for internal zeros are the next-to-last or the second-to-last positions. However, we also show that there are infinitely many values of  $n$  for which the sequence does not have any internal zeros.

## Résumé

On démontre que la suite  $(S_{n,q}(c))_{c \geq 0}$ , dénombrant les permutations de longueur  $n$  contenant  $c$  copies du motif  $q$  contient des zéros internes si  $q$  est monotone et plus longue que  $2$ . Nous prouvons aussi que dans le cas de  $q = 132$ , il existe un nombre infini d'entiers  $n$  tel que la suite  $(S_{n,q}(c))_{c \geq 0}$  ne contient pas de zéros internes.

## 1 Introduction

Let  $q = q_1 q_2 \dots q_k$  be a permutation in the symmetric group  $S_k$ . We call  $k$  the *length* of  $q$ . We say that the permutation  $p = p_1 p_2 \dots p_n \in S_n$  contains a  *$q$ -pattern* if and only if there is a set of indices  $1 \leq i_{q_1} < i_{q_2} < \dots < i_{q_k} \leq n$  such that the elements of the subsequence  $p_{i_1} p_{i_2} \dots p_{i_k}$  are in the same relative order as those in  $q$ . For example,  $41523$  contains exactly two  $132$ -patterns, namely  $152$  and  $153$ . We let

$$c_q(p) = \text{the number of copies of } q \text{ in } p,$$

so that  $c_{132}(41523) = 2$ . Permutations containing a given number of  $q$ -patterns have been extensively studied recently [1–11].

In this paper, we consider permutations with a given number of  $q$ -patterns from a new angle. Let

$$S_{n,q}(c) = \text{the number of } n\text{-permutations with exactly } c \text{ patterns of type } q.$$

---

\*University of Florida, Gainesville FL 32611. Email: bona@math.ufl.edu.

†Michigan State University, E. Lansing, MI 48824. Email: sagan@math.msu.edu

‡Michigan State University, E. Lansing, MI 48824. Email: vattervi@msu.edu

For  $n$  and  $q$  fixed, the sequence  $(S_{n,q}(c))_{c \geq 0}$  is called the *frequency sequence* of the pattern  $q$  for  $n$ . We also say that an  $n$ -permutation  $p$  is  *$q$ -optimal* if there is no  $n$ -permutation with more copies of  $q$  than  $p$ , and let

$$M_{n,q} = c_q(p) \text{ for an optimal } p.$$

The only  $q$  for which the frequency sequence is well understood is  $q = 21$  (or equivalently  $q = 12$ ). Occurrences of this pattern are called inversions. It is well known [12] that for all  $n$ , the frequency sequence of inversions is log-concave, and so is unimodal.

When  $q$  is has length greater than 2, numerical evidence suggests that the frequency sequence of  $q$  will no longer be unimodal, let alone log-concave. In fact, internal zeros seem to be present in most frequency sequences. An integer  $c$  is called an *internal zero* of the sequence  $(S_{n,q}(c))_{c \geq 0}$  if for some  $c$  we have  $S_{n,q}(c) = 0$ , but there exist  $c_1$  and  $c_2$  with  $c_1 < c < c_2$  and  $S_{n,q}(c_1), S_{n,q}(c_2) \neq 0$ .

In the rest of this paper we study the frequency sequences of the monotone pattern  $q = 12 \dots k$  and for the pattern  $q = 132$ . We will show that in the first case, when  $k \geq 3$  (the case  $k = 2$  has already been mentioned) the sequence always has internal zeros for large enough  $n$ . For 132-patterns we will show that there are infinitely many  $n$  where the sequence has internal zeros and infinitely many where it has none. Note that, by symmetry, this completely characterizes what can happen for patterns of length at most three. We point out that a somewhat related family of problems, the analysis of the sequence  $(S_{n,132}(c))_{n \geq 1}$  for fixed  $c$  has been studied in several papers following [8].

## 2 The monotone case

We will now consider the sequence  $(S_{n,q}(c))_{c \geq 0}$  where  $q = 12 \dots l$ . For later reference, we single out the known case when  $l = 2$  discussed in the introduction.

**Proposition 2.1** *The sequence  $(S_{n,12}(c))_{c \geq 0}$  has no internal zeros (and is, in fact, log concave). The unique optimal permutation is  $p = 12 \dots n$  with*

$$M_{n,12} = \binom{n}{2} \quad \diamond$$

It turns out that this is the only monotone pattern (aside from 21) whose sequence has no internal zeros. To prove this result, define an *inversion* (respectively, *noninversion*) in  $p = p_1 p_2 \dots p_n$  to be a pair  $(p_i, p_j)$  such that  $i < j$  and  $p_i > p_j$  (respectively,  $p_i < p_j$ ).

**Theorem 2.1** *Let  $q = 12 \dots l$  where  $l \geq 3$ . Then in  $S_n$ , the unique optimal permutation is  $p = 12 \dots n$  and*

$$M_{n,q} = \binom{n}{l}.$$

The set of permutations having the next greatest number of copies of  $q$  are those obtained from  $p$  by an adjacent transposition and this number of copies is

$$\binom{n-1}{l} + \binom{n-2}{l-1}. \quad (1)$$

**Proof:** Consider any  $r \in S_n$  different from  $p$ . Then  $r$  has an inversion  $(r_i, r_j)$ . So the number of copies of  $q$  in  $r$  is the number not containing  $r_i$  plus the number which do contain  $r_i$ . The permutations in the latter case can not contain  $r_j$ . So (1) gives an upper bound for the number of copies of  $q$  which is strict unless  $r$  has exactly one inversion. The theorem follows.  $\diamond$

**Corollary 2.2** Let  $q = 12\dots l$  where  $l \geq 3$ . Then  $(S_{n,q}(c))_{c \geq 0}$  has internal zeros

**Proof:** From the previous theorem, we see that the number of zeros directly before  $S_{n,q}(M_{n,q}) = 1$  is

$$\binom{n}{l} - \binom{n-1}{l} - \binom{n-2}{l-1} = \binom{n-2}{l-2} \geq n-2 \geq 1$$

since  $n \geq l \geq 3$ .  $\diamond$

For use in the 132 case, we record the following observation.

**Lemma 2.3** For any integer  $c$  with  $0 \leq c \leq \binom{n}{2}$  there is a permutation  $p \in S_n$  having  $c$  copies of the pattern 21 and no copies of 132.

**Proof:** We induct on  $n$ . The result is clearly true if  $n \leq 2$ . Assuming it is true for  $n-1$ , first consider  $c \leq \binom{n-1}{2}$  and let  $p \in S_{n-1}$  satisfy the lemma. Then the concatenation  $pn \in S_n$  works for such  $c$ . On the other hand, if  $\binom{n-1}{2} < c \leq \binom{n}{2}$  then consider  $c' = c - (n-1) \leq \binom{n-1}{2}$ . Pick  $p \in S_{n-1}$  with  $c'$  copies of 21 and none of 132. Then  $np \in S_n$  is the desired permutation.  $\diamond$

### 3 The case $q = 132$ and layered patterns

The rest of this paper is devoted to the study of the frequency sequences of the pattern 132. To simplify notation, we will write  $F_n$  for the sequence  $(S_{n,132}(c))_{c \geq 0}$ . One crucial property of this pattern is that it is layered. This section gives an overview of some important results on layered patterns.

A pattern is *layered* if it is the concatenation of subwords (the *layers*) where the entries decrease within each layer, and increase between the layers. For example,

$3\ 2\ 1\ 5\ 4\ 8\ 7\ 6\ 9$  is a layered pattern with layers 3 2 1, 5 4, 8 7 6, and 9. Layered patterns are examined in Stromquist's work [14] and in Price's thesis [9]. The most important result for our current purposes is the following theorem.

**Theorem 3.1 ([14])** *Let  $q$  be a layered pattern. Then the set of  $q$ -optimal  $n$ -permutations contains at least one layered permutation.*

Layered 132-optimal permutations have a simple recursive structure. This comes from the fact, which we will use many times, that to form a 132 pattern in a layered permutation one must take a single element from some layer and a pair of elements from a subsequent layer

**Proposition 3.2** *Let  $p$  be a layered 132-optimal  $n$ -permutation whose last layer is of length  $m$ . Then the leftmost  $k = n - m$  elements of  $p$  form a 132-optimal  $k$ -permutation.*

**Proof:** Let  $D_k$  be the number of 132-copies of  $p$  that are disjoint from the last layer. The number of 132-copies of  $p$  is clearly  $k \binom{m}{2} + D_k$ . So once  $k$  is chosen,  $p$  will have the maximum number of copies if  $D_k$  is maximal.  $\diamond$

We point out that the proof of this proposition uses the fact that 132 has only two layers, the first of which is a singleton. Let  $M_n = M_{n,132}$ . Then the previous proposition implies that

$$M_n = \max_{1 \leq k < n} \left( M_k + k \binom{m}{2} \right). \quad (2)$$

The integer  $k$  for which the right hand side attains its maximum will play a crucial role throughout this paper. Therefore, we introduce specific notation for it.

**Definition 3.3** *For any positive integer  $n$ , let  $k_n$  be the positive integer for which  $M_n = \max_k (M_k + k \binom{m}{2})$  is maximal. If there are several integers with this property, then let  $k_n$  be the largest among them.*

In other words,  $k_n$  is the largest possible length of the remaining permutation after removing the last layer of an optimal  $n$ -permutation  $p$ . When there is no danger of confusion, we will only write  $k$  instead of  $k_n$ , to simplify notation. We will also always use  $m = n - k$  to denote the length of the last layer of  $p$ .

## 4 Construction of Permutations with a given number of copies

We will first show that there are infinitely many integers  $n$  such that  $F_n$  does not have internal zeros. We will call such an integer, or its corresponding sequence, *NIZ* (*no*

*internal zero*), and otherwise *IZ*. Our strategy is recursive: We will show that if  $k_n$  is NIZ, then so is  $n$ . As  $k_n < n$ , this will lead to an infinite sequence of NIZ integers. There is a problem, however. In order for this strategy to work, we must ensure that given  $k$ , then there is an  $n$  such that  $k = k_n$ . This is the purpose of the following theorem.

**Theorem 4.1** *The sequence  $(k_n)_{n \geq 1}$  diverges to infinity and satisfies*

$$k_n \leq k_{n+1} \leq k_n + 1$$

*for all  $n \geq 1$ . So, in particular, for all positive integers  $k$  there is a positive integer  $n$  so that  $k_n = k$ .*

The next section is devoted to a proof of this theorem. We suggest that the reader assume the result now and continue with this section to preserve continuity. Before starting the proof of our main theorem, we need only note the useful fact that

$$M_k \geq \binom{k-1}{2} \tag{3}$$

which follows by considering the permutation  $1k(k-1)(k-2)\cdots 32$ .

**Theorem 4.2** *There are infinitely many NIZ integers.*

**Proof:** It is easy to verify that  $n = 4$  is NIZ. So, by Theorem 4.1, it suffices to show that if  $k_n \geq 4$  is NIZ then so is  $n$ . To simplify notation in what follows, we will write  $k$  for  $k_n$ .

Now given  $c$  with  $0 \leq c \leq M_n = M_k + k \binom{m}{2}$  we will construct a permutation  $p \in S_n$  having  $c$  copies of 132. Because of (3) and  $k \geq 4$  we have  $M_k \geq k-1$ . So it is possible to write  $c$  (not necessarily uniquely) as  $c = ks + t$  with  $0 \leq s \leq \binom{m}{2}$  and  $0 \leq t \leq M_k$ . Since  $k$  is NIZ, there is a permutation  $p' \in S_k$  with  $c_{132}(p') = t$ . Also, by Lemma 2.3, there is a permutation in  $S_m$  with no copies of 132 and  $s$  copies of 21. Let  $p''$  be the result of adding  $k$  to every element of that permutation. Then, by construction,  $p = p'p'' \in S_n$  and  $c_{132}(p) = ks + t = c$  as desired. ◇

One can modify the proof of the previous theorem to locate precisely where the internal zeros could be for an IZ sequence. We will need the fact (established by computer) that for  $n \leq 12$  the only IZ integers were 6, 8, and 9, and that they all satisfied the following result.

**Theorem 4.3** *For any positive integer  $n$ , the sequence  $F_n$  does not have internal zeros, except possibly for  $c = M_n - 1$  or  $c = M_n - 2$ , but not both.*

**Proof:** We prove this theorem by induction on  $n$ . As previously remarked, it is true if  $n \leq 12$ . Now suppose we know the statement for all integers smaller than  $n$ , and prove it for  $n$ . If  $n$  is NIZ, then we are done.

If  $n$  is IZ then, by the proof of Theorem 4.2,  $k = k_n$  is IZ. So  $k \geq 6$  and we have  $M_k \geq k+2$  by (3). Now take  $c$  with  $0 \leq c \leq M_n - 3$  so that we can write  $c = ks + t$  with  $0 \leq s \leq \binom{m}{2}$  and  $0 \leq t \leq M_k - 3$ . Since the portion of  $F_k$  up to  $S_{k,132}(M_k - 3)$  has no internal zeros by induction, we can use the same technique as in the previous Theorem to construct a permutation  $p$  with  $c_{132}(p) = c$  for  $c$  in the given range. Furthermore, this construction shows that if  $S_{k,132}(M_k - i) \neq 0$  for  $i = 1$  or  $2$  then  $S_{n,132}(M_n - i) \neq 0$ . This completes the proof.  $\diamond$

## 5 The sequence $(k_n)_{n \geq 0}$

In order to prove Theorem 4.1, we first need a lemma about the lengths of various parts of a 132-optimal permutation  $p$ . In all that follows, we use the notation

- $b =$  the length of the penultimate layer of  $p$
- $a =$  the length of the permutation gotten by removing the last two layers of  $p$
- $= n - m - b$
- $= k - b.$

Also observe that the sequence  $(M_n)_{n \geq 2}$  is strictly increasing. This is because there exists a layered optimal permutation  $p \in S_n$  with  $n \geq 2$  having at least two layers. (In fact, for  $n \geq 3$  it must have at least two layers.) So the permutation obtained by inserting  $n+1$  between any two layers will have more 132-patterns than  $p$ . It follows from (2) that  $m \geq 2$  for  $n \geq 3$ , a fact that will be useful in proving the following result.

**Lemma 5.1** *For  $n \geq 3$ , we have the following inequalities*

- (i)  $b \leq m - 1$ ,
- (ii)  $a \leq (m - 1)/2$ ,
- (iii)  $m > k$  which implies  $m > n/2$  and  $k < n/2$ ,
- (iv)  $m \leq 2(n + 1)/3$ .

**Proof:** The basic idea behind all four of the inequalities is as follows. Let  $p'$  be the permutation obtained from our 132-optimal permutation  $p$  by replacing its last two layers with a last layer of length  $m'$  and a next-to-last layer of length  $b'$ . Then in passing from  $p$  to  $p'$  we lose some 132-patterns and gain some. Since  $p$  was optimal, the number lost must be at least as large as the number gained. And this inequality can be manipulated to give the one desired.

For the details, the following chart gives the relevant information to describe  $p'$  for each of the four inequalities. In the second case, the last two layers of  $p$  are combined

into one, so the value of  $b'$  is irrelevant.

$m'$	$b'$	number of gained 132-patterns $\leq$ number of lost 132-patterns
$b + 1$	$m - 1$	$(m - 1)\binom{b}{2} \leq b\binom{m-1}{2}$
$b + m$	—	$abm \leq b\binom{m}{2}$
$m + 1$	$b - 1$	$(a + b - 1)m \leq \binom{m}{2} + a(b - 1)$
$m - 1$	$b + 1$	$\binom{m-1}{2} + ab \leq (a + b)(m - 1)$

Now (i) and (ii) follow immediately by cancelling  $b(m - 1)$  and  $bm$ , respectively, from the inequalities in the first two rows of the table. Note that neither of the canceled quantities are zero because of the remarks preceding this lemma. From these two, it follows that  $a(b - 1) < \binom{m}{2}$ . So using the third line of the chart

$$(k - 1)m = (a + b - 1)m \leq \binom{m}{2} + a(b - 1) < 2\binom{m}{2} = m(m - 1)$$

and cancelling  $m$  gives (iii). For (iv) we have

$$\binom{m-1}{2} \leq \binom{m-1}{2} + ab \leq (a + b)(m - 1) = (n - m)(m - 1).$$

Canceling  $m - 1$  and solving for  $m$  completes the proof.  $\diamond$

We now turn to the proof of Theorem 4.1. First note that, by Lemma 5.1 (iv), we have

$$k = n - m \geq \frac{n - 2}{3}.$$

So  $(k_n)_{n \geq 1}$  clearly diverges to infinity. For our next step, we prove that  $(k_n)_{n \geq 1}$  is monotonically weakly increasing. Let  $p_{n,i}$  denote an  $n$ -permutation whose last layer is of length  $n - i$ , and whose leftmost  $i$  elements form an optimal  $i$ -permutation, and let  $c_{n,i} = c_{132}(p_{n,i})$ . Clearly

$$c_{n,i} = M_i + i\binom{n-i}{2}.$$

**Proposition 5.2** *For all  $n \geq 1$ , we have  $k_n \leq k_{n+1}$ .*

**Proof:** It is easy to verify that this is true for  $n \leq 3$ . So let  $n \geq 3$  and  $k = k_n$ . Then it suffices to show that  $c_{n+1,k} > c_{n+1,i}$  for all  $i < k$ . This is equivalent to showing that

$$M_k + k\binom{n+1-k}{2} > M_i + i\binom{n+1-i}{2}. \quad (4)$$

However, by definition of  $k$ , we know that for all  $i < k$ ,

$$M_k + k \binom{n-k}{2} \geq M_i + i \binom{n-i}{2}. \quad (5)$$

Subtracting (5) from (4), we are reduced to proving  $k(n-k) > i(n-i)$ . Rearranging terms so we can cancel  $k-i$  gives the equivalent inequality  $n > k+i$ . But  $k < n/2$  by Lemma 5.1 (iii), and so the bound on  $i$  gives  $k+i < 2k < n$ .  $\diamond$

The proof of the upper bound on  $k_{n+1}$  is a bit more involved but follows the same general lines as the previous demonstration. Note that this will finish the proof of Theorem 4.1.

**Lemma 5.3** *For all positive integers  $n$ , we have  $k_n \leq k_{n+1} \leq k_n + 1$ .*

**Proof:** Induct on  $n$ . The statement is easy to check for  $n \leq 13$ . Suppose we the lemma is true for integers smaller than or equal to  $n$ , and prove it for  $n+1$ . For simplicity, let  $k = k_n$ ,  $m = n - k$ , and  $c_i = c_{n+1,i}$ . Since we have already proved the lower bound, it suffices to show that

$$c_i \geq c_{i+1} \text{ for } k+1 \leq i < \left\lfloor \frac{n+1}{2} \right\rfloor \text{ with strict inequality for } i = k+1. \quad (6)$$

Note that we do not have to consider  $i \geq \lfloor (n+1)/2 \rfloor$  because of Lemma 5.1 (iii).

We prove (6) by induction on  $i$ . For the base case,  $i = k+1$ , we wish to show

$$M_{k+1} + (k+1) \binom{m}{2} > M_{k+2} + (k+2) \binom{m-1}{2}. \quad (7)$$

But since  $p_{n,k}$  is optimal by assumption, we have

$$M_k + k \binom{m}{2} \geq M_{k+1} + (k+1) \binom{m-1}{2}. \quad (8)$$

Subtracting (8) from (7) and rearranging terms, it suffices to prove

$$m-1 > (M_{k+2} - M_{k+1}) - (M_{k+1} - M_k). \quad (9)$$

Let  $p' \in S_k$ ,  $p'' \in S_{k+1}$ , and  $p''' \in S_{k+2}$  be layered 132-optimal permutations having last layer lengths  $m'$ ,  $m''$ , and  $m'''$ , respectively, as short as possible. Since  $n \geq 14$  we have  $k < n/2$  as well as  $k+2 \leq n$  and so, by induction, these three permutations satisfy the lemma. If  $m'' = m' + 1$  then let  $x$  be the largest element in the last layer of  $p''$  (namely  $x = k+1$ ). Otherwise,  $m'' = m'$  and removing the last layer of both  $p'$  and  $p''$  leaves permutations in  $S_{k-m'}$  and  $S_{k-m'+1}$ , respectively. So we can iterate this process until we find the single layer where  $p'$  and  $p''$  have different lengths (those lengths must differ by 1) and let  $x$  be the largest element in that layer of  $p''$ . Similarly we can find the element  $y$  which is largest in the unique layer were  $p''$  and  $p'''$  have different lengths.

Now let

- $r$  = the number of 132-patterns in  $p'''$  containing neither  $x$  nor  $y$ ,
- $s$  = the number of 132-patterns in  $p'''$  containing  $x$  but not  $y$ ,
- $t$  = the number of 132-patterns in  $p'''$  containing  $y$  but not  $x$ , and
- $u$  = the number of 132-patterns in  $p'''$  containing both  $x$  and  $y$ .

Note that there is a bijection between the 132-patterns of  $p'''$  not containing  $y$  and the 132-patterns of  $p''$ . A similar statement holds for  $p''$  and  $p'$ . So

$$M_k = r, \quad M_{k+1} = r + s, \quad M_{k+2} = r + s + t + u.$$

Note also that  $s \geq t$  because increasing the length of the layer of  $x$  results in the most number of 132-patterns being added to  $p'$ . It follows that  $(M_{k+2} - M_{k+1}) - (M_{k+1} - M_k) = t + u - s \leq u$ .

Now, by Lemma 5.1 (iii), to get inequality (9) it suffices to show that  $u < k$ . But there are only  $k$  elements of  $p'''$  other than  $x, y$ , so  $u \leq k$ . If  $u = k$  then  $x, y$  form a 132-pattern with every other element of  $p'''$ . This forces  $p'''$  to have exactly two layers of lengths  $m''' = 2, k''' = k$  or  $m''' = k - 1, k''' = 1$  depending on whether  $x, y$  are both in the same layer or different layers, respectively. But  $n \geq 14$  so  $k \geq 5$  which implies  $m''' \geq 3$  and  $k''' \geq 2$  by the induction on  $n$ . This contradiction completes the proof of (9) and of the base case for the induction on  $i$ .

The proof of the induction step is similar. Assume that (6) is true for  $i - 1$  so that

$$M_{i-1} + (i-1)\binom{l+1}{2} \geq M_i + i\binom{l}{2}. \quad (10)$$

where  $l = n + 1 - i$ . We wish to prove

$$M_i + i\binom{l}{2} \geq M_{i+1} + (i+1)\binom{l-1}{2}. \quad (11)$$

Subtracting as usual and simplifying, we need to show

$$2l - i - 1 \geq (M_{i+1} - M_i) - (M_i - M_{i-1}).$$

Proceeding exactly as in the base case, we will be done if we can show that  $2l - i - 1 \geq i - 1$  or equivalently  $l \geq i$ . But this follows because  $l = n + 1 - i$  and  $i < \lfloor (n + 1)/2 \rfloor$ .  $\diamond$

## References

- [1] M. Bóna, The number of permutations with exactly  $r$  132 subsequences is P-recursive in the size!, *Adv. Appl. Math.* **18** (1997), 510–522.
- [2] M. Bóna, Permutations with one of two 132 subsequences, *Discrete Math.* **181** (1998), 267–274.
- [3] T. Chow and J. West, Forbidden subsequences and Chebyshev polynomials, *Discrete Math.* **32** (1980), 125–161.

- [4] M. Jani and R. G. Rieper, Continued fractions and the Catalan problem, preprint.
- [5] C. Krattenthaler, Permutations with restricted patterns and Dyck paths, preprint.
- [6] T. Mansour, Permutations containing and avoiding certain patterns, preprint.
- [7] J. Noonan, The number of permutations containing exactly one increasing subsequence of length 3, *Discrete Math.* **152** (1996), 307–313.
- [8] J. Noonan, D. Zeilberger, The enumeration of permutations with a prescribed number of “forbidden” subsequences, *Adv. Appl. Math.* **17** (1996), 381–407.
- [9] A. Price, “Packing densities of layered patterns,” Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, 1997.
- [10] A. Robertson, H. S. Wilf, and D. Zeilberger, Permutation patterns and continued fractions, *Electronic J. Combin.* **6** (1999), 6 pages.
- [11] R. Simion and F. W. Schmidt, Restricted permutations, *European J. Combin.* **6** (1985), 383–406.
- [12] R. P. Stanley, Log-concave and unimodal sequences in algebra, combinatorics, and geometry, in “Graph Theory and Its Applications: East and West,” *Ann. NY Acad. Sci.* **576** (1989), 500–535.
- [13] R. P. Stanley, “Enumerative Combinatorics, Volume 1,” Cambridge University Press, Cambridge, 1997.
- [14] W. Stromquist, Packing layered posets into posets, manuscript.

# Enumération et génération aléatoire de cactus $m$ -aires

Michel Bousquet<sup>1</sup>, Cedric Chauve<sup>2</sup>, Gilles Schaeffer<sup>3</sup>

<sup>1</sup> LaCIM, Université du Québec à Montréal,  
201 Avenue Président-Kennedy, Montréal (Québec), Canada

<sup>2</sup> LaBRI, Université Bordeaux I  
351 Cours de la Libération, 33405 Talence, France

<sup>3</sup> LORIA, CNRS, Campus Scientifique  
615 rue du Jardin Botanique - BP 101, 54602 Villers-lès-Nancy, France

**Résumé** Dans cette note nous appliquons une preuve bijective de la formule d'inversion de Lagrange multidimensionnelle pour donner une explication combinatoire de deux formules d'énumération de cactus  $m$ -aires, selon la distribution des couleurs et selon la distribution des degrés. Nous en déduisons un algorithme de génération aléatoire pour ces structures.

**Abstract.** In this note we apply a bijective proof of the multidimensional Lagrange inversion formula to give a combinatorial explanation of two formulae for the enumeration of  $m$ -ary cacti, with respect to the distribution of colors and the distribution of degrees. We deduce from this explanations a uniform random generation algorithm for these structures.

## 1 Introduction

Un  $m$ -cactus est un graphe simple connexe dans lequel chaque arête appartient à exactement un cycle élémentaire. De façon équivalente, chaque composante 2-connexe d'un cactus est un cycle élémentaire, c'est-à-dire un  $m$ -gone (un polygone à  $m$  côtés). Les premières études combinatoires de ces graphes sont dues à Harary et Uhlenbeck [14] et Harary et Palmer [13], suite à un article de Husimi [15]. Un  $m$ -cactus planaire est un plongement d'un  $m$ -cactus dans le plan, de sorte que chaque arête soit incidente à la région infinie. Un *cactus  $m$ -aire* est un  $m$ -cactus planaire tel que les sommets autour d'un  $m$ -gone sont coloriés successivement par les couleurs  $1, 2, \dots, m$  dans le sens trigonométrique. Dans cet article, nous nous intéressons aux *cactus  $m$ -aires enracinés*, c'est-à-dire aux cactus  $m$ -aires dont un polygone (ou de manière équivalente une arête joignant un sommet de couleur 1 et un sommet de couleur 2) est distingué. La Figure 1 présente un cactus ternaire dont le triangle racine (resp. l'arête racine) est rempli (resp. hachurée).

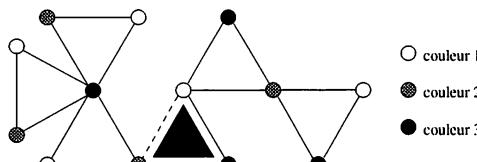


Figure 1: Un cactus ternaire enraciné.

Les cactus sont des objets combinatoires apparaissant dans divers champs des mathématiques, comme l'énumération des factorisations de permutations circulaires dans le groupe symétrique [11] ou la classification topologique des polynômes complexes [8].

On appelle *distribution des couleurs* d'un cactus  $C$  le vecteur  $\mathbf{n} = (n_1, \dots, n_m)$  tel que pour  $i \in [m]$   $n_i$  est le nombre de sommets de  $C$  de couleur  $i$ . On note  $n = \sum_{i=1}^m n_i$ . Par exemple, la distribution des couleurs du cactus de la Figure 1 est  $(5,4,4)$ .

On vérifie aisément (cf [4, 5]) que pour une distribution des couleurs  $\mathbf{n}$  donnée, il y a au moins un cactus  $m$ -aire ayant  $n_i$  sommets de couleur  $i$ , pour  $i \in [m]$ , si et seulement s'il existe un entier  $p$  tel que  $(n - 1) = p(m - 1)$  ( $p$  est alors le nombre de polygones) et, pour  $i \in [m]$ ,  $1 \leq n_i \leq p$ . On dit alors qu'une telle distribution  $\mathbf{n}$  est *valide*.

**Théorème 1** [4, 5] *Soit  $\mathbf{n} = (n_1, \dots, n_m)$  une distribution des couleurs valide. Le nombre de cactus  $m$ -aires de distribution des couleurs  $\mathbf{n}$  est*

$$\frac{1}{p} \prod_{i=1}^m \binom{p}{n_i}, \quad (1)$$

où  $n = \sum_{i=1}^m n_i$  et  $p = (n - 1)/(m - 1)$ .

On appelle *degré* d'un sommet dans un cactus le nombre de polygones auxquels ce sommet est incident. La *distribution des degrés* d'un cactus est une matrice  $D = (d_{i,j})$ , de taille  $m \times \infty$ , où  $d_{i,j}$  est le nombre de sommets de couleur  $i$  et de degré  $j$ . On considère chaque ligne  $(d_{i,1}, d_{i,2}, \dots)$  comme un vecteur  $\mathbf{d}_i$ . On note  $n_i = \sum_j d_{i,j}$  le nombre de sommets de couleur  $i$ ,  $n = \sum_{i=1}^m n_i$  le nombre de sommets et on dit qu'une matrice  $D$  est *valide* si et seulement s'il existe un entier  $p$  tel que  $(n - 1) = p(m - 1)$  et, pour  $i \in [m]$ ,  $1 \leq n_i \leq p$ . Par exemple, la distribution des degrés du cactus de la Figure 1 est

$$D = \begin{bmatrix} 4 & 1 & 0 & 0 & \dots \\ 2 & 2 & 0 & 0 & \dots \\ 3 & 0 & 1 & 0 & \dots \end{bmatrix}.$$

**Théorème 2** [11, 4] *Soit  $D = (d_{i,j})_{m \times \infty}$  une distribution des degrés valide. Le nombre de cactus  $m$ -aires de distribution des degrés  $D$  est*

$$p^{m-1} \prod_{i=1}^m \frac{1}{n_i} \binom{n_i}{\mathbf{d}_i}, \quad (2)$$

où  $n_i = \sum_{j \geq 1} d_{i,j}$ ,  $n = \sum_{i=1}^m n_i$  et  $p = (n - 1)/(m - 1)$ .

Les preuves des Théorèmes 1 et 2 données par Goulden et Jackson [11] ou Bóna, Bousquet, Labelle et Leroux [4] utilisent la forme explicite de la formule d'inversion de séries formelles multivariées de Good-Lagrange (voir les ouvrages [10, 3] pour une description détaillée de cette formule et de ses différentes variantes).

**Théorème 3** [9] *Soient  $\mathbf{R}(\mathbf{x}) = (R_1(\mathbf{x}), \dots, R_m(\mathbf{x}))$  un vecteur de séries formelles en les variables  $\mathbf{x} = (x_1, \dots, x_m)$ , telles que  $R_i(\mathbf{0}) \neq 0$ , pour  $i \in [m]$ , et  $\mathbf{A}(\mathbf{x})$  un vecteur de séries formelles satisfaisant*

$$A_i(x_1, \dots, x_m) = x_i R_i(A_1, \dots, A_m), \quad i = 1, \dots, m. \quad (3)$$

*Alors, pour toute série formelle  $F(\mathbf{x})$  et tout vecteur  $\mathbf{n}$  de  $m$  entiers on a*

$$[\mathbf{x}^\mathbf{n}]F(\mathbf{A}(\mathbf{x})) = [\mathbf{x}^\mathbf{n}] \left\{ \det \left( \delta_{i,j} - \frac{x_i}{R_j(\mathbf{x})} \frac{\partial R_j(\mathbf{x})}{\partial x_i} \right)_{m \times m} F(\mathbf{x}) \prod_{i=1}^m R_i^{n_i}(\mathbf{x}) \right\} \quad (4)$$

où  $[\mathbf{x}^\mathbf{n}]G(\mathbf{x})$  est le coefficient de  $x_1^{n_1} \cdots x_m^{n_m}$  dans  $G(\mathbf{x})$  et  $\delta_{i,j} = 1$  si  $i = j$  et 0 sinon.

Il n'existe pas d'interprétation bijective directe connue de cette formule permettant d'expliquer combinatoirement les résultats découlant de son application. Cela est notamment dû à la présence de termes négatifs dans le déterminant du membre droit de (4). Récemment, Bender et Richmond, motivés par la recherche de formules asymptotiques pour les structures lagrangiennes [1, 2], et Goulden et Kulkarni [12] ont proposé une reformulation du Théorème 3, que l'on appelle *forme arborescente*, basée sur la notion de dérivée d'un vecteur de séries formelles selon une arborescence orientée.

**Définition 1** Soient  $\mathcal{G}$  un graphe orienté étiqueté ayant ensemble de sommets  $S = \{1, \dots, d\}$  et ensemble d'arcs  $A$ ,  $\mathbf{x} = (x_1, \dots, x_d)$  un vecteur de variables formelles et  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x}))$  un vecteur de séries formelles en  $\mathbf{x}$ . On définit la dérivée de  $\mathbf{f}(\mathbf{x})$  par rapport à  $\mathcal{G}$  par

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathcal{G}} = \prod_{j \in S} \left\{ \left( \prod_{(i,j) \in A} \frac{\partial}{\partial x_i} \right) f_j(\mathbf{x}) \right\}.$$

**Théorème 4** [12, 1] Soient  $\mathbf{R}(\mathbf{x}) = (R_1(\mathbf{x}), \dots, R_m(\mathbf{x}))$  un vecteur de séries formelles en les variables  $\mathbf{x}$ , telles que  $R_i(\mathbf{0}) \neq 0$ , pour  $i \in [m]$ , et  $\mathbf{A}(\mathbf{x})$  un vecteur de séries formelles satisfaisant le système d'équations fonctionnelles (3). Alors, pour toute série formelle  $F(\mathbf{x})$  et tout vecteur  $\mathbf{n}$  de  $m$  entiers on a

$$[\mathbf{x}^{\mathbf{n}}]F(\mathbf{A}(\mathbf{x})) = \left( \prod_{i=1}^m \frac{1}{n_i} \right) [\mathbf{x}^{\mathbf{n}-1}] \left\{ \sum_{\mathcal{T}} \frac{\partial(F(\mathbf{x}), R_1^{n_1}(\mathbf{x}), \dots, R_m^{n_m}(\mathbf{x}))}{\partial \mathcal{T}} \right\}, \quad (5)$$

où  $\mathbf{n} - \mathbf{1} = (n_1 - 1, \dots, n_m - 1)$ , la somme étant sur toutes les arborescences  $\mathcal{T}$  ayant  $\{0, 1, \dots, m\}$  pour ensemble de sommets, de racine 0, et dont les arcs sont orientés vers 0.

Goulden et Kulkarni ont montré que la formule (5), qui ne fait apparaître dans son membre droit qu'une somme de termes positifs, est équivalente à (4), et ont donné une interprétation combinatoire des deux membres de (5), en termes d'endofonctions coloriées et d'arborescences coloriées. Deux preuves bijectives de cette forme arborescente de la formule de Good-Lagrange ont été proposées [12, 6], toutes deux basées sur une bijection entre arborescences  $m$ -coloriées et endofonctions  $m$ -coloriées.

Le but de cette note est d'illustrer, en prenant l'exemple des cactus  $m$ -aires, deux aspects intéressants de ces preuves. Premièrement, elles permettent d'expliquer combinatoirement des formules obtenues par application de la formule de Lagrange multidimensionnelle. Deuxièmement, de telles explications combinatoires peuvent induire des algorithmes de génération aléatoire de structures lagrangiennes (structures dont la série génératrice vérifie une équation de la forme (3)). Dans la section suivante, nous présentons rapidement les structures combinatoires utilisées dans la preuve bijective du Théorème 4, puis nous décrivons une bijection entre les cactus  $m$ -aires et une famille d'arborescences, que nous appelons *arborescences cactacées*, qui met en évidence le caractère lagrangien des cactus  $m$ -aires enracinés. Finalement, nous déduisons de ces deux constructions une bijection entre les arborescences cactacées et une famille d'endofonctions que nous appelons *endofonctions cactacées*. La Section 3 contient les preuves combinatoires des Théorèmes 1 et 2, basées sur cette bijection. Finalement la Section 4 propose un algorithme de génération aléatoire et uniforme pour les cactus  $m$ -aires de distribution de couleurs ou de degrés fixée, à  $n$  sommets, en temps et espace  $O(n)$ , basé sur les preuves de la Section 3.

*Remarque 1.* Les preuves bijectives du Théorème 4 de [12, 6] utilisant des structures coloriées étiquetées, nous considérons des cactus  $m$ -aires enracinés étiquetés : pour chaque couleur  $i \in [m]$ , on étiquette les  $n_i$  sommets de couleur  $i$  par les entiers de 1 à  $n_i$  (on note alors  $j_i$  le  $j^{\text{ème}}$  élément de couleur  $i$ ). Les cactus enracinés étant des structures asymétriques (du fait de l'enracinement), il y a  $\mathbf{n}! = \prod_{i=1}^m n_i!$  fois plus de cactus  $m$ -aires enracinés étiquetés ayant  $n_i$  sommets de couleur  $i$  que de cactus  $m$ -aires enracinés non étiquetés ayant  $n_i$  sommets de couleur  $i$ .

## 2 Des cactus aux endofonctions

Dans cette section, nous nous plaçons dans le cadre de la théorie des espèces de structures développée par l'école de combinatoire montréalaise et particulièrement adaptée à l'énumération de structures arborescentes (voir [3, 16]). Le premier paragraphe est un rappel de [12, 6] (voir aussi [7]).

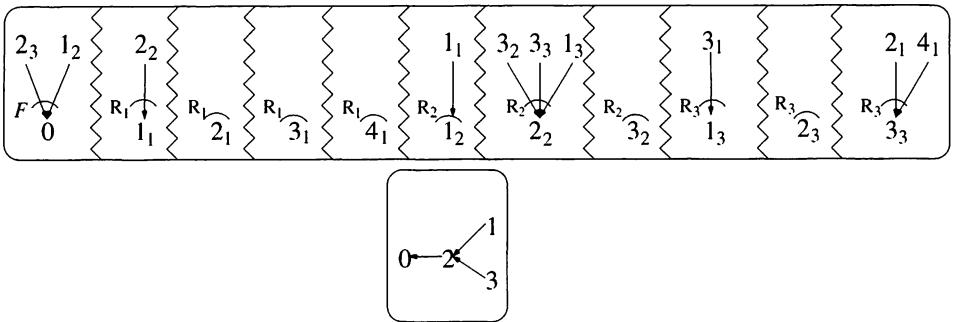
### 2.1 Arborescences, endofonctions et formule de Lagrange multidimensionnelle

Soient  $\mathbf{R} = (R_1, \dots, R_m)$  un vecteur de  $m$  espèces de structures  $m$ -coloriées et  $F$  une espèce  $m$ -coloriée, de séries génératrices exponentielles multivariées respectives  $R_i(\mathbf{x})$ , pour  $i \in [m]$ , et  $F(\mathbf{x})$  ( $\mathbf{x} = (x_1, \dots, x_m)$ ). Les preuves bijectives du Théorème 4 font intervenir deux types de structures combinatoires : les arborescences  $(F, \mathbf{R})$ -enrichies et les endofonctions restreintes  $(F, \mathbf{R})$ -enrichies.

On note  $A_{\mathbf{R},i}$  l'espèce des arborescences  $m$ -coloriées (chaque sommet est colorié par une couleur choisie parmi les couleurs  $1, \dots, m$ )  $\mathbf{R}$ -enrichies (pour  $j \in [m]$ , la fibre - l'ensemble des fils - d'un sommet de couleur  $j$  est munie d'une structure d'espèce  $R_j$ ) dont la racine est de couleur  $i$ . L'espèce des arborescences  $(F, \mathbf{R})$ -enrichies est l'espèce des  $F$ -assemblées de  $(\cup_{i=1}^m A_{\mathbf{R},i})$ -structures. L'ensemble des arborescences  $(F, \mathbf{R})$ -enrichies ayant  $n_i$  éléments de couleur  $i$ , pour  $i \in [m]$ , est noté  $F(\mathbf{A}_{\mathbf{R}})[\mathbf{n}]$ . Dans la suite de cet article, on considère une arborescence  $(F, \mathbf{R})$ -enrichie sur  $[\mathbf{n}]$  comme une arborescence sur  $\{0\} \cup [n_1] \cup \dots \cup [n_m] = [0, \mathbf{n}]$ , de racine 0, telle que la fibre de 0 est munie d'une  $F$ -structure, les fibres des sommets de couleur  $i$  étant toujours munies d'une  $R_i$ -structure. Ces arborescences sont reliées au Théorème 4 par le fait que, si on note  $A_{\mathbf{R},i}(\mathbf{x})$  la série génératrice exponentielle multivariée des  $A_{\mathbf{R},i}$ , on a l'interprétation combinatoire suivante du membre gauche de (5) [12, 6] :

$$|F(\mathbf{A}_{\mathbf{R}})[\mathbf{n}]| = \mathbf{n}![\mathbf{x}^{\mathbf{n}}]F(A_{\mathbf{R},1}(\mathbf{x}), \dots, A_{\mathbf{R},m}(\mathbf{x})). \quad (6)$$

Une endofonction partielle  $(F, \mathbf{R})$ -enrichie sur  $[\mathbf{n}]$  est une fonction  $f$  de  $[\mathbf{n}]$  dans  $[0, \mathbf{n}]$  dont chaque fibre  $f^{-1}(u)$  est munie d'une  $R_i$ -structure si  $u$  est de couleur  $i \in [m]$  et d'une  $F$ -structure si  $u = 0$ . Soit  $f$  une endofonction partielle  $(F, \mathbf{R})$ -enrichie sur  $[\mathbf{n}]$  : on appelle *graphe des couleurs* de  $f$ , noté  $G(f)$ , le graphe orienté (non ordonné : les voisins d'un sommet sont structurés en un ensemble non ordonné) étiqueté ayant  $[0, m]$  pour ensemble de sommets et ayant, pour  $i \in [m]$ , un arc de  $i$  vers  $j$  ( $j \in [0, m]$ ) si et seulement si  $1_i$  est dans la fibre d'un élément de couleur  $j$  ( $1_i \in f^{-1}(k_j)$ , avec  $k \in [n_j]$ ), et un arc de  $i$  vers 0 si  $1_i$  est dans la fibre de 0 ( $1_i \in f^{-1}(0)$ ). On représente habituellement une endofonction partielle  $f$  par le produit des fibres de ses éléments. Par exemple, si  $m = 3$ ,  $n_1 = 4$ ,  $n_2 = 4$  et  $n_3 = 3$ , la figure suivante représente une endofonction partielle  $f$  de  $[\mathbf{n}]$  (un arc de  $i_j$  vers  $k_l$  indique que  $f(i_j) = k_l$ ) et son graphe des couleurs  $G(f)$ .



**Figure 2:** Une endofonction partielle restreinte.

Une endofonction partielle  $(F, \mathbf{R})$ -enrichie  $f$  est dite *restreinte* si et seulement si  $G(f)$  est une arborescence de racine 0 dont tous les arcs sont orientés vers la racine. L'endofonction partielle précédente est donc une endofonction partielle restreinte. Ces structures sont reliées au Théorème 4 par la relation suivante : si on note  $\text{End}_{F, \mathbf{R}}^{\mathcal{R}}[\mathbf{n}]$  l'ensemble des endofonctions restreintes  $(F, \mathbf{R})$ -enrichies sur  $[\mathbf{n}]$ , on a l'interprétation combinatoire suivante du membre droit de (5) [12, 6] :

$$|\text{End}_{F, \mathbf{R}}^{\mathcal{R}}[\mathbf{n}]| = (\mathbf{n} - 1)! [\mathbf{x}^{\mathbf{n}-1}] \sum_{\mathcal{T}} \frac{\partial(F(\mathbf{x}), R_1^{n_1}(\mathbf{x}), \dots, R_m^{n_m}(\mathbf{x}))}{\partial \mathcal{T}}. \quad (7)$$

On a alors le résultat suivant, reliant arborescences et endofonctions (et qui, combiné à (6) et (7), prouve bijectivement le Théorème 4).

**Proposition 1.** [12, 6] *Il existe une bijection  $\Phi$  entre les arborescences  $(F, \mathbf{R})$ -enrichies sur  $[\mathbf{n}]$  et les endofonctions restreintes  $(F, \mathbf{R})$ -enrichies sur  $[\mathbf{n}]$ .*

Cette bijection  $\Phi$  vérifie de plus la propriété suivante (dite *de conservation de la structure des fibres*), qui s'avèrera nécessaire à la preuve du Théorème 2.

**Propriété 1** *Soit  $A$  une arborescence  $(F, \mathbf{R})$ -enrichie sur  $[\mathbf{n}]$  et  $u$  un sommet quelconque de  $A$  : la fibre de  $u$  dans  $A$  contient  $j$  sommets de couleur  $k \in [m]$  si et seulement si la fibre de  $u$  dans l'endofonction restreinte  $(F, \mathbf{R})$ -enrichie sur  $[\mathbf{n}]$   $\Phi(A)$  contient elle aussi  $j$  sommets de couleur  $k$ .*

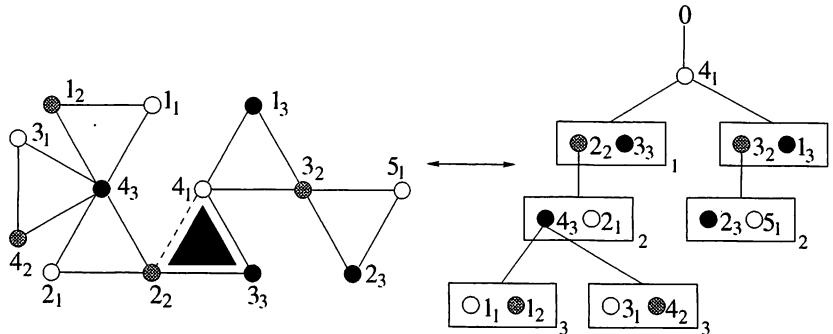
## 2.2 Cactus $m$ -aires et endofonctions

On note  $L$  l'espèce des ordres linéaires,  $X_i$  l'espèce des singletons de couleur  $i$  et, pour un vecteur  $(V_1, \dots, V_m)$  d'espèces,  $\hat{V}_i = \prod_{j \neq i} V_j$  l'espèce du produit de tous les  $V_j$  excepté  $V_i$ .

On appelle *arborescences cactacées* l'espèce des arborescences  $(F, \mathbf{R})$ -enrichies telles que  $F = X_1$  et  $R_i = L(\hat{X}_i)$  : la fibre de 0 est réduite à un seul sommet, de couleur 1, et pour tout sommet  $j_i$ , sa fibre est structurée en une liste ordonnée de  $i$ -*blocs*, un  $i$ -bloc étant un ensemble de  $(m - 1)$  sommets de couleurs différentes de  $i$  et deux à deux différentes.

**Proposition 2.** *Il existe une bijection  $\Psi$  entre les arborescences cactacées sur  $[\mathbf{n}]$  et les cactus  $m$ -aires sur  $[\mathbf{n}]$ .*

Il s'agit d'un résultat classique concernant les cactus  $m$ -aires enracinés, basé sur un parcours en profondeur des polygones d'un cactus, en partant du polygone racine, au cours duquel on remplace chaque polygone rencontré par un  $i$ -bloc si on est entré dans ce polygone par un sommet de couleur  $i$ . On associe donc un bloc à chaque polygone (excepté le polygone racine). La figure suivante, dans laquelle on a représenté les  $i$ -blocs par des rectangles marqués  $i$  et on a "fusionné" les arêtes joignant un bloc à son père en une seule arête, illustre cette construction.



**Figure 3:** Un cactus  $m$ -aire et l'arborescence cactacée correspondante.

Si on appelle degré d'un élément d'une arborescence cactacée le nombre de blocs présents dans sa fibre, on vérifie alors immédiatement la propriété suivante.

**Propriété 2** Soit  $C$  un cactus  $m$ -aire. Si un sommet  $j_i$  de  $C$  différent du (resp. égal au) sommet de couleur 1 du polygone racine est incident à  $k$  polygones dans  $C$  ( $k > 0$ ), alors  $j_i$  a degré ( $k - 1$ ) (resp.  $k$ ) dans l'arborescence cactacée  $\Psi(C)$ .

Par la Proposition 1, on peut alors affirmer que les arborescences cactacées sont en bijection avec la famille des endofonctions restreintes  $(F, \mathbf{R})$ -enrichies telles que  $F = X_1$  et  $R_i = L(\hat{X}_i)$ , que nous appelons *endofonctions cactacées*.

**Propriété 3** Une endofonction cactacée  $f$  sur  $[n]$  satisfait notamment les contraintes suivantes :

1. la fibre  $f^{-1}(0)$  de 0 est constituée du seul élément  $1_1$  (par la contrainte sur le graphe des couleurs d'une endofonction restreinte),
2. le graphe des couleurs est une arborescence de racine 0 ayant pour seul fils le sommet  $1_1$  (conséquence de la contrainte précédente) et dont tous les arcs sont orientés vers 0.

### 3 Enumération de cactus $m$ -aires

Dans cette section, nous proposons une explication combinatoire des Théorèmes 1 et 2 basée sur la bijection entre cactus  $m$ -aires et endofonctions cactacées. Dans un premier temps, nous faisons un rappel sur une construction classique reliant arborescences et mots, le codage de Prüfer.

#### 3.1 Codage de Prüfer

Il s'agit d'une bijection entre les arborescences sur  $[n]$  et les mots de longueur  $(n - 1)$  sur l'alphabet  $[n]$  [17]. Soit  $m$  un mot de  $(n - 1)$  lettres sur l'alphabet  $[n]$ . On construit une arborescence  $A_m$  à partir de  $m$  de la manière suivante :

- au départ  $A_m$  est constituée des seuls sommets  $1, \dots, m$ , sans aucune arête;

- soit  $F$  l'ensemble des lettres de  $[n]$  non présentes dans  $m$ ;
- pour  $i$  allant de 1 à  $(n - 1)$ , soit  $u = m(i)$  la  $i^{\text{ème}}$  lettre de  $m$  et  $v$  le plus grand élément de  $F$ : ajouter un arc de  $u$  vers  $v$  à  $A_m$ , supprimer  $v$  de  $F$  et si  $u$  n'apparaît plus dans  $m$ , ajouter  $u$  à  $F$ .

On peut alors faire les remarques suivantes.

- Tous les arcs sont orientés vers la racine, la dernière lettre de  $m$  étant la racine de  $A_m$ .
- Cet algorithme induit un étiquetage sur  $[n - 1]$  des arêtes de  $A_m$ : si l'arête allant de  $u$  vers  $v$  est créée lors de la  $i^{\text{ème}}$  étape, on l'étiquette par  $i$ .

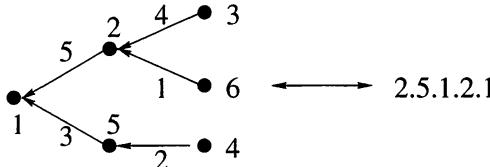


Figure 4: Arborescence et mot associés suivant le codage de Prüfer.

### 3.2 Enumération de cactus selon la distribution des couleurs

Compte-tenu de la bijection entre cactus  $m$ -aires et arborescences cactacées et de la Proposition 1, il suffit d'énumérer les endofonctions  $m$ -cactacées ayant  $n_i$  éléments de couleur  $i \in [m]$ . Nous réalisons l'énumération de ces endofonctions en trois étapes : répartition des blocs, placement des éléments minimaux (éléments  $1_i$ ) et placement des éléments non minimaux. On rappelle qu'une endofonction cactacée ayant  $n_i$  sommets de couleur  $i$  comporte exactement

$$p = \frac{(\sum_{i=1}^m n_i) - 1}{m - 1}$$

blocs, et que la fibre de l'élément 0 est constituée du seul élément  $1_1$ . Un  $i$ -bloc étant constitué de  $m - 1$  éléments de couleurs deux à deux distinctes et autres que  $c_i$ , on en déduit la propriété suivante.

Le nombre de 1-blocs est  $p - n_1 + 1$  et le nombre de  $i$ -blocs est  $p - n_i$ , pour  $i > 1$ . (8)

**1. Répartition des blocs.** Le nombre de partitions d'un ensemble de  $k$  éléments indistincts en  $l$  parts (ordonnées entre elles) éventuellement vides est

$$\binom{k + l - 1}{l - 1}. \quad (9)$$

On appelle *partition des blocs* d'une endofonction cactacée le vecteur  $(k_{1,1}, \dots, k_{1,n_1}, \dots, k_{m,1}, \dots, k_{m,n_m})$  de  $n$  entiers positifs ou nuls tel que  $k_{i,j}$  est le nombre de  $i$ -blocs présents dans la fibre de l'élément  $j_i$ . On déduit de (8) et (9) que le nombre de partitions des blocs possibles pour les endofonctions cactacées sur  $[n]$  est

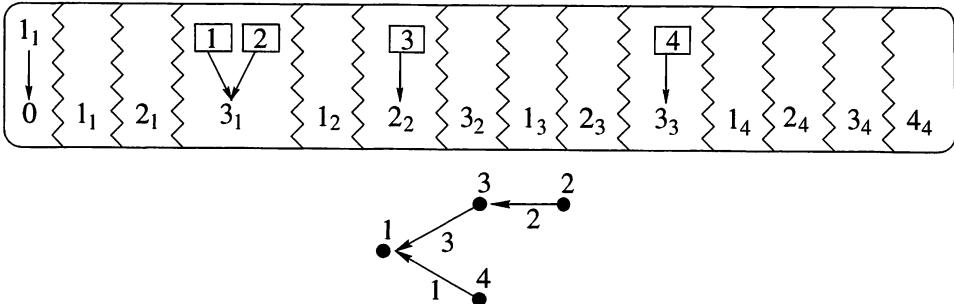
$$\binom{p}{n_1 - 1} \prod_{i=2}^m \binom{p - 1}{n_i - 1} = \frac{1}{(p - n_1 + 1)p^{m-1}} \prod_{i=1}^m n_i \binom{p}{n_i}. \quad (10)$$

Maintenant, pour une partition des blocs fixée, nous regardons le nombre de répartitions des éléments dans les blocs satisfaisant à la contrainte 2 de la Propriété 3 sur le graphe des couleurs d'une endofonction cactacée.

**2. Placement des éléments minimaux.** Rappelons que l'élément  $1_1$  étant l'unique élément présent dans la fibre de 0, il n'appartient à aucun bloc. On considère alors les éléments minimaux  $1_i, i = 2, \dots, m$ , qui sont les seuls à intervenir dans la contrainte sur le graphe des couleurs.

$$\text{Il y a } p^{m-2}(p - n_1 + 1) \text{ façons de placer les éléments } 1_i, i = 2, \dots, m. \quad (11)$$

En effet, supposons les  $p$  blocs numérotés arbitrairement de 1 à  $p$  et considérons un mot  $w$  de longueur  $(m - 1)$  sur l'alphabet  $[p]$ : à chaque lettre de  $w$  correspond un des  $p$  blocs de l'endofonction. A partir de  $w$ , on construit le mot  $w'$  en remplaçant chaque numéro de bloc par son type : si le bloc  $a$  dans  $w$  est un  $i$ -bloc, on remplace  $a$  par  $i$  dans  $w'$ . Ce mot  $w'$  est alors un mot de longueur  $(m - 1)$  sur l'alphabet  $[m]$ . En appliquant le codage de Prüfer au mot  $w'$ , on obtient une arborescence orientée sur  $[m]$  dont les arcs sont étiquetés sur  $[m - 1]$  et orientés vers la racine. On peut alors associer à cette arborescence le placement suivant des éléments minimaux : si on a un arc de  $i$  vers  $j$ , étiqueté  $k$ , on place  $1_i$  dans le bloc associé à la  $k^{\text{ième}}$  lettre de  $w$  (il est clair que ce bloc ne peut être un  $i$ -bloc). Par exemple, pour  $m = 4$  et  $\mathbf{n} = (3, 3, 3, 4)$ , on considère la partition des blocs donnée à la Figure 5 (les 4 blocs sont numérotés de 1 à 4 de gauche à droite). Si  $w = 2.4.1$ , on a alors  $w' = 1.3.1$  et le codage de Prüfer appliqué à  $w'$  donne l'arborescence apparaissant au bas de la Figure 5.



**Figure 5:** Arborescence correspondant au mot  $w = 2.4.1$

On place donc  $1_2$  dans le bloc 4,  $1_3$  dans le bloc 1 et  $1_4$  dans le bloc 2. Pour que la contrainte sur le graphe des couleurs soit vérifiée par ce placement des éléments minimaux, il faut et il suffit que sa racine soit le sommet 1, ce qui est équivalent à imposer que la dernière lettre de  $w'$  est la lettre 1 (on obtient alors une arborescence de racine 1 que l'on relie à 0 par un arc de joignant 1 et 0) et donc que le bloc associé à la dernière lettre de  $w$  est un 1-bloc. On a donc  $p$  choix pour chacune des  $(m - 2)$  premières lettres de  $w$  et  $(p - n_1 + 1)$  pour sa dernière lettre, ce qui prouve (11).

**3. Placement des éléments non minimaux.** Une fois que les éléments minimaux sont placés, la répartition des éléments restants dans les blocs ne pose aucun problème. Par (10), (11), la Remarque 1, et le fait qu'il y a  $(\mathbf{n} - 1)!$  façons de répartir les éléments non minimaux, et après simplification, on obtient la formule (1).

### 3.3 Enumération de cactus selon la distribution des degrés

La preuve du Théorème 2 est basée sur le même principe que la preuve précédente. Si on appelle degré d'un élément d'une endofonction cactacée le nombre de blocs présents

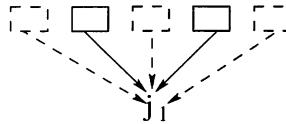
dans sa fibre, par la bijection entre cactus  $m$ -aires et arborescences cactacées, la Proposition 1, la Propriété 1 de conservation de la structure des fibres et la Propriété 2, l'ensemble des cactus  $m$ -aires de partition des degrés  $D$  et de racine de degré  $l$  (le sommet de couleur 1 du polygone pointé est incident à  $l$  polygones) est en bijection avec l'ensemble des endofonctions cactacées ayant

- (a)  $d_{i,j}$  éléments de couleur  $i$  et de degré  $j - 1$  si  $i > 1$  ou ( $i = 1$ ,  $j \neq l$  et  $j \neq (l + 1)$ ),
- (b)  $d_{1,l} - 1$  éléments de couleur 1 et de degré  $l - 1$  et
- (c)  $d_{1,l+1} + 1$  éléments de couleur 1 et de degré  $l$ .

*1.a. Répartition des  $i$ -blocs pour  $i > 1$ .* Pour chaque couleur  $i > 1$ , on a donc  $d_{i,j}$  éléments de degré  $(j - 1)$  et le nombre de façons de répartir ces  $i$ -blocs dans les fibres des  $n_i$  sommets de couleur  $i$  est

$$\binom{n_i}{d_{i,1}, d_{i,2}, \dots}. \quad (12)$$

*1.b. Répartition et pointage des 1-blocs.* En ce qui concerne la couleur 1, après avoir réparti  $d_{1,j}$  1-blocs, pour tous les  $j > 0$ , il découle de (b) et (c) qu'il reste encore un 1-bloc à placer dans la fibre d'un élément de couleur 1. Comme le montre la Figure 6 (dans laquelle les blocs en pointillés représentent les positions possibles d'insertion d'un bloc supplémentaire), on a exactement  $k + 1$  façons de placer ce 1-bloc dans la fibre d'un élément de degré  $k$  en le distinguant des autres blocs déjà placés,



et on donc a exactement

$$n_1 + \sum_{j \geq 1} (j - 1)d_{1,j} - 1 = n_1 + (p - n_1 + 1) - 1 = p \quad (13)$$

façons de placer un bloc distingué dans la fibre d'un élément de couleur 1. On en déduit qu'on a

$$p \prod_{i=1}^m \binom{n_i}{d_{i,1}, d_{i,2}, \dots}. \quad (14)$$

façons de placer les  $p$  blocs tout en distinguant un 1-bloc.

*2. Placement des éléments minimaux.* De la même manière que pour la preuve précédente, on remarque que le nombre de placements des éléments minimaux  $1_i$  ( $i > 1$ ) respectant la contrainte sur le graphe des couleurs est donné, pour toute numérotation arbitraire des blocs sur  $[p]$ , par le nombre de mots de longueur  $(m - 1)$  sur l'alphabet  $[p]$  tels que la dernière lettre est le numéro du 1-bloc distinguée. Il y en a exactement

$$p^{m-2}. \quad (15)$$

*3. Placement des éléments non minimaux.* Finalement, comme précédemment, le placement des éléments non minimaux ne pose aucun problème et on en a  $(n - 1)!$ , ce qui, combiné à (14), (15) et à la Remarque 1, et après simplification, donne la formule (2).

## 4 Génération aléatoire de cactus $m$ -aires

Les preuves présentées dans la section précédente, outre le fait qu'elles donnent la première explication combinatoire, à notre connaissance, des formules d'énumération de cactus  $m$ -aires, permettent aussi la mise au point d'algorithmes efficaces de génération aléatoire uniforme de cactus  $m$ -aires, selon la distribution des couleurs ou des degrés.

Pour énumérer les endofonctions cactacées selon la partition des couleurs ou des degrés, nous avons suivi un cheminement en trois étapes qui sous-tendent les grandes lignes d'un algorithme de génération aléatoire de cactus  $m$ -aires. Pour engendrer aléatoirement et uniformément une telle structure selon la distribution des couleurs (resp. des degrés), on peut alors appliquer l'Algorithme 1 suivant.

---

### **Algorithme 1:** Génération aléatoire de cactus $m$ -aire

---

**Données :** distribution  $\mathbf{n}$  (resp.  $D$ ) des couleurs (resp. des degrés)

**Résultat :** cactus  $m$ -aire

**début**

génération d'une endofonction cactacée ayant distribution des couleurs  $\mathbf{n}$  (resp. des degrés  $D$ );

1. partition des blocs;
2. placement des éléments minimaux;
3. placement des éléments non minimaux;

application de la bijection  $\Phi$  entre endofonctions cactacées et arborescences cactacées;

application de la bijection  $\Psi$  entre arborescences cactacées et cactus  $m$ -aires;

**fin**

---

**Lemme 1.** *L'algorithme 1 permet d'engendrer aléatoirement et uniformément un cactus  $m$ -aire enraciné à  $n$  sommets selon la distribution des couleurs ou des degrés, en temps et espace  $O(n)$ .*

*Preuve.* D'après les preuves des Théorèmes 1 et 2, il est clair que cet algorithme engendre un cactus  $m$ -aire. Il reste à vérifier que cette génération est uniforme et que les complexités en temps et en espace sont linéaires.

**1. Génération uniforme.** Le fait que cet algorithme engendre uniformément les cactus  $m$ -aires, c'est-à-dire que tous les cactus de distribution des couleurs  $\mathbf{n}$  (resp. des degrés  $D$ ) ont la même probabilité d'être engendré, découle du fait que lors de l'étape de génération aléatoire des endofonctions cactacées :

- toutes les partitions des blocs ont la même probabilité d'être engendrées,
- le nombre de façons de placer les éléments minimaux ne dépend pas de la partition des blocs (et tous les placements sont donc équiprobables),
- le nombre de façons de placer les éléments non minimaux ne dépend ni de la partition des blocs, ni du placement des éléments minimaux (et tous les placements sont donc équiprobables).

**2. Complexité linéaire en temps et en espace.** Dans les deux cas, la phase de répartition des blocs consiste à partitionner l'ensemble des  $i$ -blocs en  $n_i$  parts, ce qui se fait en temps et espace linéaires. Le placement des éléments minimaux consiste à choisir un mot aléatoirement sur un alphabet fixé, puis à appliquer le codage de Prüfer, ce qui là aussi ce fait en temps et espace linéaires. Le placement des éléments non minimaux

consiste à choisir aléatoirement une permutation sur les éléments non minimaux, ce qui là aussi se fait en temps et espace linéaires. Finalement, les bijections  $\Phi$  et  $\Psi$  pouvant être implantées en temps et espace  $O(n)$ , on obtient bien une complexité linéaire.  $\square$

*Remarque 2.* Récemment, Dimitri Zvonkine [18] a proposé une autre preuve combinatoire du Théorème 2.

**Remerciements.** Nous remercions Pierre Leroux pour de très utiles conversations.

## Références

1. Bender (E. A.) et Richmond (L. B.). – A multivariate Lagrange inversion formula for asymptotic calculations. *Electron. J. Combin.*, vol. 5, n° 1, 1998, pp. Research Paper 33, 4 pp.
2. Bender (E. A.) et Richmond (L. B.). – Multivariate asymptotics for products of large powers with applications to Lagrange inversion. *Electron. J. Combin.*, vol. 6, n° 1, 1999, pp. Research Paper 8, 21 pp.
3. Bergeron (F.), Labelle (G.) et Leroux (P.). – *Combinatorial species and tree-like structures*. – Cambridge, Cambridge University Press, 1998.
4. Bóna (M.), Bousquet (M.), Labelle (G.) et Leroux (P.). – Enumeration of m-ary cacti. *Adv. in Appl. Math.*, vol. 24, 2000, pp. 22–56.
5. Bousquet (M.). – *Théorie des espèces et applications au dénombrement de cartes et de cactus planaires*. – Thèse de Doctorat, LaCIM, Université du Québec à Montréal, 1999.
6. Bousquet (M.), Chauve (C.), Labelle (G.) et Leroux (P.). – A bijective proof of the arborescent form of the multivariable Lagrange's inversion formula. In : *Colloquium on Mathematics and Computer Science*, Trends in Mathematics. – Birkhäuser, 2000. (à paraître).
7. Chauve (C.). – *Structures arborescentes : problèmes algorithmiques et combinatoires*. – Thèse de Doctorat, LaBRI, Université Bordeaux I, 2000. (en préparation).
8. El Marraki (M.), Hanusse (N.), Zipperer (J.) et Zvonkin (A.). – Cacti, braids and complex polynomials. *Sém. Lothar. Combin.*, vol. 37, 1996, pp. Art. B37b, 36 pp.
9. Good (I. J.). – Generalizations to several variables of Lagrange's expansion, with applications to stochastic processes. *Proc. Cambridge Philos. Soc.*, vol. 56, 1960, pp. 367–380.
10. Goulden (I. P.) et Jackson (D. M.). – *Combinatorial enumeration*. – New York, John Wiley & Sons Inc., 1983.
11. Goulden (I. P.) et Jackson (D. M.). – The combinatorial relationship between trees, cacti and certain connection coefficients for the symmetric group. *European J. Combin.*, vol. 13, n° 5, 1992, pp. 357–365.
12. Goulden (I. P.) et Kulkarni (D. M.). – Multivariable Lagrange inversion, Gessel-Viennot cancellation, and the matrix tree theorem. *J. Combin. Theory Ser. A*, vol. 80, n° 2, 1997, pp. 295–308.
13. Harary (F.) et Palmer (E. M.). – *Graphical enumeration*. – New York, Academic Press, 1973.
14. Harary (F.) et Uhlenbeck (G. E.). – On the number of Husimi trees. *Proc. Nat. Acad. Sci. U.S.A.*, vol. 39, 1953, pp. 315–322.
15. Husimi (K.). – Note on Mayer's theory of cluster integrals. *J. Chem. Phys.*, vol. 18, 1950, pp. 682–684.
16. Labelle (G.). – Une nouvelle démonstration combinatoire des formules d'inversion de Lagrange. *Adv. in Math.*, vol. 42, n° 3, 1981, pp. 217–247.
17. Moon (J. W.). – *Counting labelled trees*. – Canadian Mathematical Congress, Montréal, Québec, 1970. Canadian Mathematical Monographs, No. 1.
18. Zvonkine (D.). Communication personnelle.

# Algorithmes géométriques appliqués à la manipulation de tissus dans un environnement virtuel \*

Yves Chiricota<sup>†</sup>

*Département d'informatique et de mathématiques, Université du Québec à Chicoutimi,  
555, boul. de l'Université, Chicoutimi (Québec), Canada, G7H 2B1*

## Abstract

This article presents geometrical methods for interactive modelling and displacement of polygon meshes in three dimensions. Our main algorithm is based in a breath first search in the graph associated to a polygon mesh. The edges are sorted and their lengths corrected as to produce natural movement of the mesh, whenever it is manipulated by an user. The main application of this work is related to CAD systems used in garment industry.

## Résumé

Ce travail présente une méthode géométrique de déplacement d'un maillage dans un environnement virtuel. Le principal algorithme est basé sur un tri des arêtes du maillage permettant d'effectuer un déplacement par correction géométrique de la longueur des arêtes. Une des applications de cette méthode est la manipulation de surfaces représentant des tissus dans un environnement virtuel. Cette application est axée sur les systèmes de conception de vêtements assistés par ordinateur.

Mots clefs: Algorithme, géométrique, simulation, vêtement, virtuel, modèle, déformable

## 1 Introduction

Le contexte dans lequel se situe ce travail est celui de la simulation d'objets déformables en informatique. Il porte plus particulièrement sur la simulation de surfaces discrètes (maillages) servant à représenter des tissus. Les résultats concernant les objets déformables reposent en grande partie sur des algorithmes de simulation physique. Terzopoulos [10] et al. ont introduit des méthodes de simulation physique d'objets déformables basées sur la théorie de l'élasticité. On procède en calculant la solution d'une équation différentielle décrivant la position d'une surface en fonction du temps. On retrouve dans Breen *et al.* [1, 2] un algorithme de simulation basé sur la minimisation d'une fonction décrivant l'énergie d'un système représentant une surface discrétisée. Certains travaux portent sur la manipulation et l'édition interactive de surfaces. Mentionnons ceux de Welch *et al.* [12, 13], de Fiume *et al.* [5], ainsi que ceux de Heckbert *et al.* [7], tous basés sur des méthodes variationnelles et sur des simulations physiques. Une application directe de ces travaux est la représentation de surfaces servant à modéliser des tissus dans un environnement virtuel. Les travaux de Volino *et al.* [11] se situent aussi dans ce contexte. La figure 1 donne un exemple de modèle calculé par simulation de tissus dans un système de conception de vêtements assistée par ordinateur<sup>1</sup>.

\*Travail réalisé grâce au soutien financier du programme PAIR, UQAC

<sup>†</sup>Courriel: Yves.Chiricota@uqac.quebec.ca

<sup>1</sup>Développé par PAD System Technologies, Montréal

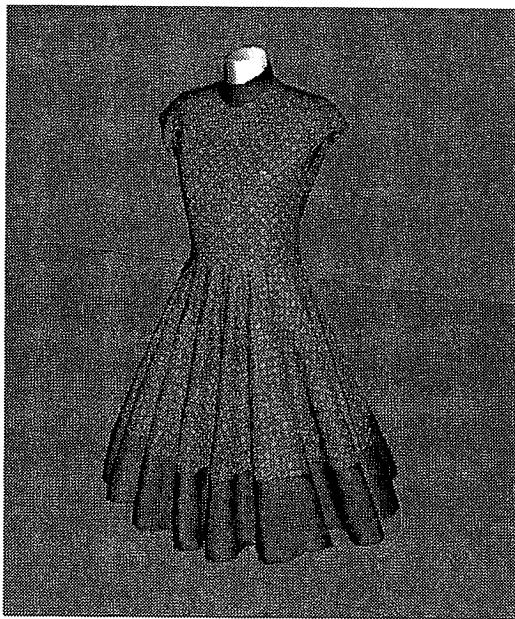


Figure 1:

Les méthodes de simulation physique demandent un temps de calcul relativement long, ce qui n'est pas toujours pratique dans le contexte de la manipulation en temps réel de surfaces représentées par des maillages, contexte dans lequel se situe notre travail. De plus, ces méthodes peuvent produire un effet d'élasticité qui n'est pas toujours désirable. Une partie de l'élasticité provient de la méthode d'intégration numérique utilisée pour résoudre l'équation décrivant le mouvement de la surface. Il est possible de diminuer le pas d'intégration pour obtenir un résultat moins élastique, au détriment du temps de calcul.

Provot [9] a introduit une méthode de correction géométrique pour diminuer l'effet d'élasticité produit par la méthode d'intégration. L'idée consiste à appliquer, durant la simulation, une contraction (resp. une dilatation) aux arêtes d'un maillage qui s'étirent trop (resp. se contractent trop), ramenant celles-ci à leur longueur "au repos". Cette méthode permet d'augmenter le pas d'intégration avec la méthode d'Euler. Desbrun *et al.* [4] ont utilisé cette méthode avec un schéma d'intégration semi-implicite dans un environnement virtuel. Leur système permet la manipulation en temps réel de surfaces discrétisées. Toutefois, le temps de calcul inhérent à la simulation physique représente une partie importante du temps total.

Le principal algorithme présenté ici est une variante de la méthode de correction géométrique de Provot. Il permet la manipulation d'une surface sans effectuer de simulation physique, entraînant ainsi un gain en temps de calcul. Cet algorithme a donné lieu à un prototype logiciel grâce auquel un utilisateur peut déplacer en temps réel une surface discrétisée sous la forme d'un maillage. L'algorithme débute par le calcul d'un maillage par *grille* d'une courbe polygonale représentant une pièce de tissu. Ensuite, il effectue un tri des arêtes du graphe simple associé au maillage. L'ordre de parcours des arêtes dépend du point servant à déplacer le maillage. Durant le déplacement, les arêtes sont ramenées à leur longueur naturelle selon l'ordre établit par ce tri. Le déplacement

du maillage se fait dans une direction établie par l'utilisateur via l'interface graphique du logiciel. L'effet d'élasticité lors du déplacement est acceptable<sup>2</sup>.

## 2 Mailages

### 2.1 Graphes et mailages

Établissons d'abord quelques notations concernant les graphes (voir aussi [6],[8]). Soit  $X$  un ensemble fini de *sommets*. Un graphe simple  $G$  sur  $X$  est un couple  $(X, A)$ , où  $A \subset \mathcal{P}_2(X)$  est un ensemble d'*arêtes* ( $\mathcal{P}_2$  désigne l'ensemble des parties de  $X$  à deux éléments). Un graphe orienté sur  $X$  est un couple  $(X, U)$ , où  $U \subset X \times X$  est un ensemble de *flèches*. Si  $Y \subset X$ , l'ensemble  $\{x \in X : \exists \{x, y\} \in A, x \in X \setminus Y \text{ et } y \in Y\}$  est appelé l'ensemble des *voisins* de  $Y$  et est dénoté par  $\mathcal{V}(Y)$ .

Soit  $t$  un triangle dans  $\mathbf{E}^3$  (l'espace euclidien de dimension 3). L'intérieur de  $t$  sera dénoté par  $\text{int}(t)$ . Un *maillage*  $M$  de deux dimensions dans  $\mathbf{E}^3$  est un polyèdre formé de triangles  $T = \{t_1, t_2, \dots, t_n\}$  satisfaisant les propriétés suivantes:

- $\forall i, j, i \neq j \Rightarrow \text{int}(t_i) \cap \text{int}(t_j) = \emptyset$ ,
- $\forall i, j, \text{ si } i \neq j \text{ alors un et un seul des cas suivant est possible: } t_i \cap t_j = \emptyset, t_i \cap t_j \text{ est un point, ou encore } t_i \cap t_j \text{ est un coté commun aux deux triangles.}$

Considérons l'ensemble  $X$  des points formé par les sommets des triangles de  $M$  et l'ensemble  $A$  des côtés de ces triangles. Le couple  $(X, A)$  détermine un graphe simple. L'ensemble des arêtes  $A$  est formé des paires  $\{x, y\}$  telles que  $x$  et  $y$  sont sommets d'un même triangle.

Dans ce qui suit, nous ne ferons pas la distinction entre un maillage et son graphe associé  $(X, A)$ , lorsque le contexte sera clair. Chaque triangle  $t$  du maillage  $M$  correspond à un sous-graphe de  $(X, A)$ . Ce sous-graphe est complet d'ordre 3. Soit  $H$  un sous-graphe de  $(X, A)$  obtenu par union de triangles. La *frontière* de  $H$  est formée des arêtes correspondant à la frontière des triangles d'origine dans le maillage.

A tout graphe (ou maillage)  $(X, A)$ , on peut associer un sous-graphe orienté  $\Gamma = (Y, U)$  avec  $Y \subset X$  et  $(x, y) \in U \Rightarrow \{x, y\} \in A$ . On dira que le sous-graphe  $\Gamma$  *recouvre*  $M$  si  $Y = X$  et  $\Gamma$  est connexe (si on ne tient pas compte de l'orientation des arêtes). Si c'est le cas, tout sommet de  $X$  est incident à une arête de  $\Gamma$ .

Le *graphe dual*  $(X^*, A^*)$  associé à un maillage est obtenu en prenant l'ensemble des triangles comme ensemble de sommets et en posant  $\{t_1, t_2\} \in A^*$  si et seulement si  $t_1$  et  $t_2$  ont un côté commun. Un *bande* dans le graphe associé à un maillage est un sous-graphe dont le graphe dual est une chaîne.

Un sous-graphe *séparateur*  $\Sigma$  dans un graphe simple  $(X, A)$  (correspondant à un maillage  $M$ ) est soit un seul point, soit un sous-graphe connexe  $(X', A')$  tel que  $A'$  est obtenu par l'union d'arêtes et de triangles. De plus, ce sous-graphe doit satisfaire les propriétés suivantes: (1) le graphe dual de  $\Sigma$  est formé d'un ensemble de chaînes et (2) le graphe obtenu de  $(X, A)$  en retirant les sommets appartenant à  $X'$  et les arêtes incidentes a exactement deux composantes connexes (dont l'une peut être vide). On obtient ainsi une partition de  $X \setminus X'$  en une ou deux classes. Le *rang* d'un sous-graphe séparateur est la longueur maximale des chaînes dans le graphe dual de  $\Sigma$ .

Le graphe de la figure 2 contient trois sous-graphes séparateurs. Les triangles appartenant à ces derniers sont dessinés en gris. Le rang de ces sous-graphes séparateurs est respectivement 0, 2 et 3.

---

<sup>2</sup>L'élasticité varie d'environ 2% à 5%.

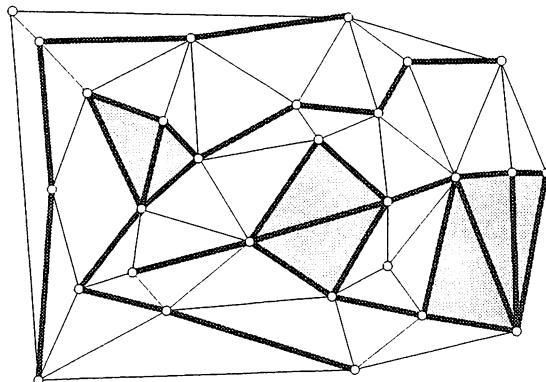


Figure 2: Sous-graphes séparateurs.

## 2.2 Maillage par grille de courbes polygonales

Dans les systèmes de conception assistée par ordinateur, les pièces servant à créer des vêtements sont généralement représentées par des courbes polygonales simples fermées. On retrouvera ici une technique de triangulation permettant de calculer un maillage à partir d'une courbe polygonale. Notons qu'il existe des méthodes classiques pour effectuer cette tâche. Nous présentons une technique adaptée à la manipulation géométrique de maillages par l'algorithme décrit à la section 4.

Fixons  $d \in \mathbf{R}$ . Considérons l'ensemble des points de la forme  $p_{i,j} = (id, jd)$  avec  $i, j \in \mathbf{Z}$ . Sur cet ensemble, formons le maillage  $\Omega$  dont les triangles sont déterminés par les points

$$\begin{aligned} p_{i,j}, p_{i,j+1}, p_{i+1,j+1} \text{ et } p_{i,j}, p_{i+1,j+1}, p_{i+1,j} &\quad \text{si } i + j \equiv 0 \pmod{2} \\ p_{i,j}, p_{i,j+1}, p_{i+1,j} \text{ et } p_{i+1,j}, p_{i,j+1}, p_{i+1,j+1} &\quad \text{si } i + j \equiv 1 \pmod{2}. \end{aligned}$$

Il est évident que ce maillage est de Delaunay (on trouvera une définition de maillage de Delaunay dans [3]).

Soit  $C$  une courbe polygonale plane simple fermée. Un *maillage par grille* de  $C$  est obtenu en conservant la partie du maillage  $\Omega$  à l'intérieur de  $C$  et en complétant le maillage en ajoutant à celui-ci les points de la courbe  $C$ , de façon à avoir un maillage de Delaunay.

La figure 3 illustre le calcul d'un maillage par grille d'une courbe polygonale. Dans partie (a), on retrouve la partie de la grille à l'intérieur de la courbe  $C$  et la partie (b) est la triangulation Delaunay résultante. Notons que certaines arêtes du maillage initial peuvent être remplacées lorsqu'on calcule le maillage de Delaunay final, comme c'est le cas dans la figure 3.

## 3 Algorithme de tri

L'algorithme de tri des sommets est basé sur un parcours en largeur du graphe, à partir d'un sommet source  $s$ . On construit itérativement un graphe orienté  $\Gamma$  recouvrant  $M$ . Ce graphe orienté est sans cycle et est tel que pour tout sommet  $x$ , il existe un chemin qui va de  $x$  à  $s$ . Cet algorithme produit une suite de sommets  $S_k$ , de flèches  $F_k$ , d'arêtes  $C_k$  et de sous-graphes  $\Sigma_k = (S_k, C_k)$ . L'ensemble  $S_k$  est formé des sommets dont la

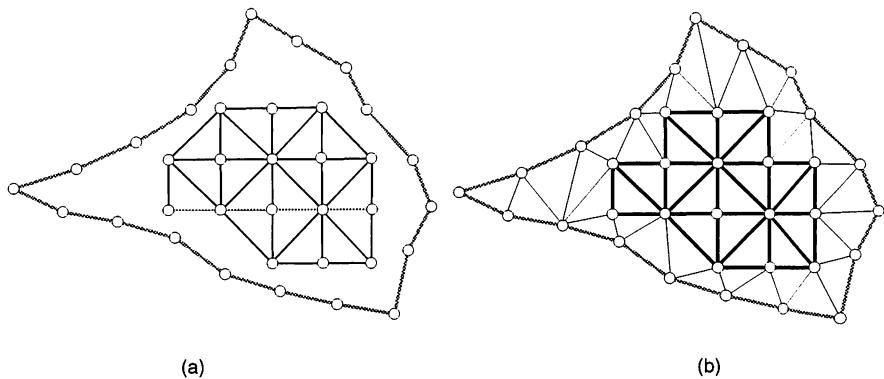


Figure 3: Maillage par grille d'une courbe polygonale.

distance relativement au sommet source est  $k$ . Les flèches de  $F_k$  relient des sommets de  $S_{k-1}$  avec ceux de  $S_k$  de telle sorte que tout sommet de  $S_k$  est relié à au moins un sommet de  $S_{k+1}$  et réciproquement.

**Algorithme 3.1** Soit  $M$  un maillage,  $(X, A)$  son graphe associé et  $s \in X$  un sommet.  
Poser  $S_0 = \{s\}$ ,  $C_0 = \emptyset$ ,  $F_0 = \emptyset$ .

Par suite, pour  $k \geq 1$ ,

$$\begin{aligned} S_k &= V(S_0 \cup S_1 \cup \dots \cup S_{k-1}), \\ F_k &= \{(x, y) : x \in S_k, y \in S_{k-1} \text{ et } \{x, y\} \in A\}, \\ C_k &= \{\{x, y\} \in A : x \in S_k \text{ et } y \in S_k\}. \\ \Sigma_k &= (S_k, C_k). \end{aligned}$$

Construire itérativement les ensembles  $S_k, F_k$  et  $C_k$  jusqu'à ce que  $S_k = \emptyset$ . Poser ensuite  $\Gamma = (S_0 \cup S_1 \cup \dots, F_0 \cup F_1 \cup \dots)$ .

Il facile de voir qu'il existe  $N \in \mathbb{N}$  tel que  $\forall n \in \mathbb{N}, n \geq N \Rightarrow S_n = \emptyset$  et  $\forall n \in \mathbb{N}, n < N \Rightarrow S_n \neq \emptyset$ . Il s'en suit que l'algorithme se termine.

On dénotera par  $\delta^+(\Sigma_k)$  l'ensemble de chaînes formées par les arêtes qui constituent la partie de la frontière de  $\Sigma_k$  "visible" à partir de  $\Sigma_{k-1}$ .

La figure 4 illustre l'algorithme. Les chaînes  $\delta^+(\Sigma_k)$  sont représentées en pointillé.

#### Remarques:

1. Tous les sommets de  $\Sigma_k$  sont la source d'au moins une arête de  $F_k$ .
2. Le graphe  $\Gamma$  est connexe (si on ne tient pas compte de l'orientation des arêtes), orienté, sans cycle avec le sommet  $s$  comme puits.
3. Chaque sous-graphe  $\Sigma_k = (S_k, C_k)$  est formé d'un ensemble de sous-graphes séparateurs de  $M$ .
4. L'ensemble  $\{S_0, S_1, \dots, S_{n-1}\}$  forme une partition de  $X$ .
5. Si  $\delta^+(\Sigma_k)$  est convexe, alors  $\Sigma_k$  est une chaîne.
6. Les triangles se situant entre deux sous-graphes séparateurs consécutifs forment des bandes.

La figure 2 a été obtenue en appliquant l'algorithme à partir du sommet du coin supérieur gauche. On retrouvera les ensembles d'arêtes  $F_k$  en orientant les arêtes dessinées avec des traits fins vers le sommet en question.

Pour un maillage par grille, on a la proposition suivante.

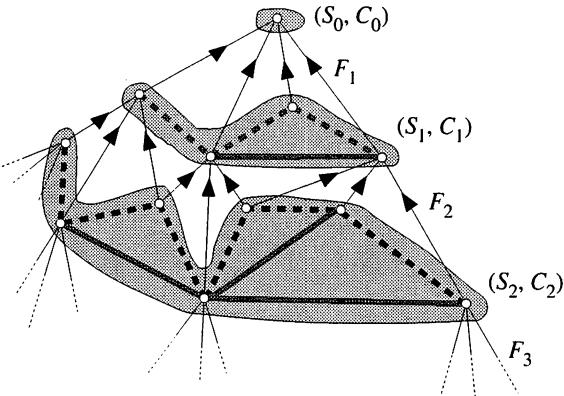


Figure 4: Algorithme de tri

**Proposition 3.1** Si on applique l'algorithme 3.1 sur un maillage par grille, les propriétés suivantes sont vérifiées.

- Le sous-graphe  $\Sigma_k = (S_k, C_k)$  est séparateur et de rang 0 ou 1.
- Le graphe  $\Gamma$  détermine un ordre partiel sur les sommets de  $M$ . Cet ordre partiel est obtenu en considérant la suite  $F_1, F_2, \dots$

La figure 5 illustre l'application de l'algorithme de tri sur un maillage par grille. Le sommet en gras est le sommet source. Les sous-graphes séparateurs sont en gras. Chaque arête en trait fin fait partie d'un des ensemble  $F_k$ , et elle est orientée vers le sommet source.

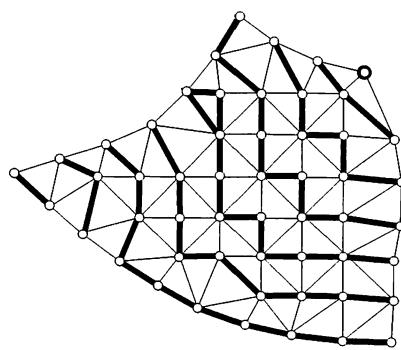


Figure 5:

Le calcul de l'ensemble  $C_k$  se fait en temps  $c|S_k|$ . Le coût du calcul des ensembles  $F_k$  est  $O(|A|)$  et celui du calcul des ensembles  $S_k$  est  $O(|X|)$ . Le coût total est donc  $O(|X| + |A|)$ .

## 4 Applications à la manipulation interactive de maillages

Cette section présente une application de l'algorithme de tri. L'algorithme permet à un utilisateur d'effectuer le déplacement d'un maillage de manière interactive. L'implémentation de l'algorithme a été réalisée dans un prototype logiciel. Un outil graphique permet à l'utilisateur de choisir un point source et de déplacer ce point.

### 4.1 Déplacement géométrique d'un maillage

Lorsqu'on calcule un maillage à partir d'une courbe polygonale, la longueur initiale d'une arête est appelée sa *longueur au repos*. Un *mouvement* d'un maillage est défini par une application qui associe à chaque point  $x$  du maillage, un vecteur établissant une translation de  $x$ . Lorsqu'on effectue un mouvement de maillage, la longueur des arêtes se trouve modifiée. L'algorithme de correction géométrique d'un maillage consiste à appliquer, suite à un mouvement, une homothétie à chaque arête de telle sorte que la longueur actuelle (après le mouvement) de l'arête soit ramenée à sa longueur au repos. Puisque les arêtes sont définies par les points du maillage, la correction d'une arête modifie la longueur des arêtes voisines. En pratique, après avoir effectué la correction géométrique, la différence de longueur est pratiquement imperceptible pour l'utilisateur.

Pour une arête  $\{x, y\}$  ou une flèche  $(x, y)$  (avec  $x, y \in \mathbf{E}^3$ ) posons  $\lambda = \frac{l}{|x-y|}$ , où  $l$  est la longueur au repos. On suppose qu'aucun mouvement est tel qu'on puisse avoir  $x = y$ . Deux types d'homothéties sont utilisées dans l'algorithme de déplacement de maillages:

- Pour une flèche  $(x, y)$ , on pose  $(x, y) \mapsto (\lambda x + (1 - \lambda)y, y)$ .
- Pour une arête  $\{x, y\}$ , on pose  $\{x, y\} \mapsto \{\frac{\lambda}{2}(x - y), \frac{\lambda}{2}(y - x)\}$ .

Une arête est donc dilatée relativement à son centre. Une flèche est dilatée relativement au sommet d'arrivée.

Soit  $s$  un point dans un maillage  $M$  et  $\gamma = (v_0, v_1, \dots, v_n)$  une suite de vecteurs ( $v_i \in \mathbf{R}^3$ ). Le point  $s$  et la suite  $\gamma$  définissent un *chemin* dans  $\mathbf{E}^3$ . Les positions du chemin sont  $q_k = s + \sum_{j=0}^k v_i$ . L'algorithme suivant effectue le déplacement du maillage  $M$  le long du chemin  $\gamma$  en "tirant" sur point  $s$ .

#### Algorithme 4.1 Algorithme de déplacement

- Appliquer l'algorithme 3.1 au maillage  $M$  avec le sommet  $s$  comme source.
- Pour  $i = 0, 1, \dots, n$ :
  - Poser  $s = s + v_i$ .
  - Pour  $k = 0, 1, \dots$ , tant que  $S_k \neq \emptyset$ :
    - Appliquer une homothétie de type A aux flèches de  $F_k$
    - Appliquer une homothétie de type B aux arêtes de  $C_k$

**Remarque:** Pour un maillage par grille, la déformation lors d'un déplacement donne un résultat réaliste (qui s'apparente à ce qu'on obtiendrait si on tirant sur du vrai tissu). Ceci est lié au fait que les flèches appartenant aux ensembles  $F_k$  sont orientées approximativement dans l'axe de déplacement du point  $s$  et que les arêtes des ensembles  $C_k$  sont à peu près perpendiculaires à ce déplacement.

La complexité de cet algorithme est  $O(|X| + |A|)$ . De plus, puisque le maillage est trié une seule fois à l'étape 1, le coût d'un mouvement isolé est  $O(|A|)$ . Le mouvement (étape 2) s'effectue en temps linéaire qui dépend seulement du nombre d'arêtes.

Si nous avions défini le déplacement à partir d'une force artificielle qui attire le sommet  $s$  dans une direction donnée durant une simulation, il y aurait un effet de retour en arrière du point  $s$ , due à l'élasticité. Ce retour en arrière se produit lorsque la force cesse d'être appliquée à la fin du déplacement. L'algorithme de correction géométrique simple ne corrige pas cet effet puisque seules des corrections de type A sont appliquées aux arêtes de  $M$ . La proposition suivante nous assure que l'algorithme de déplacement n'introduit pas d'effet de retour en arrière du point source.

**Proposition 4.1** *Les étapes 2.2, 2.2.1 et 2.2.2 de l'algorithme de déplacement ne modifient pas la position du point source.*

Notons qu'en pratique le chemin  $\gamma$  est déterminé par les mouvements de la souris dans le logiciel.

## 5 Conclusion

Les algorithmes présentés ici permettent la manipulation de surfaces déformables représentant des tissus dans un environnement 3D. Ces algorithmes offrent un compromis entre la rapidité d'exécution et le réalisme que donne une simulation physique. L'implémentation a été effectuée dans un prototype logiciel par le biais d'un outil graphique permettant de tirer sur un des points d'un maillage.

Le temps de calcul de déplacement a été mesuré empiriquement pour le maillage comportant environ 65 points de la figure 6. Chaque mouvement se fait en environ  $\frac{1}{10}$  secondes si on applique l'algorithme géométrique. Le calcul d'un pas d'intégration avec la méthode d'Euler semi-implicite prend environ  $\frac{2}{10}$  secondes, pour un déplacement équivalent.

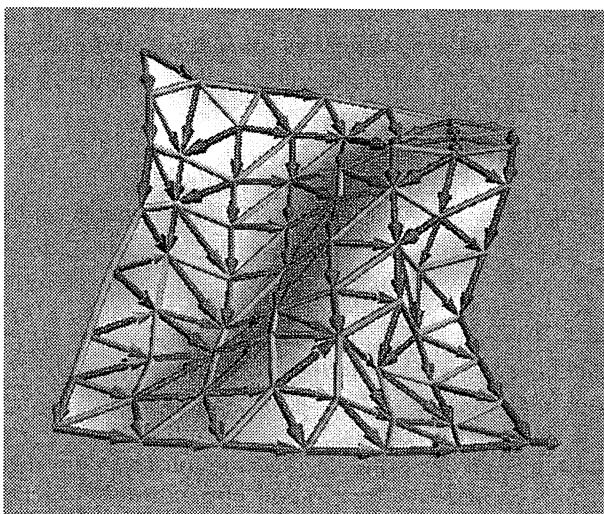


Figure 6: Déplacement d'un maillage.

Ce travail est étroitement lié à la conception d'interfaces graphiques. Une avenue de recherche porte sur l'introduction d'éléments d'interface servant à manipuler des tissus en trois dimensions. Par exemple, il est possible de modifier l'algorithme de tri de façon à ce que  $S_0$  contiennent plusieurs points du maillage. Cette modification pourrait permettre la conception d'outils d'interfaces plus variés pour manipuler les surfaces. La principale application de notre recherche est la conception d'un système interactif de couture dans un environnement virtuel.

## 6 Remerciements

Je tiens à remercier François Lemieux pour les commentaires judicieux dont il m'a fait part suite à la lecture du manuscrit.

## References

- [1] D.E. Breen, R.W. DeVaul, and D.H. House. Towards simulating cloth dynamics using interacting particles. *Int. Journal of Clothing Sc. and Tech.*, 8(3):75–94, 1996.
- [2] D.E. Breen, D.H. House, and M.J. Wozny. Predicting the drape of woven cloth using interacting particles. *Computer Graphics*, 28:365–372, 1994.
- [3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 1997.
- [4] M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Proceedings of Graphics Interface '99*, pages 1–8, 1999.
- [5] E. Fiume and K. Singh. Wires: a geometric deformation technique. *Computer Graphics Proceedings*, pages 405–414, 1998.
- [6] F. Harary. *Graph Theory*. Addison-Wesley, Reading, 1972.
- [7] P. Heckbert and W. Welch. Using particles to sample and control implicit surfaces. *Computer Graphics Proceedings*, 28:269–277, 1994.
- [8] J. Labelle. *Théorie des graphes*. Modulo Éditeur, Montréal, 1981.
- [9] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Proc. of Graphics Interface*, pages 147–154, 1995.
- [10] D. Terzopoulos, J.C. Platt, H. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):42–50, 1987.
- [11] P. Volino, N. Magnenat Thalmann, S. Jianhua, and D. Thalmann. The evolution of a 3d system for simulating deformable clothes on virtual actors. *Computer Graphics and Applications*, 16(5):205–214, 1996.
- [12] W. Welch and A. Witkin. Variational surface modeling. *Computer Graphics Proceedings*, 26:157–166, 1992.
- [13] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. *Computer Graphics Proceedings*, 28:247–256, 1994.

# Towards the inverse of a word

Bob Clarke \*

Pure Mathematics Department  
 The University of Adelaide  
 Adelaide, South Australia 5005

## Abstract

Let  $w$  be a word, i.e., a permutation with repetitions. We present a construction for the inverse of  $w$  in the case where  $w$  has at most three distinct letters, and indicate steps towards the construction in the general case.

## Résumé

Soit  $w$  un mot (avec répétitions). On présente ici une construction pour l'inverse de  $w$  au cas où  $w$  a au plus trois lettres distinctes. On indique également des étapes vers la construction dans le cas général.

## 1 Introduction

The starting point of this investigation is an article by Dominique Foata and Guo-Niu Han, [2], in which the authors investigate the notion of the inverse of a word. They present four properties that such an inverse should satisfy and show that there exists a transformation that satisfies all of these properties. However, they leave the construction of the transformation as an open problem. In section 1 below we summarize Foata and Han's investigation, and in section 2 solve the problem for all words on fewer than four distinct letters. In section 3 we indicate some steps towards a complete solution. The constructions introduced in sections 2.1 and 3.1 are perhaps of independent interest.

### 1.1 Words

Consider the alphabet  $\mathcal{X} = [r] = \{1, 2, \dots, r\}$ .

A *word*  $w = x_1 x_2 \cdots x_n$  on  $\mathcal{X}$  is a finite string of not-necessarily distinct elements of  $\mathcal{X}$ . The *rearrangement class*  $R(w)$  of a word  $w$  is the set of all words that can be obtained by permuting the letters of  $w$ . If the letters of  $w$  are distinct, then  $w$  is a permutation and  $R(w)$  is the set of elements of the symmetric group  $\mathcal{S}_n$ .

Let  $w = x_1 x_2 \cdots x_n$  be a word. The *non-decreasing rearrangement* of  $w$  is the word  $\bar{w} = y_1 y_2 \cdots y_n$ , with  $y_i \leq y_{i+1}$  and  $\bar{w} \in R(w)$ . The *biword associated to  $w$*  is  $\Gamma(w) = \binom{\bar{w}}{w}$ . Our task is now to define  $w^{-1}$ .

---

\*Much of this work was carried out while the author was visiting Chalmers University in Göteborg, Sweden, during spring 1999. The author thanks Einar Steingrímsson (Göteborg) for many valuable conversations.

## 1.2 Desirable properties of $w^{-1}$ and the Foata-Han inverse

Foata and Han [2] list the following desirable properties of the inverse of a word.

**W0** If  $w = \sigma$  is a permutation then  $w^{-1} = \sigma^{-1}$ .

**W1** The map  $w \mapsto w^{-1}$  is an involution.

**W2** The number of occurrences of any biletter  $\binom{a}{b}$  in  $\Gamma(w)$  equals the number of occurrences of  $\binom{b}{a}$  in  $\Gamma(w^{-1})$ .

**W3**  $\text{inv } w^{-1} = \text{inv } w$ , where

$$\text{inv } w = \#\{(i, j) \mid i < j, w_i > w_j\}.$$

They also define an inverse  $w^*$  for a word  $w$  that satisfies the first three of these properties, but not the fourth. We describe their inverse briefly here.

Two biwords  $\gamma$  and  $\gamma'$  are *equivalent* if one can be obtained from the other by interchanging adjacent columns, where columns (bilettters)  $\binom{a}{b}$  and  $\binom{a'}{b'}$  are interchangeable iff  $a \neq a'$ .

A *cycle* is a biword  $\begin{pmatrix} x_1 & x_2 & \dots & x_k \\ x_2 & x_3 & \dots & x_1 \end{pmatrix}$  such that all  $x_i$  are distinct. We use the notation  $[x_1 x_2 \dots x_k]$  for this cycle. The cycle is called a *dominant cycle* if  $x_1$  is the largest element.

**Theorem (Foata, [1]).** *Any biword is equivalent to a (juxtaposition) product of dominant cycles. Any two such factorizations can be obtained from one another by interchanging two consecutive cycles having no letter in common.*

To find the Foata-Han inverse  $w^*$ , write  $\Gamma(w)$  as a product of dominant cycles  $\gamma_1 \dots \gamma_l$ . Turn each cycle  $\gamma_i$  upside down to get  $\delta_i$ , rearrange  $\delta_1 \dots \delta_l$  so that the top row is non-decreasing, then the bottom row is  $w^*$ .

**Theorem (Foata & Han, [2]).** *The Foata-Han inverse satisfies properties W0–W2.*

## 1.3 Internal and external inversions

Let  $c = 1^{c_1} \dots r^{c_r}$  be a word in non-decreasing letters and let  $w \in R(c)$ . Write

$$\Gamma(w) = \binom{c}{w} = \begin{pmatrix} 1^{c_1} & 2^{c_2} & \dots & r^{c_r} \\ w_1 & w_2 & \dots & w_r \end{pmatrix}.$$

**Definition 1.** For  $w$  as above,  $\text{intinv } w$  is the total number of inversions of  $w_1, w_2, \dots, w_r$  and  $\text{extinv } w$  is the total number of inversions between letters of distinct words  $w_1, w_2, \dots, w_r$ .

Clearly  $\text{inv } w = \text{intinv } w + \text{extinv } w$ .

**Definition 2.** The *type* of  $w$  is the  $r \times r$  matrix  $A = [a_{ij}]$ , where

$$\begin{aligned} a_{ij} &= \text{number of letters } j \text{ in } w_i; \\ &= \text{number of bilettters } \binom{i}{j} \text{ in } \Gamma(w). \end{aligned}$$

For example, if  $w = 3\ 3\ 2\ 3\ 3\ 1\ 3\ 3\ 1\ 2\ 2\ 2\ 2\ 1\ 1$ ,  $w_1 = 3\ 3\ 2\ 3$ ,  $w_2 = 3\ 1\ 3\ 3\ 1$  and  $w_3 = 2\ 2\ 2\ 2\ 1\ 1$  and

$$A = \begin{pmatrix} 0 & 1 & 3 \\ 2 & 0 & 3 \\ 2 & 4 & 0 \end{pmatrix}.$$

For a type matrix  $A$ ,

$$c_i = i \text{ th row sum of } A = i \text{ th column sum of } A.$$

Let  $R(c, A)$  be the set of all words in  $R(c)$  of type  $A$ . Then property W2 is equivalent to

$$w \in R(c, A) \iff w^{-1} \in R(c, A^T).$$

**Theorem (Foata & Han, [2]).**

$$\sum_{w \in R(c, A)} q^{\text{intinv } w} = \sum_{w \in R(c, A^T)} q^{\text{intinv } w}. \quad (1)$$

For all  $w \in R(c, A)$  and  $v \in R(c, A^T)$ ,

$$\text{extinv } w = \text{extinv } v \quad (2)$$

*Remark.* This proves that the Foata-Han inverse  $w^*$  satisfies

$$\text{extinv } w = \text{extinv } w^*.$$

However, it does *not* satisfy property W3.

**Corollary.** *There is a bijection  $w \mapsto w^{-1}$  on  $R(c)$  satisfying W0–W3.*

Foata and Han leave the construction of  $w^{-1}$  as an open problem.

## 2 The case $r \leq 3$

### 2.1 Transposing classes

Consider rearrangement classes  $R(c)$  and  $R(d)$ , where  $c = 1^{c_1} \dots r^{c_r}$ ,  $d = 1^{d_1} \dots r^{d_r}$  and the  $d_i$  are a permutation of the  $c_i$ . Then one can see by generating functions that  $\text{inv}$  is equidistributed on  $R(c)$  and  $R(d)$ . Here we construct a bijection expressing this fact. We may assume that  $d$  is obtained from  $c$  by interchanging the numbers of  $i$ s and  $i+1$ s, for some  $i$ . Thus for some  $i$  with  $1 \leq i < r$ ,  $d_i = c_{i+1}$ ,  $d_{i+1} = c_i$  and  $d_j = c_j$  otherwise. We define a bijection  $\tau_i : R(c) \rightarrow R(d)$  such that  $\text{inv } \tau_i(w) = \text{inv } w$  for all  $w \in R(c)$ . Consider  $w \in R(c)$ . Let  $v$  be the subword of  $w$  consisting only of letters  $i$  and  $i+1$ . Let  $u$  be the subword of  $w$  consisting of letters other than  $i$  and  $i+1$ . Let  $v'$  be the word obtained from  $v$  by first writing its letters in reverse order, then interchanging letters  $i$  and  $i+1$ . Finally, let  $w'$  be the word obtained by combining  $u$  and  $v'$  so that the letters of  $u$  retain the same positions in  $w'$  as they had in  $w$ . Then we define  $\tau_i(w)$  to be  $w'$ .

*Example 1.* Let  $w = 4\ 2\ 1\ 2\ 3\ 1\ 2\ 4\ 3\ 4 \in R(1^2 2^3 3^2 4^3)$  and  $i = 2$ . Then  $\text{inv } w = 7 + 2 + 1 + 2 + 1 = 13$ . Now  $u = 4\ 1\ 1\ 4\ 4$  and  $v = 2\ 2\ 3\ 2\ 3$ . Reversing  $v$  gives  $3\ 2\ 3\ 2\ 2$  and interchanging 2 and 3 gives  $v' = 2\ 3\ 2\ 3\ 3$ . Thus  $w' = 4\ 2\ 1\ 3\ 2\ 1\ 3\ 4\ 3\ 4 \in R(1^2 2^2 3^3 4^3)$ . Moreover,  $\text{inv } w' = 7 + 2 + 2 + 1 + 1 = 13$ .

*Problem.* The transformations  $\tau_1, \dots, \tau_{r-1}$  form a group  $H$  that is not in general isomorphic to  $S_r$ . Identify  $H$ .

## 2.2 The construction

Consider a rearrangement class  $R(c)$ , where  $c = 1^{c_1} \dots r^{c_r}$  and  $r \leq 3$ . For each type matrix  $A$ , we construct a bijection  $\Phi_A$  from  $R(c, A)$  to  $R(c, A^T)$  that preserves intinv. We can then define  $w^{-1}$  for any  $w \in R(c)$  by  $w^{-1} = \Phi_A(w)$ .

Fix a type matrix  $A = [a_{ij}]$ .

If  $r \leq 2$ ,  $A$  will be symmetric, so we may define  $\Phi_A$  to be the identity, and  $w^{-1} = w$ . So suppose  $r = 3$ . We proceed by induction on the sum of the entries of  $A$  (i.e., on the number of letters in any word  $w \in R(c, A)$ ). Thus we assume that  $v^{-1}$  can be defined for any word  $v$  of length shorter than a word in  $R(c, A)$ .

Let  $c = 1^{c_1}2^{c_2}3^{c_3}$ . Let  $w = w_1w_2w_3 \in R(c, A)$ , so that  $w_i \in R(1^{a_{i1}}2^{a_{i2}}3^{a_{i3}})$ .

Write  $F(u)$  and  $L(u)$  for the first and last letters respectively of any word  $u$ .

**Case 1a:**  $a_{11} \neq 0$ .

Then  $w_1$  contains  $\ell = a_{11}$  letters equal to 1. Mark these letters and their positions  $i_1, \dots, i_\ell$ . Let  $w' = w'_1w'_2w'_3$  be the word obtained from  $w$  by deleting the marked 1s. Then we define  $w^{-1} = \Phi_A(w)$  to be the word obtained from  $(w')^{-1}$  by inserting 1s into the positions  $i_1, \dots, i_\ell$ .

**Case 1b:**  $a_{33} \neq 0$ .

This follows similarly.

**Case 2:**  $a_{22} \neq 0$ .

Notice that the argument for the previous two cases is not valid—removal and reinstatement of 2s in  $w_2$  will have unpredictable consequences for intinv.

Suppose that  $a_{22} \neq 0$ . Let  $u = \tau_2(w_1)\tau_2(w_3)\tau_2(w_2)$ . Then  $u \in R(c', A')$ , where  $c' = 1^{c_1}2^{c_3}3^{c_2}$  and  $A'$  is the matrix obtained by interchanging the second and third rows and columns of  $A$ . Thus  $a'_{33} \neq 0$ . As in the proof of the previous case, we may obtain  $u^{-1} = v = v_1v_2v_3$  in terms of the inverse of the word obtained by deleting the 3s from  $\tau_2(w_2)$ . Now we define

$$w^{-1} = \tau_2(v_1)\tau_2(v_3)\tau_2(v_2).$$

**Case 3a:**  $F(w_1) = 2, F(w_2) = 1$ .

We assume each  $a_{ii} = 0$ , or we are in a previous case.

Let  $w'_1$  and  $w'_2$  be the words obtained by deleting the first letters of  $w_1$  and  $w_2$  respectively, and let  $w' = w'_1w'_2w'_3$ . Let  $(w')^{-1} = v_1v_2v_3$ . We define  $w^{-1} = v'_1v'_2v'_3$ , where  $v'_1$  and  $v'_2$  are the words obtained by inserting first letters of 2 and 1 in  $v_1$  and  $v_2$  respectively.

The following 5 cases are similar.

**Case 3b:**  $F(w_1) = 3, L(w_3) = 1$ .

**Case 3c:**  $F(w_2) = 3, F(w_3) = 2$ .

**Case 3d:**  $L(w_1) = 2, L(w_2) = 1$ .

**Case 3e:**  $L(w_1) = 3, F(w_3) = 1$ .

**Case 3f:**  $L(w_2) = 3, L(w_3) = 2$ .

**Case 4a:**  $F(w_1) = L(w_1) = 2, F(w_2) = L(w_2) = 3, F(w_3) = L(w_3) = 1$ .

Let  $w'_1, w'_2$  and  $w'_3$  be the words obtained by deleting the first and last letters of  $w_1, w_2$  and  $w_3$  respectively, and let  $w' = w'_1w'_2w'_3$ . Let  $(w')^{-1} = v_1v_2v_3$ . We define  $w^{-1} = v'_1v'_2v'_3$ , where  $v'_1, v'_2$  and  $v'_3$  are the words obtained by inserting first and last letters of 3, 1 and 2 into  $v_1, v_2$  and  $v_3$  respectively.

**Case 4b:**  $F(w_1) = L(w_1) = 3$ ,  $F(w_2) = L(w_2) = 1$ ,  $F(w_3) = L(w_3) = 2$ .

This is similar to the previous case.

It is easy to see that at least one of the previous cases occurs, and that where two cases occur, the order in which they are treated is immaterial. One can now easily prove the following theorem.

**Theorem 1.** *The mapping  $w \mapsto w^{-1}$  described above is a bijection on  $R(c)$  that satisfies properties W0–W3.*

*Example 2.* Let

$$w = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 2 & 3 & 2 & 1 & 2 & 2 & 3 & 3 & 3 & 3 & 1 & 1 & 2 & 1 & 2 & 2 & 1 \end{pmatrix},$$

so that  $w = w_1 w_2 w_3$  with  $w_1 = 2\ 3\ 2\ 1\ 2$ ,  $w_2 = 2\ 3\ 3\ 3\ 3\ 1\ 1$ ,  $w_3 = 2\ 1\ 2\ 2\ 1$ .

Note that  $\text{intinv } w = \text{inv } w_1 + \text{inv } w_2 + \text{inv } w_3 = 5 + 10 + 4 = 19$ .

*Step 1* Now  $w_1$  has a 1 in position 4. Delete this to get  $w'_1 = 2\ 3\ 2\ 2$ .

*Step 2* Also,  $w_2$  contains a 2. Now  $v = \tau_2(w'_1)\tau_2(w_3)\tau_2(w_2) = u_1 u_2 u'_3$ , where  $u_1 = 3\ 3\ 2\ 3$ ,  $u_2 = 3\ 1\ 3\ 3\ 1$ ,  $u'_3 = 2\ 2\ 2\ 2\ 3\ 1\ 1$ . Delete the 3 in position 5 of  $u'_3$  to get  $u_3 = 2\ 2\ 2\ 2\ 1\ 1$ . We now have a word  $u = u_1 u_2 u_3 = 3\ 3\ 2\ 3\ 3\ 1\ 3\ 3\ 1\ 2\ 2\ 2\ 2\ 1\ 1$  containing no fixed points, whose inverse we calculate as follows.

	word 1				word 2				word 3							
3	3	2	3	3	3	1	3	3	1	2	2	2	2	1	1	
<i>Step 3</i>	3	.	.	.	.	.	.	.	.	.	.	.	.	1	<i>Case 3b</i>	
<i>Step 4</i>	3	3	.	.	.	.	.	.	.	.	.	.	.	1	1	<i>Case 3b</i>
<i>Step 5</i>	3	3	.	.	3	.	.	.	.	2	.	.	.	1	1	<i>Case 3c</i>
<i>Step 6</i>	3	3	2	.	3	1	.	.	.	2	.	.	.	1	1	<i>Case 3d</i>
<i>Step 7</i>	3	3	2	.	3	1	3	.	.	2	2	.	.	1	1	<i>Case 3c</i>
<i>Step 8</i>	3	3	2	.	3	1	3	3	.	2	2	2	.	1	1	<i>Case 3c</i>
<i>Step 9</i>	3	3	2	2	3	1	3	3	3	2	2	2	1	1	1	<i>Case 4b</i>

Hence  $u^{-1} = v = 3\ 3\ 2\ 2\ 3\ 1\ 3\ 3\ 3\ 2\ 2\ 2\ 1\ 1\ 1 = v_1 v_2 v_3$ , where  $v_1 = 3\ 3\ 2\ 2$ ,  $v_2 = 3\ 1\ 3\ 3\ 3$ ,  $v_3 = 2\ 2\ 2\ 1\ 1\ 1$ .

*Step 10* Insert 3 into position 5 in  $v_3$  to get  $v'_3 = 2\ 2\ 2\ 1\ 3\ 1\ 1$ .

*Step 11* Find  $\tau_2(v_1)\tau_2(v'_3)\tau_2(v_2) = 3\ 3\ 2\ 2\ 2\ 3\ 3\ 1\ 3\ 1\ 1\ 2\ 1\ 2\ 2\ 2$ . Finally, insert 1 into position 4 of this word to get

$$w^{-1} = 3\ 3\ 2\ 1\ 2\ 2\ 3\ 3\ 1\ 3\ 1\ 1\ 2\ 1\ 2\ 2\ 2.$$

We have  $\text{intinv } w^{-1} = 7 + 11 + 1 = 19 = \text{intinv } w$ .

### 3 The general case

The above arguments for Cases 1 and 2 can be adapted to the general case. However, it seems difficult to identify a small number of cases corresponding to cases 3 and 4 above that exhaust all possibilities. It is necessary to proceed in a different way, and we outline below a possible approach.

#### 3.1 Critical points

In this section we give a method of deleting a suitable element, an *introvert*, from a word while fixing the number of inversions.

**Definition 3.** Let  $w = x_1 x_2 \cdots x_n$  be a word. Then  $x_i$  is a *right trough* of  $w$  if  $x_i \geq x_1$  and  $x_i < x_j$  for all  $j > i$ . Also,  $x_i$  is a *right peak* of  $w$  if  $x_i \leq x_1$  and  $x_i > x_j$  for all  $j < i$ . Similarly, we define *left trough* and *left peak* of  $w$ .

The element  $x_i$  is an *introvert* if  $x_i$  is a right trough or a left peak;  $x_i$  is an *extrovert* if  $x_i$  is a right peak or a left trough;  $x_i$  is a *critical point* if  $x_i$  is an introvert or an extrovert.

Note that the first and last elements of a word are always either introverts or extroverts.

**Lemma 2.** The word  $w$  contains an introvert if and only if  $F(w) \leq L(w)$ ;  $w$  contains an extrovert if and only if  $F(w) \geq L(w)$ .

**Definition 4.** A word  $w$  is said to be *Type A* if  $F(w) < L(w)$ , *Type B* if  $F(w) > L(w)$ , *Type M* if  $F(w) = L(w)$ .

Thus  $w$  contains an introvert if and only if  $w$  is of type *A* or *M*.

Let  $w = x_1 x_2 \cdots x_n$ , and suppose that  $x = x_i$  is a right trough of  $w$ . Let  $v = z_1 z_2 \cdots z_m$  be the subword of  $w$  consisting of those letters of  $w$  that are greater than  $x$ , and let  $u = y_1 y_2 \cdots y_l$  be the subword of  $w$  consisting of those letters of  $w$  that are less than or equal to  $x = x_i$ . (Note that all the letters of  $u$  will be to the left of  $x_i$ ). Suppose that the letters in  $v$  occupy the places  $i_1, i_2, \dots, i_m$ , where  $1 < i_1 < \dots < i_m \leq n$ . Form the word  $w'$  of length  $n - 1$  as follows:

Firstly, insert the letters  $z_1, \dots, z_m$  of  $v$  into places  $i_1 - 1, i_2 - 1, \dots, i_m - 1$ . Secondly, insert the letters  $y_1, \dots, y_l$  of  $u$  from left to right into the vacant places in  $w'$ .

More succinctly, letters greater than  $x$  are moved one space to the left; the letters less than or equal to  $x$ , other than  $x = x_i$ , are reinserted in the same order into the spaces left;  $x$  itself is deleted.

**Definition 5.** The *rt-deleted* word  $w \setminus x$  is the word  $w'$  obtained above.

*Example 3.* Consider  $w = 2\ 5\ 1\ 3\ 6\ 6\ 4\ 3\ 4\ 5$  and its right trough  $x_9 = 4$ . Then  $w \setminus 4 = 5\ 2\ 1\ 6\ 6\ 3\ 4\ 3\ 5$ .

If  $x$  is a left trough of  $w$ , we define the *lt-deleted* word  $w \setminus x$  in the same way as above, interchanging “left” and “right”. Similarly, if  $x$  is a right peak or left peak of  $w$ , we define the *rp-deleted* or *lp-deleted* word  $w \setminus x$  as above, with “less than” and “greater than” interchanged.

The following theorem is not difficult to prove.

**Theorem 3.** Let  $x$  be an introvert of  $w$ . Then

$$\text{inv } w \setminus x = \text{inv } w.$$

Let  $x$  be an extrovert of the word  $w$  of length  $n$ , and suppose that  $w$  has  $m$  letters not equal to  $x$ . Then

$$\text{inv } w \setminus x = \text{inv } w - m.$$

### 3.2 Insertion

Given any word  $w = x_1 x_2 \cdots x_n$  and any letter  $x$ , we can perform the reverse of the above construction and insert  $x$  into  $w$  as either a right trough, right peak, left trough or left peak.

Suppose we wish to insert  $x$  into  $w$  as a right trough. Let  $x_i$  be the right-most element of  $w$  such that  $x_i \leq x$ . If no such element exists, set  $i = 0$ . Let  $v = z_1 z_2 \cdots z_m$

be the subword of  $w$  consisting of those letters of  $w$  that are greater than  $x$ , and let  $u = y_1 y_2 \cdots y_l$  be the subword of  $w$  consisting of those letters of  $w$  that are less than or equal to  $x = x_i$ . (Note that all the letters of  $u$  will be to the left of  $x_{i+1}$ ). Form the word  $w'$  of length  $n + 1$  as follows: letters greater than  $x$  are moved one space to the right; the letters of  $w$  less than or equal to  $x$  are reinserted in the same order into the spaces left;  $x$  itself is inserted into the remaining space (which will be place  $i + 1$ ). We call the word  $w'$  created by the above process the *rt-inserted word*  $w + x$  (or sometimes  $w +_{\text{rt}} x$ ).

Similarly, we can insert  $x$  into  $w$  in any of the other three ways.

*Example 4.* Consider  $w = 5 \ 2 \ 1 \ 6 \ 6 \ 3 \ 4 \ 3 \ 5$ . We can insert  $x = 4$  into  $w$  in any of four ways to form the words

$$\begin{array}{ll} w +_{\text{rt}} 4 = 2 \ 5 \ 1 \ 3 \ 6 \ 6 \ 4 \ 3 \ 4 \ 5; & w +_{\text{rp}} 4 = 5 \ 6 \ 2 \ 1 \ 6 \ 4 \ 3 \ 5 \ 3 \ 4; \\ w +_{\text{lp}} 4 = 4 \ 2 \ 1 \ 5 \ 6 \ 3 \ 3 \ 6 \ 3 \ 4 \ 5; & w +_{\text{lt}} 4 = 5 \ 4 \ 2 \ 6 \ 6 \ 1 \ 3 \ 4 \ 5 \ 3. \end{array}$$

### 3.3 The Graph of a Word

Suppose now that we have a word  $w = w_1 w_2 \dots w_r$ , where  $w_i \in R(1^{a_{i1}} \dots r^{a_{ir}})$ . Form a directed graph  $G = G(w)$  as follows. The vertices of  $G$  are  $1, 2, \dots, r$ . A vertex  $i$  is of *type A, B or M* according as the word  $w_i$  is of type A, B or M. If  $i$  is a vertex of type A or M then an arrow of type A is drawn from  $i$  to each vertex  $j$  such that  $j$  is an introvert of  $w_i$ . If  $i$  is a vertex of type B or M then an arrow of type B is drawn from  $i$  to each vertex  $j$  such that  $j$  is an extrovert of  $w_i$ . Since from each vertex of  $G$  there will be at least one arrow,  $G$  must contain an oriented cycle.

An *A-cycle* resp. *B-cycle* of  $G$  is an oriented cycle all of whose arrows are of type A resp. B. Thus an A-cycle corresponds to a sequence of letters  $i_1, \dots, i_m$  such that each  $w_{i_j}$  is of type A or M and  $i_{j+1}$  is an introvert of  $w_{i_j}$ . Thus  $i_{j+1}$  will be either a right trough or a left peak of  $w_{i_j}$  (or possibly both). We denote the cycle by  $[i_1 i_2 \dots i_m]$ , where  $i_1$  is the largest letter in the cycle.

### 3.4 The construction of the inverse—first step

Suppose we wish to find the inverse of the word  $w$ , and assume the previous notation. We may assume that we can find  $v^{-1}$  for all words  $v$  that are shorter than  $w$ .

Suppose that  $w$  has an A-cycle. Then we can use the procedure described in section 3.1 to delete this cycle from  $w$  while fixing the number of internal inversions. Thus if  $[i_1 i_2 \dots i_m]$  is the cycle, we know that  $i_2$  is an introvert of  $w_{i_1}$ . If  $i_2$  is a right trough, form the rt-deleted word  $w'_{i_1} = w_{i_1} \setminus i_2$ . Otherwise,  $i_2$  must be a left peak and we can form the lp-deleted word  $w'_{i_1} = w_{i_1} \setminus i_2$ . Continue in this way with  $i_3, \dots, i_m, i_1$ . By replacing each  $w_i$  that occurs in the cycle by  $w'_i$  we will have a shorter word  $w'$  with the same number of inversions as  $w$ . Now find  $v = (w')^{-1} = v_1 \dots v_r$ . We now insert the inverse cycle  $[i_m \dots i_2 i_1]$  into  $v'$  as an A-cycle. Thus, if  $i_2$  was a right trough of  $w_{i_1}$ , insert  $i_1$  into  $v_{i_2}$  as a right trough using the procedure of section 3.2, to form the rt-inserted word  $v'_{i_2} = v_{i_2} +_{\text{rt}} i_1$ . Continue in this way with  $i_2, \dots, i_m$ . By replacing each  $v_i$  that occurs in the cycle by  $v'_i$ , we obtain the word  $v' = w^{-1}$ .

Suppose that  $w$  has a B-cycle. Then we use the analogous procedure. However in this case the number of inversions of  $w$  does not remain fixed, and we must check that the number of inversions lost when the cycle  $[i_1 i_2 \dots i_m]$  is deleted from  $w$  to form  $w'$  equals the number of inversions regained when the inverse cycle  $[i_m \dots i_2 i_1]$  is inserted into  $(w')^{-1}$  to form  $w^{-1}$ . This follows quickly from Theorem 3 (or from the observation

that, if every subword  $w_i$  is reversed, those vertices of  $G(w)$  of type B become type A, and the B-cycle becomes an A-cycle).

### 3.5 Unresolved issues

A word  $w$  may have more than one A or B-cycle. In this case, one can decide which cycle is to be used by ordering the cycles lexicographically. One needs to ensure that, whatever method is used to choose an A or B-cycle  $\alpha$  from  $w$ , the method will choose the inverse cycle  $\alpha^{-1}$  from  $w^{-1}$ .

The case in which  $w$  has neither an A nor a B-cycle, is as yet unresolved, but will presumably involve a more general deletion/insertion process.

*Example 5.* Let

$$w = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 \\ 2 & 4 & 3 & 3 & 2 & 3 & 3 & 3 & 1 & 3 & 3 & 1 & 4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 2 \end{pmatrix},$$

so that  $w = w_1 w_2 w_3 w_4$  with  $w_1 = 2\ 4\ 3\ 3\ 2\ 3$ ,  $w_2 = 3\ 3\ 1\ 3\ 3\ 1\ 4$ ,  $w_3 = 2\ 2\ 2\ 2\ 1\ 1\ 1$ ,  $w_4 = 1\ 2$ .

Since 2 is a right trough of  $w_1$ , 4 is a right trough of  $w_2$  and 1 is a right trough of  $w_4$ , we have an A-cycle [412] which we may delete to obtain  $w' = w'_1 w'_2 w'_3 w'_4$ , where  $w'_1 = 4\ 3\ 3\ 2\ 3$ ,  $w'_2 = 3\ 3\ 1\ 3\ 3\ 1$ ,  $w'_4 = 2$ .

Now 4 is a right peak of  $w'_1$ , 1 is a right peak of  $w_3$ , 3 is a right peak of  $w'_2$  and 2 is a right peak of  $w'_4$ , so we have a B-cycle [4231] which we may delete to obtain  $w'' = w''_1 w''_2 w''_3 w''_4$ , where  $w''_1 = 3\ 3\ 2\ 3$ ,  $w''_2 = 3\ 1\ 3\ 3\ 1$ ,  $w''_3 = 2\ 2\ 2\ 2\ 1\ 1$ ,  $w''_4 = \emptyset$ .

Now  $u = w''$  is the word on 3 letters considered in example 2, whose inverse we obtained as  $v = v_1 v_2 v_3$ , where  $v_1 = 3\ 3\ 2\ 2$ ,  $v_2 = 3\ 1\ 3\ 3\ 3$ ,  $v_3 = 2\ 2\ 2\ 1\ 1\ 1$ .

Then rp-inserting the cycle [4132] into  $v$  gives  $v' = v'_1 v'_2 v'_3 v_4$ , where  $v'_1 = 3\ 3\ 3\ 2\ 2$ ,  $v'_2 = 4\ 3\ 1\ 3\ 3\ 3$ ,  $v'_3 = 2\ 2\ 2\ 2\ 1\ 1\ 1$ ,  $v_4 = 1$ .

Finally, rt-inserting the cycle [421] into  $v'$  gives  $w^{-1} = v''_1 v''_2 v'_3 v'_4$ , where  $v''_1 = 3\ 3\ 3\ 2\ 2\ 4$ ,  $v''_2 = 1\ 4\ 3\ 1\ 3\ 3\ 3$ ,  $v'_3 = 2\ 2\ 2\ 2\ 1\ 1\ 1$ ,  $v'_4 = 1\ 2$ . Thus

$$w^{-1} = 3\ 3\ 3\ 2\ 2\ 4\ 1\ 4\ 3\ 1\ 3\ 3\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2.$$

## References

- [1] D. Foata, *Rearrangements of Words*, in M. Lothaire, *Combinatorics on Words*, CUP, Cambridge, 1997
- [2] D. Foata and G-N Han, *Inverses of Words*, Séminaire Lotharingien de Combinatoire, B39d, 1997

# Bisuites de parking

*ou Configurations récurrentes dans les graphes bipartis complets*

Robert Cori<sup>\*</sup>  
Dominique Poulalhon<sup>†</sup>

## Résumé

Les suites de parking se sont révélées être au centre de différents problèmes combinatoires. Nous introduisons ici des couples de suites d'entiers dont la définition est voisine de celle des suites de parking, que nous nous proposons d'appeler *bisuites de parking*. Nous en donnons une interprétation imagée, tirée de la constitution d'une liste de candidats pour un scrutin. Les suites de parking peuvent d'autre part être considérées comme les configurations récurrentes dans un automate du tas de sable sur le graphe complet; nous montrons que les bisuites de parking correspondent aux configurations récurrentes sur le graphe biparti complet auquel on a ajouté un puits relié à tous les sommets. Ceci nous permet d'en donner une formule d'énumération.

## Abstract

Parking functions are central in many aspects of combinatorics. We define in this communication a generalization of parking functions which we call *parking bifunctions*. These can be introduced in a very simple way as ordered lists of candidates for elections which satisfy some constraints. We give a characterization of these bifunctions in terms of parking functions and we show that they can be interpreted as recurrent configurations in the sandpile model for some graphs. These are complete bipartite graphs to which is added a sink adjacent to all the vertices. As a consequence we obtain an enumeration formula.

## 1 Introduction

Les suites de parking, introduites il y a plus de trente ans [9, 10] pour l'analyse d'algorithmes de hachage, présentent un intérêt majeur en combinatoire des objets étiquetés : comme le montre par exemple la jolie preuve de Pollack (voir [6]), elles sont dénombrées par la famille  $(n^{n-2})_{n \in \mathbb{N}}$ , qui joue vis-à-vis des objets étiquetés un rôle comparable à celui des nombres de Catalan en ce qui concerne les objets non étiquetés ; elles peuvent donc être mises en bijection avec de nombreuses classes d'objets comme les arbres de Cayley, les factorisations minimales d'un grand cycle de  $S_n$  en produit de transpositions, les chaînes maximales du treillis des partitions non croisées ou encore les cellules de l'arrangement d'hyperplans *sandwich* (voir par exemple [4, 7, 13, 14]).

Parmi les multiples manières de définir ces suites, nous adopterons ici la suivante : une *suite de parking* de longueur  $n$  est une suite d'entiers  $u = u_1 \ u_2 \ \dots \ u_n$  compris entre

---

<sup>\*</sup>Labri, Université Bordeaux 1 et LIX, École Polytechnique, France. cori@labri.u-bordeaux.fr

<sup>†</sup>LIX, École Polytechnique, France. poulalho@lix.polytechnique.fr

0 et  $n - 1$ , telle qu'il existe une permutation  $\alpha = a_1 \ a_2 \ \dots \ a_n$  dans  $S_n$  satisfaisant, pour tout  $i$ ,  $u_i < a_i$ . On dira que la permutation  $\alpha$  est un *certificat* pour la suite  $u$ . Par exemple, 3 0 1 3 1 est une suite de parking, dont la permutation 4 1 2 5 3 est un certificat; en revanche, 0 3 2 3 2 n'en est pas une.

Cette terminologie s'explique par le fait que les suites de parking sont exactement les mots pour lesquels l'algorithme décrit ci après, dit *de parking*, réussit. Il s'agit d'une version imagée de l'algorithme de hachage par adressage ouvert : considérons une rue à sens unique dans laquelle se trouve une rangée de  $n$  places de parking, numérotées de 0 à  $n - 1$  dans l'ordre;  $n$  conducteurs arrivent l'un après l'autre, le  $i$ -ème conducteur ayant l'intention de se garer en place  $u_i$ . Si celle-ci est encore libre, il s'y gare, et sinon, il continue sa route et se gare dans la première place libre située au-delà. Si l'un des conducteurs ne parvient pas à se garer, il quitte la rue et l'algorithme échoue.

Notre propos consistera ici à définir une classe d'objets pouvant être considérée comme une généralisation de l'ensemble des suites de parking, et susceptible de posséder également des propriétés combinatoires intéressantes.

## 2 Définition

Dans la suite, pour tout couple d'entiers  $(a, b)$  avec  $a \leq b$ ,  $\llbracket a, b \rrbracket$  désigne l'ensemble des entiers compris (au sens large) entre  $a$  et  $b$ .

Soit  $p$  et  $q$  deux entiers, et  $n = p + q$ ; à toute permutation  $\alpha = a_1 \ a_2 \ \dots \ a_n$  dans  $S_n$ , on associe le couple formé par les deux suites  $L_\alpha = x_1 \ x_2 \ \dots \ x_p$  et  $R_\alpha = y_1 \ y_2 \ \dots \ y_q$  définies par :

$$\forall i \leq p, \ x_i = |\{1 \leq j \leq q \mid a_{p+j} < a_i\}|$$

et

$$\forall j \leq q, \ y_j = |\{1 \leq i \leq p \mid a_i < a_{p+j}\}|.$$

Ainsi par exemple, pour  $p = 5$  et  $q = 4$ , à la permutation 3 6 5 2 8 4 1 9 7 est associé le couple (1 2 2 1 3 , 2 0 5 4). On note que le couple  $(L_\alpha, R_\alpha)$  ne détermine pas  $\alpha$  de manière unique puisque par exemple les deux suites précédentes sont aussi associées à la permutation 2 5 6 3 8 4 1 9 7.

**Définition** *On appelle bisuite de parking de taille  $(p, q)$  un couple  $(u, v)$  de suites d'entiers positifs tel qu'il existe une permutation  $\alpha$  satisfaisant l'inégalité  $(u, v) \leq (L_\alpha, R_\alpha)$ , ie :*

$$\forall i \in \llbracket 1, p \rrbracket, \ u_i \leq x_i \quad \text{et} \quad \forall j \in \llbracket 1, q \rrbracket, \ v_j \leq y_j.$$

*On dira que la permutation  $\alpha$  est un certificat pour la bisuite  $(u, v)$ .*

On peut donner le prétexte de la constitution d'une liste électorale pour introduire ces suites; il s'agit dans ce contexte de constituer une liste ordonnée de candidats à des élections, cette liste devant comporter  $p$  femmes  $F_1, F_2, \dots, F_p$  et  $q$  hommes  $H_1, H_2, \dots, H_q$ , et satisfaire leurs exigences respectives : chaque femme  $F_i$  indique le nombre maximum  $u_i$  d'hommes qu'elle accepte de voir figurer avant elle, et symétriquement, chaque homme  $H_j$  indique le nombre maximum  $v_j$  de femmes qu'il accepte de voir figurer avant lui.

On vérifie aisément que ces demandes sont réalisables si et seulement si les suites  $u'$  et  $v'$  données par  $u'_i = q - u_i$  et  $v'_j = p - v_j$  forment une bisuite de parking. En effet, soit  $\alpha = a_1 \ a_2 \ \dots \ a_n$  un certificat pour  $(u', v')$ ; on constitue une liste telle que, pour  $i \in \llbracket 1, p \rrbracket$ ,  $n + 1 - a_i$  est la position de la femme  $F_i$  et pour  $j \in \llbracket 1, q \rrbracket$ ,  $n + 1 - a_{p+j}$  est celle de l'homme  $H_j$ . Réciproquement on vérifie qu'une liste satisfaisant les exigences

respectives des hommes et des femmes permet d'obtenir une permutation  $\alpha$  qui est un certificat pour  $(u', v')$ .

### 3 Lien avec les suites de parking

On se propose tout d'abord de donner un critère simple pour vérifier si un couple de suites d'entiers positifs  $(u, v)$  est une bisuite de parking. Pour toute suite d'entiers  $u = u_1 \ u_2 \ \dots \ u_p$ , soit  $\rho_u$  la fonction *rang* définie par l'égalité, pour tout  $i \in \llbracket 1, p \rrbracket$ :

$$\rho_u(i) = |\{1 \leq j \leq p \mid u_j < u_i\}| + |\{1 \leq j < i \mid u_j = u_i\}|.$$

$\rho_u$  est donc une bijection de  $\llbracket 1, p \rrbracket$  dans  $\llbracket 0, p - 1 \rrbracket$  telle que les nombres  $u_i + \rho_u(i)$  sont tous distincts et satisfont  $u_i + \rho_u(i) < u_j + \rho_u(j)$  dès que  $u_i < u_j$ . Notons  $\vec{u}$  la suite dont le  $i$ -ème terme est  $u_i + \rho_u(i)$ . Alors :

**Proposition 1** *Un couple  $(u, v)$  de suites d'entiers forme une bisuite de parking si et seulement si la suite  $w$  obtenue en concaténant  $\vec{u}$  et  $\vec{v}$  est une suite de parking.*

*Démonstration*

Soit  $p$  et  $q$  les longueurs respectives de  $u$  et  $v$  et soit  $\alpha$  une permutation dans  $S_{p+q}$  telle que pour tous  $i, j \in \llbracket 1, p+q \rrbracket$ , si  $w_i < w_j$ , alors  $a_i < a_j$ . Montrons que  $\alpha$  est un certificat pour  $w$  si et seulement si c'est un certificat pour  $(u, v)$ .

Notons  $L_\alpha = x_1 \ \dots \ x_p$ . Alors, pour tout  $i \in \llbracket 1, p \rrbracket$ ,  $x_i = |\{1 \leq j \leq q \mid a_{p+j} < a_i\}|$ . D'autre part,  $\rho_u(i) = |\{1 \leq k \leq p \mid a_k < a_i\}|$ . Comme  $\alpha$  est une permutation, ceci entraîne que  $x_i + \rho_u(i) = a_i - 1$ , donc

$$\forall i \in \llbracket 1, p \rrbracket, \ x_i - u_i = a_i - 1 - w_i.$$

De même, si  $R_\alpha = y_1 \ \dots \ y_q$ , alors pour tout  $j \in \llbracket 1, q \rrbracket$ ,  $y_j - v_j = a_{p+j} - 1 - w_{p+j}$ . Donc  $w < \alpha$  si et seulement si  $(u, v) \leqslant (L_\alpha, R_\alpha)$ . Pour achever la preuve, il suffit de remarquer que toute (bi)suite de parking possède un certificat satisfaisant la contrainte supplémentaire de monotonie qui a été imposée à  $\alpha$ . Nous dirons d'un tel certificat qu'il *respecte l'ordre*.

□

Un corollaire de cette caractérisation des bisuites de parking indique que l'ensemble des suites de parking est en quelque sorte la diagonale de celui des bisuites :

**Proposition 2** *Une suite  $u = u_1 \ u_2 \ \dots \ u_n$  est de parking si et seulement si le couple  $(u, u)$  est une bisuite de parking.*

*Démonstration*

Il est clair que, si  $u$  est une suite de parking, alors  $\rho_u + 1$  en est un certificat. Donc  $u$  est de parking si et seulement si, pour tout  $i \in \llbracket 1, n \rrbracket$ ,  $u_i \leq \rho_u(i)$ , c'est-à-dire si et seulement si, pour tout  $i \in \llbracket 1, n \rrbracket$ ,  $\vec{u}_i \leq 2\rho_u(i)$ .

Considérons la suite  $w$ , carré de  $\vec{u}$  pour la concaténation. Remarquons que  $\vec{u}$  est une suite d'éléments distincts, dont la fonction rang est  $\rho_u$ . La fonction rang de  $w$  est donc donnée par :

$$\forall i \in \llbracket 1, n \rrbracket, \ \rho_w(i) = 2\rho_u(i) \text{ et } \rho_w(n+i) = 2\rho_u(i) + 1.$$

D'après ce qui précède,  $w$  est donc de parking si et seulement si  $w_{n+i} = w_i \leq 2\rho_u(i)$  pour tout  $i \in \llbracket 1, n \rrbracket$ , ce qui donne le résultat.

□

## 4 Énumération

À la suite des travaux de D. Dhar ([3, 12], voir aussi [1]), on définit le modèle du tas de sable sur un graphe de la manière suivante. Soit un graphe  $G$  connexe ayant pour sommets  $x_1, x_2, \dots, x_n$ , et dont un des sommets, appelé le *puits*, est distingué. Dans ce qui suit, le puits sera le sommet  $x_n$ , et on suppose que la suppression du puits ne disconnecte pas le graphe. Une *configuration* est une suite d'entiers positifs ou nuls  $u_1, u_2, \dots, u_{n-1}$  attribués aux sommets de  $G$  distincts du puits; dans le contexte du modèle physique,  $u_i$  représente le nombre de grains de sable situés en  $x_i$ , et d'autre part on ne tient pas compte du nombre de grains situés dans le puits. L'ensemble des configurations peut être muni d'une loi d'addition  $((u + v)_i = u_i + v_i$  pour tout  $i \in \llbracket 1, n-1 \rrbracket)$  qui lui confère une structure de monoïde isomorphe à  $\mathbb{N}^{n-1}$ .

Une configuration est dite *stable* si  $u_i < d_i$  pour  $1 \leq i \leq n-1$ , où  $d_i$  dénote le degré du sommet  $x_i$ . Dans une configuration instable, un des sommets  $x_i$  vérifie  $u_i \geq d_i$ , et peut alors *s'ébouler* pour donner la configuration  $v$  donnée par :

$$\begin{cases} v_i &= u_i - d_i \\ v_j &= u_j + 1 \text{ si } x_j \text{ est voisin de } x_i \\ v_j &= u_j \text{ sinon.} \end{cases}$$

On notera  $\rightarrow$  la relation d'éboulement, et  $\xrightarrow{*}$  sa fermeture transitive. On démontre facilement que, pour toute configuration  $u$ , il existe une configuration stable unique  $\hat{u}$  telle que  $u \xrightarrow{*} \hat{u}$ .

Parmi les configurations stables, certaines jouent un rôle prépondérant dans l'étude des tas de sable : les configurations *récurrentes*. Ce sont les configurations stables  $u$  telles qu'il existe une configuration  $v$  non nulle satisfaisant  $u + v \xrightarrow{*} u$ . Rappelons quelques résultats dûs à D. Dhar :

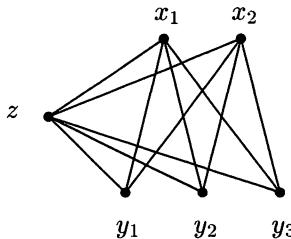
- l'ensemble des configurations récurrentes peut être muni d'une structure de groupe grâce à l'opération  $(u, v) \mapsto \widehat{u + v}$ ,
- il est en bijection avec l'ensemble des arbres couvrants de  $G$ ,
- si une configuration  $u$  est récurrente, alors  $u + \beta \xrightarrow{*} u$ , où  $\beta$  est donnée par :

$$\beta_i = \begin{cases} 1 & \text{si } x_i \text{ est voisin du puits} \\ 0 & \text{sinon.} \end{cases}$$

Les suites de parking s'interprètent naturellement comme les configurations récurrentes de l'automate du tas de sable sur le graphe complet (voir [2]); elles sont d'autre part en bijection avec les arbres de Cayley. Nous introduisons ici un graphe dont les configurations récurrentes sont les bisuites de parking, ce qui permet d'obtenir une bijection entre ces bisuites et les arbres couvrants du graphe; cette bijection peut aussi être décrite directement comme ce sera fait plus loin. Ceci nous permettra d'énumérer les bisuites de parking via le décompte de ces arbres.

Pour deux entiers  $p$  et  $q$ , considérons le graphe  $B_{p,q}$  à  $p+q+1$  sommets répartis en trois sous-ensembles,  $X_p = \{x_1, x_2, \dots, x_p\}$ ,  $Y_q = \{y_1, y_2, \dots, y_q\}$  et  $\{z\}$ , et dont les arêtes sont les éléments de l'ensemble  $(X_p \times Y_q) \cup (X_p \times \{z\}) \cup (Y_q \times \{z\})$ . On définit alors l'automate du tas de sable en choisissant pour puits le sommet  $z$ , qui est relié à tous les sommets du graphe biparti complet  $K_{p,q}$ .

EXEMPLE:  $B_{2,3} =$



Alors :

**Proposition 3** Les bisuites de parking de taille  $(p, q)$  sont en bijection avec les configurations récurrentes du tas de sable sur le graphe  $B_{p,q}$ .

Démonstration

Soit  $(u_1 \ u_2 \ \dots \ u_p, \ v_1 \ v_2 \ \dots \ v_q)$  une bisuite de parking qui a pour certificat la permutation  $\alpha = a_1 \ a_2 \ \dots \ a_n$ . On considère la configuration dans laquelle chaque sommet  $x_i$  (respectivement  $y_j$ ) possède  $n - u_i$  grains de sable (respectivement  $n - v_j$ ). On peut vérifier qu'elle est récurrente en appliquant l'algorithme de Dhar, qui consiste à ébouler le puits et à vérifier que cela entraîne un éboulement de tous les sommets : l'éboulement du puits ajoute 1 à chacune des valeurs attribuées aux sommets, et la permutation  $\alpha$  fournit ensuite un ordre d'éboulement des sommets cohérent du fait qu'elle constitue un certificat pour  $(u, v)$ .

□

On peut déduire la proposition suivante de ce qui précède, mais nous préférerons en donner une démonstration directe :

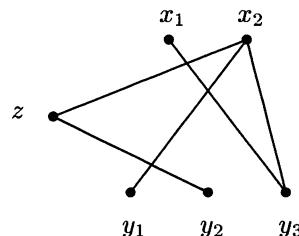
**Proposition 4** Les bisuites de parking de taille  $(p, q)$  sont en bijection avec les arbres couvrants du graphe  $B_{p,q}$ .

Démonstration

Soit  $(u, v)$  une bisuite de taille  $(p, q)$ , ie un couple tel que  $u$  est un mot de longueur  $p$  sur l'alphabet  $\llbracket 0, q \rrbracket$ , et réciproquement pour  $v$ . On construit un graphe  $T$  par l'algorithme suivant :

- on définit les indices  $i_1, i_2, \dots, i_p$  pour que  $u_{i_1} \leq u_{i_2} \leq \dots \leq u_{i_p}$ , et si  $u_{i_j} = u_{i_{j+1}}$  alors  $i_j < i_{j+1}$ : en d'autres termes, pour tout  $k \in \llbracket 1, p \rrbracket$ ,  $i_k = \rho_u^{-1}(k-1)$ . Les indices  $j_1, j_2, \dots, j_q$  sont définis de manière analogue.
- pour tout indice  $i$  dans  $\llbracket 1, p \rrbracket$ ,  $T$  contient un arc d'origine  $x_i$ , et d'extrémité  $z$  si  $u_i = 0, y_{j_{u_i}}$  sinon.
- pour tout indice  $j$  dans  $\llbracket 1, q \rrbracket$ ,  $T$  contient un arc d'origine  $y_j$ , et d'extrémité  $z$  si  $u_i = 0, x_{i_{v_j}}$  sinon.

EXEMPLE:  $(3 \ 0, \ 1 \ 0 \ 1) \longmapsto$



On obtient ainsi un graphe partiel de  $B_{p,q}$  (orienté) à  $p+q$  arêtes, dont les composantes connexes ont toutes autant d'arêtes que de sommets, sauf celle qui contient  $z$ , qui présente un déficit d'une arête, et qui est donc un arbre.

Montrons que  $T$  est un arbre couvrant de  $B_{p,q}$  (les arcs étant orientés d'un sommet vers son père dans l'arbre) si et seulement si  $(u,v)$  est une bisuite de parking. En effet, si  $(u,v)$  est une bisuite de parking, soit  $\alpha = a_1 a_2 \dots a_n$  un certificat qui respecte l'ordre. On constate alors que si  $(x_i, y_j)$  est un arc, alors  $a_i > a_{p+j}$ , et de même si  $(y_k, x_\ell)$  est un arc alors  $a_{p+k} > a_\ell$ . Ainsi il n'y a pas de circuit dans  $T$ , qui est donc un arbre.

Réciproquement, si  $T$  est un arbre, soit  $z$  sa racine; chaque arête de  $T$  est alors naturellement orientée d'un sommet vers son père. En numérotant d'une part les sommets de  $X_p$ , et d'autre part ceux de  $Y_q$ , dans l'ordre d'un parcours en largeur, on obtient un étiquetage qui correspond aux indices  $i_k$  et  $j_k$ , et permet donc de retrouver la bisuite (unique) dont  $T$  est l'image. L'arbre fournissant de plus un certificat pour cette bisuite, elle est de parking.

□

**Proposition 5** *Le nombre d'arbres couvrants du graphe  $B_{p,q}$  est donné par*

$$(p+q+1) (p+1)^{q-1} (q+1)^{p-1}.$$

#### Démonstration

On s'inspire pour cela d'une construction d'A. Joyal [8] et G. Labelle [11]. On note  $S_{p,q} = X_p \cup Y_q \cup \{z\}$  l'ensemble des sommets de  $B_{p,q}$ . On considère d'une part l'ensemble  $\mathcal{T}$  des arbres couvrants de  $B_{p,q}$  dont deux sommets sont distingués, l'un dans l'ensemble  $X_p \cup \{z\}$ , l'autre dans  $Y_q \cup \{z\}$ , et d'autre part, l'ensemble  $\mathcal{E}$  des applications de l'ensemble  $S_{p,q}$  dans lui-même telles que l'image de  $X_p$  soit incluse dans  $Y_q \cup \{z\}$ , et celle de  $Y_q$  dans  $X_p \cup \{z\}$ . L'ensemble  $\mathcal{E}$  a pour cardinal  $(p+q+1) (p+1)^q (q+1)^p$ , puisque l'image de  $z$  peut être choisie parmi  $p+q+1$  éléments, et celle de chaque élément de  $X_p$  (respectivement  $Y_q$ ) parmi  $q+1$  (respectivement  $p+1$ ).

On munit l'ensemble  $S_{p,q}$  de l'ordre suivant :

$$x_1 < x_2 < \dots < x_p < z < y_1 < y_2 < \dots < y_q.$$

Soit  $\varphi$  une endofonction de  $S_{p,q}$  qui appartient à  $\mathcal{E}$ ; elle définit un sous-graphe orienté  $G_\varphi$  de  $B_{p,q}$  constitué de composantes connexes contenant un seul circuit et des arbres enracinés en chaque sommet du circuit. On considère les sommets minimaux (au sens de l'ordre donné plus haut) de chacun de ces circuits; ce sont des éléments de  $X_p \cup \{z\}$ . Notons  $x_{i_1}, x_{i_2}, \dots, x_{i_k}$  ces sommets classés en ordre décroissant, on obtient alors de la façon suivante un arbre couvrant bipointé de  $B_{p,q}$ :

- supprimer dans  $G_\varphi$  les arcs  $(\varphi^{-1}(x_{i_j}), x_{i_j})$  pour  $j \in \llbracket 1, k \rrbracket$ .
- ajouter les arcs  $(\varphi^{-1}(x_{i_j}), x_{i_{j+1}})$  pour  $j \in \llbracket 1, k-1 \rrbracket$ .
- distinguer les sommets  $x_{i_1}$  et  $\varphi^{-1}(x_{i_k})$ .

On vérifie que cette opération réalise une bijection entre  $\mathcal{E}$  et  $\mathcal{T}$  par des arguments analogues à ceux développés pour démontrer que la transformation de Foata [5] met en bijection l'ensemble des permutations à  $k$  cycles et celui des permutations à  $k$  minima locaux.

□

## 5 Bisuites de parking croissantes

Une bisuite  $(u, v)$  de taille  $(p, q)$  sera dite *croissante* si  $u_i \leq u_{i+1}$  pour tout  $i$  dans  $\llbracket 1, p-1 \rrbracket$ , et  $v_j \leq v_{j+1}$  pour tout  $j$  dans  $\llbracket 1, q-1 \rrbracket$ . On montre :

**Proposition 6** *Le nombre de bisuites de parking croissantes de taille  $(p, q)$  est donné par :*

$$\frac{1}{n+1} \binom{n+1}{p} \binom{n+1}{q}$$

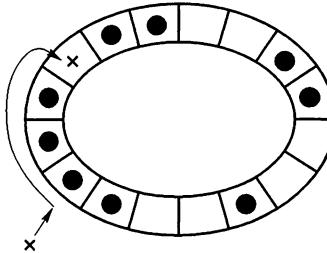
où  $n = p + q$ .

### Démonstration

Nous donnons ici une extension de la preuve de Pollack pour le dénombrement des suites de parking (voir par exemple [6]).

Dans le cas d'une suite croissante  $u$ , la fonction  $\rho_u$  est donnée par  $\rho_u(i) = i - 1$ . L'application  $u \mapsto \vec{u}$  réalise donc une bijection entre l'ensemble des suites croissantes de longueur  $p$  à valeurs dans  $\llbracket 0, q \rrbracket$  et celui des suites strictement croissantes de longueur  $p$  à valeurs dans  $\llbracket 0, p+q-1 \rrbracket$ , ie des sous-ensembles de  $\llbracket 0, p+q-1 \rrbracket$  de cardinal  $p$ . D'autre part, d'après la proposition 1, dire qu'une bisuite croissante  $(u, v)$  est de parking se traduit par le fait que la concaténation de  $\vec{u}$  et de  $\vec{v}$  est une suite de parking.

Considérons un parking circulaire à  $p+q+1$  places, numérotées de 0 à  $p+q$ , et dans lequel la règle de parking impose de se garer dans la première place libre située au-delà de la place initialement choisie en ordre cyclique; dans ce modèle, il n'y a donc plus rejet de voitures tant qu'il reste des places libres.



Parmi les couples de sous-ensembles de  $\llbracket 0, p+q \rrbracket$  de taille respective  $p$  et  $q$ , ceux qui correspondent aux bisuites de parking croissantes sont ceux qui laissent la place  $p+q$  libre. Le nombre de couples de tels sous-ensembles laissant une place donnée libre ne dépendant pas de la place choisie parmi les  $p+q+1$  places du parking, on obtient le résultat annoncé. □

Remarquons qu'une autre extension des suites de parking est proposée dans [7], sous le nom de *(k)-valet fonctions*, dont le cas particulier  $k = 2$  est équivalent à celui des bisuites de parking croissantes.

Il est intéressant de noter que les nombres  $\frac{1}{n+1} \binom{n+1}{p} \binom{n+1}{q}$ , appelés  *nombres de Narayana*, ont de nombreuses interprétations, parmi lesquelles les arbres de Catalan à  $n+2$  sommets dont  $p+1$  de hauteur paire, les arbres de Catalan à  $n+2$  sommets dont  $p+1$  feuilles, ou encore les partitions non-croisées de  $n+1$  en  $p+1$  blocs, et qu'une bijection avec le premier de ces ensembles découle de la restriction de la bijection du paragraphe précédent aux bisuites de parking croissantes.

*Remerciements.* Tous nos remerciements à Gilles Schaeffer pour sa contribution par de nombreuses discussions.

## Références

- [1] N. Biggs, *The Tutte polynomial as a growth function*, J. Alg. Comb. 10, 115–133, 1999.
- [2] R. Cori et D. Rossin, *On the sandpile group of dual graphs*, Eur. J. Comb. 21, 447–459, 2000.
- [3] D. Dhar, *Self-organized critical state of sandpile automaton models*, Phys. Rev. Lett. 64, 1613–1616, 1990.
- [4] P.H. Edelman et R. Simion, *Chains in the lattice of noncrossing partitions*, Disc. Math. 126, 107–119, 1994.
- [5] D. Foata, *Rearrangements of Words* in Lothaire, Combinatorics on Words, Addison Wesley, Reading (MA), 1983.
- [6] D. Foata, J. Riordan, *Mappings of acyclic and parking functions*, Aequ. Math. 10, 10–22, 1974.
- [7] J.D. Gilbey et L. H. Kalikow, *Parking functions, valet functions and priority queues*, Discr. Math. 197/198, 351–373, 1999.
- [8] A. Joyal, *Une théorie combinatoire des séries formelles*, Adv. in Math. 42, 1–82, 1981.
- [9] A.G. Konheim et B. Weiss, *An occupancy discipline and applications*, Siam J. Appl. Math. 14, 1266–1274, 1966.
- [10] G. Kreweras, *Sur les partitions non croisées d'un cycle* Disc. Math. 1, 333–350, 1972.
- [11] G. Labelle, *Une nouvelle démonstration combinatoire des formules de Lagrange*, Adv. in Math. 42, 217–247, 1981.
- [12] S. M. Majundar et D. Dhar, *Equivalence of the abelian sandpile model and the  $q \rightarrow 0$  limit of the Potts model*, Physica A. 185, 129–135, 1992.
- [13] R.P. Stanley, *Parking functions and noncrossing partitions*, Elec. J. Comb. 4, R20, 1997.
- [14] R.P. Stanley, *Hyperplane arrangements, parking functions and tree inversions*, Mathematical essays in honor of Gian-Carlo Rota, Progr. Math., 161, Birkhäuser, Boston/Basel/Berlin, 1998, 359–375.

# Énumérations de piles de sable

Sylvie Corteel, Dominique Gouyou-Beauchamps \*  
 LRI - Bâtiment 490 - Université Paris Sud  
 91405 Orsay Cedex - France  
 dgb@lri.fr, corteel@lri.fr

29 Mai 2000

## Abstract

We study some particular families of integer partitions related to discrete dynamical systems called “sand piles”. Our aim is to understand the structure of these combinatorial objects by decomposing and enumerating them. We first study the Ice Pile model  $IPM(k)$ . We give the area, width and height generating functions and explicit asymptotic bounds on the number of sand piles in  $IPM(k)$  with area  $n$ . We then study the  $L(\theta)$  model. Finally we propose a more general model called Frobenius sand piles.

## Résumé

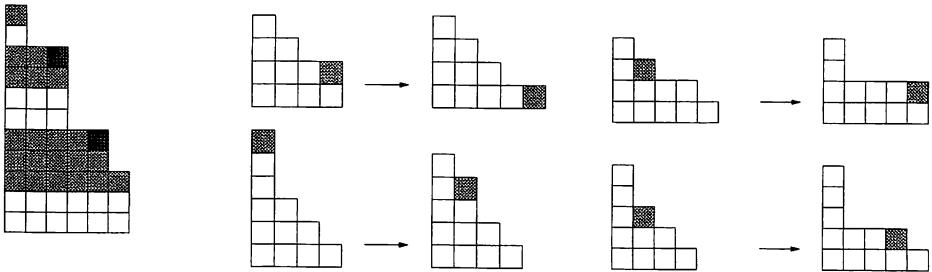
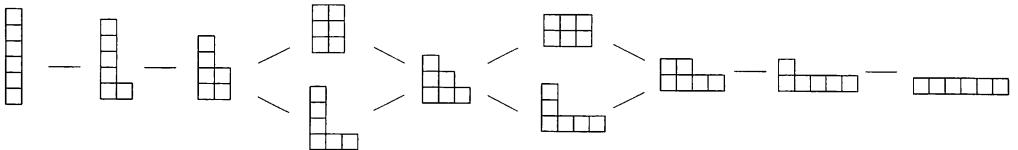
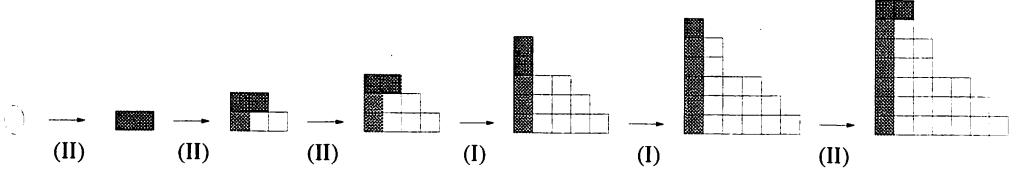
Nous étudions des familles spéciales de partages liés aux systèmes dynamiques discrets que sont les piles de sable. Nous essayons de mieux comprendre la structure de ces piles de sable en les décomposant et les comptant. Pour le modèle  $IPM(k)$ , nous trouvons les séries génératrices selon l'aire, la hauteur et la largeur de ces piles de sable ; puis nous établissons des bornes pour le nombre de piles de sable d'aire  $n$  dans  $IPM(k)$  pour  $n$  grand. Nous exposons ensuite la série selon l'aire et la hauteur pour le modèle  $L(\theta)$ . Enfin nous proposons un modèle plus général que nous appelons piles de sable de Frobenius.

## 1 Introduction

Les piles de sable sont des objets mathématiques pour Bryslawski [5], des systèmes physiques dits “auto-organisants” pour Bak et al [4] et des objets combinatoires pour Anderson et al [1], Spencer [14], Goles and Kiwi [7], Morvan et al [11, 9, 8, 12, 13] ... Une pile de sable faite de  $n$  grains est un partage de l'entier  $n$ , c'est à dire une suite d'entiers  $(\pi_1, \dots, \pi_k)$  telle que  $\pi_1 \geq \pi_2 \geq \dots \geq \pi_k \geq 1$  et  $\sum_{i=1}^k \pi_i = n$ . Nous associons à tout partage une représentation graphique appelée diagramme de Ferrers. Nous le dessinons “à la française” : la  $i^{\text{ème}}$  ligne du diagramme de Ferrers contient  $\pi_i$  carrés ou grains, les lignes sont numérotées de bas en haut et justifiées à gauche. Notons que, pour tout partage  $\pi = (\pi_1, \dots, \pi_k)$ , nous noterons  $l(\pi)$ , le nombre de parts du partage  $\pi$  ( $l(\pi) = k$ ) et  $\pi'$  le conjugué, c'est à dire le partage obtenu en lisant le diagramme de Ferrers colonne par colonne. De plus, nous considérerons que  $\pi_i = 0$  si  $i < 1$  ou  $i > l(\pi)$ .

---

\*Travail effectué dans le cadre de l'Action de Recherche Coopérative INRIA Alcophys

FIG. 1: *Pile de sable dans IPM(1)*FIG. 2: *Règle verticale*FIG. 3: *Règle horizontale*FIG. 4: *Piles de sable d'aire 6*FIG. 5: *Construction de la pile de sable  $\pi = (7, 6, 5, 3, 3, 2, 2)$  appartenant à IPM(1)*

Les règles qui permettent le mouvement d'un grain de sable sont les suivantes :

- Règle verticale  $(\pi'_1, \dots, \pi'_{\pi_1}) \rightarrow (\pi'_1, \dots, \pi'_i - 1, \pi'_{i+1} + 1, \dots, \pi'_{\pi_1})$   
si  $\pi'_i - \pi'_{i+1} \geq 2$ . Voir figure 2. Dans ce cas, le glissement est de longueur 0.
- Règle horizontale  $(\pi'_1, \dots, \pi'_{\pi_1}) \rightarrow (\pi'_1, \dots, \pi'_i - 1, \pi'_{i+1}, \dots, \pi'_j, \pi'_{j+1} + 1, \dots, \pi'_{\pi_1})$   
si  $\pi'_i - 1 = \pi'_{i+1} = \dots = \pi'_j = \pi'_{j+1} + 1$ ,  $j > i + 1$ . Voir figure 3. Dans ce cas, le glissement est de longueur  $j - i - 1$ .

Le modèle  $SPM(n)$  (*Sand Pile Model*) a été introduit par Goles et Kiwi [7]. Dans ce modèle, la configuration initiale est faite d'une colonne de  $n$  grains et la seule règle est la règle verticale. L'ensemble des configurations est un sous-ensemble de l'ensemble des partages. Pour obtenir tous les partages de  $n$ , il faut aussi utiliser la règle horizontale. Soit  $\mu = (\mu_1, \mu_2, \dots)$  et  $\lambda = (\lambda_1, \lambda_2, \dots)$  deux partages de  $n$ , alors  $\mu \geq \lambda$  si et seulement si on peut passer de  $\mu$  à  $\lambda$  en utilisant les règles verticale et horizontale. L'ordre ainsi défini est isomorphe à l'ordre de dominance  $L_B(n)$ , introduit par Brylawski [5] (il suffit de prendre les conjugués des partages). Voir figure 4 pour  $n = 6$ . L'ordre (un treillis en fait) est défini de la façon suivante:  $\mu \geq \lambda$  si et seulement si pour tout  $j \geq 1$ ,  $\sum_{i=1}^j \mu'_i \geq \sum_{i=1}^j \lambda'_i$ . Rappelons quelques résultats sur les modèles de piles de sable.

**Théorème 1** [5] *Soit  $n$  un entier. L'ensemble de tous les partages de  $n$  muni de l'ordre précédemment défini est un treillis où l'élément maximal est  $(1, 1, \dots, 1)$ , l'élément minimal  $(n)$  avec  $\inf(\mu, \lambda) = \pi$  avec  $\pi'_j = \min(\sum_{i=1}^j \mu'_i, \sum_{i=1}^j \lambda'_i) - \sum_{i=1}^{j-1} \pi'_i$  et  $\sup(\mu, \lambda) = \alpha$  avec  $\alpha'_j = \max(\sum_{i=1}^j \mu'_i, \sum_{i=1}^j \lambda'_i) - \sum_{i=1}^{j-1} \alpha'_i$  pour tout  $j \geq 1$ .*

Brylawsky [5] a aussi montré que la longueur minimum d'une chaîne maximum est  $2n-3$ . Goles et Kiwi dans [7] ont montré que la longueur maximum d'une chaîne maximale est  $2\binom{l+1}{3} + ll' + 1$ , avec  $n = l' + l(l+1)/2$ ,  $0 \leq l' \leq l$ . Ils ont aussi montré que l'ordre  $SPM(n)$  est un treillis, qui est de plus un sous-ordre de  $L_B(n)$ , calculé la longueur maximum d'une chaîne maximale et caractérisé le point fixe. Ensuite, Goles, Morvan et Phan [8] ont défini le modèle  $IPM(n, k)$  (*Ice Pile Model*) qui force la longueur des glissements à être inférieur à  $k$ . Ils obtiennent le même genre de résultats que [7]. Le modèle  $L(n, \theta)$  a été introduit par Phan [13]. Dans ce modèle, la configuration initiale est faite d'une colonne de  $n$  grains. Dans ce modèle, seule la règle verticale est utilisée avec la contrainte supplémentaire qu'une colonne ne peut donner un grain à la colonne suivante que si la différence de hauteur entre les colonnes est supérieure à  $\theta$ , avec  $\theta > 0$ . Ainsi il est clair que  $SPM(n) = IPM(n, 1) = L(1)$ .

Le but de ce travail est de mieux comprendre la structure des piles de sable en les décomposant sur des familles plus simples de partages et en les comptant. Dans la section 2, nous étudions le modèle  $IPM(k)$ . Nous trouvons les séries génératrices selon l'aire, la hauteur et la largeur de ces piles de sable ; puis nous établissons des bornes pour le nombre de piles de sable d'aire  $n$  dans  $IPM(k)$  pour  $n$  grand. Nous exposons dans la section 3, la série selon l'aire et la hauteur pour le modèle  $L(\theta)$ . Enfin, dans la section 4, nous proposons un modèle plus général que nous appelons piles de sable de Frobenius.

## 2 Piles de sable dans $IPM(k)$

Rappelons tout d'abord les caractérisations des piles de sable qui appartiennent à  $IPM(n, k)$ .

**Proposition 1** [13] *Une pile de sable appartenant à  $IPM(n, k)$  est un partage  $\pi = (\pi_1, \pi_2, \dots, \pi_l)$  de  $n$ , telle que :*

- $0 \leq \pi_i - \pi_{i+1} \leq k + 1$ , pour  $1 \leq i \leq l$  ;
- Pour tout  $i < j$  avec  $\pi_i - \pi_{i+1} = k + 1$  et  $\pi_j - \pi_{j+1} = k + 1$ , il existe  $z$  tel que  $i < z < j$  avec  $\pi_z - \pi_{z+1} < k$ .

Nous noterons  $IPM(k)$  l'union  $\cup_{n \geq 0} IPM(n, k)$ . Nous donnerons par la suite des expressions des séries génératrices selon le nombre de grains de sable (aire, variable  $q$ ), la hauteur de la pile de sable (nombre de parts, variable  $x$ ) et la largeur de la pile de sable (plus grande part, variable  $y$ ). Puis nous donnerons des bornes pour le nombre de piles de sable dans  $IPM(n, k)$  pour tout  $k \geq 2$ .

### 2.1 Aire et hauteur

Soit  $D_k$  l'ensemble des partages  $\lambda = (\lambda_1, \dots, \lambda_l)$ , tels que  $1 \leq \lambda_i - \lambda_{i+1} \leq k$ , pour  $1 \leq i \leq l$ . Dans la suite, nous utiliserons une notation abusive mais simple. Soit  $\lambda$  un partage dans  $D_k$ . Nous dirons que  $\lambda_i \in D_<$ , si  $1 \leq \lambda_i - \lambda_{i+1} < k$  et que  $\lambda_i \in D_=$ , si  $\lambda_i - \lambda_{i+1} = k$ . Donnons dès maintenant la forme de la série génératrice des piles de sables appartenant à  $IPM(k)$ , selon l'aire (variable  $q$ ) et le nombre de parts (variable

$x$ ). Soit  $S_k(q, x)$  cette série génératrice, i.e.  $S_k(q, x) = 1 + \sum_{\pi \in IPM(k)} x^{l(\pi)} q^{|\pi|}$ .

**Théorème 2** La série génératrice  $S_k(q, x)$  s'exprime sous la forme

$$S_k(q, x) = \sum_{\lambda \in D_k} x^{l(\lambda)} q^{|\lambda|} \prod_{\lambda_j \in D_<} \frac{1}{1 - xq^{\lambda_j}} \prod_{\lambda_j \in D_=} \left( q + \frac{1}{1 - xq^{\lambda_j}} \right). \quad (1)$$

Avant de prouver le théorème, définissons tout d'abord les notions nécessaires.

**Définition 1** Étant donné un entier  $j$ , un pseudo-partage  $\mu$  est un couple  $((\mu_1, \dots, \mu_j), (\mu(1), \dots, \mu(j)))$  tel que  $\mu_i \geq 0$  et  $0 \leq \mu(i) \leq 1$ ,  $1 \leq i \leq j$ .

**Définition 2** Un pseudo-partage  $\mu = ((\mu_1, \dots, \mu_j), (\mu(1), \dots, \mu(j)))$  est dit compatible avec un partage  $\lambda \in D_k$  si et seulement si  $j = l(\lambda)$  et si  $\mu(j) = 1$  alors  $\lambda_j \in D_=$  et  $\mu_j = 0$ . L'aire d'un pseudo-partage  $\mu$  compatible avec un partage  $\lambda \in D_k$  est  $|\mu| = \sum_{i=1}^{l(\lambda)} \mu_i \lambda_i + \mu(j)$ .

Maintenant la preuve du théorème est directe :

**Preuve.** À chaque pile de sable  $\pi = (\pi_1, \dots, \pi_j)$  appartenant à  $IPM(k)$  on peut faire correspondre de façon unique un partage  $\lambda = (\lambda_1, \dots, \lambda_l) \in D_k$  et un pseudo partage  $\mu$  compatible avec  $\lambda$ , tels que  $|\lambda| + |\mu| = |\pi|$ . Chaque part  $\lambda_i$  de  $\lambda$  est dupliquée  $\mu_i$  fois et augmentée de  $\mu(i)$  (cela correspond aux paliers de longueur  $k+1$ ). La bijection inverse est aussi simple.  $\square$

**Exemples.** Sur les figures, on représente  $\lambda$  en gris, les  $\mu_i$  en blanc et les  $\mu(i)$  en noir. A partir de  $\lambda = (6, 5, 4, 3, 2, 1) \in D_6$  et  $\mu = ((2, 0, 0, 2, 0, 1), (0, 0, 1, 0, 1, 0))$ , on obtient  $\pi = (6, 6, 6, 5, 5, 3, 3, 3, 3, 1, 1) \in IPM(1)$ . Voir figure 1.

Une autre façon de prouver le théorème précédent est d'écrire une  $q$ -équation pour  $S_k(q, x)$ . Il suffit ensuite de l'itérer pour retrouver (1).

**Théorème 3**  $S_k(q, x)$  vérifie la  $q$ -équation

$$S_k(q, x) = 1 + \sum_{i=1}^k \frac{xq^i}{1 - xq^i} S_k(q, xq^i) + xq^{k+1} S_k(q, xq^k). \quad (2)$$

**Preuve.** Une pile de sable dans  $IPM(k)$  est

- soit un partage vide (1)
- **règle (I)** : soit pour  $1 \leq i \leq k$ , un partage où l'on duplique  $i$  fois la plus grande colonne et l'on ajoute au moins une part  $i$ ,  $\left( \frac{xq^i}{1 - xq^i} S_k(q, xq^i) \right)$ ,
- **règle (II)** : soit un partage où l'on duplique  $k$  fois la plus grande colonne et l'on ajoute une part de taille  $k+1$ ,  $(xq^{k+1} S_k(q, xq^k))$ .

Il est simple de montrer qu'avec ces deux règles, on peut construire toutes les piles de sable dans  $IPM(k)$ .  $\square$

**Exemple.** La construction de la pile de sable  $\pi = (7, 6, 5, 3, 3, 2, 2)$  appartenant à  $IPM(1)$  est illustrée sur la figure 5. A partir du théorème 2, nous pouvons énoncer les deux résultats suivants :

**Corollaire 1**

$$S_1(q, x) = 1 + \sum_{n \geq 1} x^n q^{n(n+1)/2} \prod_{i=1}^n \left( q + \frac{1}{1 - xq^i} \right).$$

**Corollaire 2**

$$S_\infty(q, x) = \sum_{\lambda \in D_\infty} x^{l(\lambda)} q^{|\lambda|} \prod_j \frac{1}{1 - xq^{\lambda_j}} = \prod_{i \geq 1} \frac{1}{1 - xq^i}.$$

## 2.2 Aire et largeur

Soit  $\mathcal{S}_k(q, y)$  la série génératrice des piles de sables appartenant à  $IPM(k)$ , selon l'aire (variable  $q$ ) et la plus grande part (variable  $y$ ), i.e.  $\mathcal{S}_k(q, y) = 1 + \sum_{\pi \in IPM(k)} q^{|\pi|} y^{\pi_1}$ .

**Théorème 4** *La série génératrice  $\mathcal{S}_k(q, y)$  vérifie la  $q$ -équation :*

$$\mathcal{S}_k(q, y) = \left( \frac{1 - (yq)^{k+1}}{1 - yq} + y^k q^{k-1} \right) \mathcal{S}_k(q, yq) - y^k q^{k-1} \mathcal{S}_k(q, yq^2). \quad (3)$$

**Preuve.** Une pile de sable dans  $IPM(k)$  est

- règle (I) : soit une pile de sable où l'on duplique  $\pi_1$  et pour  $0 \leq l \leq k$ , on l'augmente de  $l$  :  $\left( \frac{1 - (yq)^{k+1}}{1 - yq} \right) \mathcal{S}_k(q, yq)$
- règle (II) : soit une pile de sable avec  $\pi_1 > \pi_2$  où l'on diminue  $\pi_1$  de 1, on la duplique et on lui ajoute  $k+1$  :  $y^k q^{k-1} (\mathcal{S}_k(q, yq) - \mathcal{S}_k(q, yq^2))$ ; puisque la série génératrice des partages  $\pi$  avec  $\pi_1 > \pi_2$  est  $\mathcal{S}_k(q, y) - \mathcal{S}_k(q, yq)$ .  $\square$

On peut alors trouver une expression de la série  $\mathcal{S}_k(q, y)$  pour  $k = 1$ .

**Corollaire 3** *La série génératrice  $\mathcal{S}_1(q, y)$  est*

$$\mathcal{S}_1(q, y) = 1 + \sum_{n \geq 1} y^n q^{n(n-1)/2} \prod_{i=1}^n \frac{1 + q - q^{i-1}}{1 - q^i}.$$

**Preuve.** Omise.  $\square$

## 2.3 Largeur et hauteur

Comptons maintenant nos piles de sable selon la hauteur et la largeur. Soit  $p_k(h, w)$  le nombre de piles de sable appartenant à  $IPM(k)$  de hauteur  $h$  et de largeur  $w$  et

$P_{k,h}(y)$  la série génératrice  $\sum_{w \geq 0} p_k(h, w)y^w$ .

**Théorème 5** La série génératrice  $P_{k,h}(y)$  suit la récurrence

$$P_{k,h}(y) = \begin{cases} 1 & \text{si } h = 0 \\ y \left( \frac{1-y^{k+1}}{1-y} \right) & \text{si } h = 1 \\ y \left( \frac{1-y^{k+1}}{1-y} \right) \left( \frac{1-y^{k+2}}{1-y} - y^{2(k+1)} \right) & \text{si } h = 2 \\ \left( \frac{1-y^{k+1}}{1-y} + y^k \right) P_{k,h-1}(y) - y^k P_{k,h-2}(y) & \text{sinon.} \end{cases} \quad (4)$$

**Preuve.** Une pile de sable appartenant à  $IPM(k)$  de hauteur  $h$  et de largeur  $w$  est :

- soit une pile de sable  $\pi$  de hauteur  $h-1$  et de largeur  $w-i$ , où  $\pi_1$  a été dupliquée et augmentée de  $i$  ( $0 \leq i \leq k$ ),
- soit une pile de sable  $\pi$  de hauteur  $h-1$  et largeur  $w-k$  avec  $\pi_1 > \pi_2$ , où  $\pi_1$  a été diminuée de 1, dupliquée et augmentée de  $k+1$ .

De plus, le nombre de piles de sable  $\pi$  de hauteur  $h-1$  et largeur  $w-k$  avec  $\pi_1 > \pi_2$  est le nombre de piles de sable  $\pi$  de hauteur  $h-1$  et largeur  $w-k$  moins le nombre de piles de sable  $\pi$  de hauteur  $h-2$  et largeur  $w-k$ .  $\square$

Soit  $\mathcal{P}_k(x, y)$  la série génératrice des piles de sable selon la largeur (variable  $y$ ) et la hauteur (variable  $x$ )  $\sum_{h \geq 0} P_{k,h}(y)x^h$ .

**Théorème 6**

$$\mathcal{P}_k(x, y) = \frac{1 - x(y^k + 1 - y^{k+1}) + x^2 y^k (1 - y)}{1 - x \left( \frac{1-y^{k+1}}{1-y} + y^k (1-x) \right)}. \quad (5)$$

**Preuve.** Le résultat découle directement de la récurrence.  $\square$

Soit  $H_{k,h}$  le nombre de piles de sable de hauteur  $h$  appartenant à  $IPM(k)$  et  $H_k(x)$  la série génératrice  $\sum_{h \geq 0} H_{k,h}x^h$ .

**Corollaire 4** La série génératrice  $H_k(x)$  est  $(1-x)/(1-(k+2)x+x^2)$ .

**Preuve.** Directe à partir de l'équation (5) comme  $H_k(x) = \mathcal{P}_k(x, 1)$ .  $\square$

**Corollaire 5** Le nombre  $H_{k,h}$  de piles de sable de hauteur  $h$  appartenant à  $IPM(k)$  est

$$H_{k,h} = \frac{1+A_1}{k+4} A_1^{1-h} + \frac{1+A_2}{k+4} A_2^{1-h},$$

avec  $A_1 = \frac{k+2+\sqrt{k(k+4)}}{2}$  et  $A_2 = \frac{k+2-\sqrt{k(k+4)}}{2}$  (racines de  $1-(k+2)x+x^2=0$ ).

**Preuve.** Il est facile d'écrire  $H_k(x) = \frac{a_1}{1-\rho_1 x} + \frac{a_2}{1-\rho_2 x}$ , avec  $a_1 = (1+A_1)A_1$ ,  $a_1 = (1+A_2)A_2$ , et  $\rho_i = 1/A_i$   $i = 1, 2$ , puis d'utiliser  $1/(1-z) = \sum_{n \geq 0} z^n$ .  $\square$

Soit  $p_k(n)$  le nombre de piles de sable appartenant à  $IPM(k)$  de demi-périmètre (hauteur+largeur)  $n$ . La série génératrice associée  $\sum_{n \geq 0} p_k(n)x^n$  sera notée  $\mathcal{P}_k(x)$ .

**Corollaire 6**

$$\mathcal{P}_k(x) = \frac{(1-x)^2(1-x^{k+1}+x^{k+2})}{1-2x-3x^{k+2}+x^{k+3}+x^{k+1}}.$$

**Preuve.**  $\mathcal{P}_k(x) = \mathcal{P}_k(x, x)$ . □

**Remarque.** Comme on peut le prédir, plus  $k$  est grand plus  $p_k(n)$  est proche de  $2^n$  (puisque on se rapproche de toutes les partages de demi-périmètre  $n$ ). Ceci peut être vérifié avec Maple en utilisant la commande **realroot** sur le polynôme  $1 - 2x - 3x^{k+2} + x^{k+3} + x^{k+1}$ . En effet le comportement asymptotique de  $p_k(n)$  est  $c_k/\rho_k^n$  avec  $c_k \in \mathbb{R}$  et  $\rho_k$  plus petite racine de  $1 - 2x - 3x^{k+2} + x^{k+3} + x^{k+1}$ .

## 2.4 Bornes asymptotiques

Les séries génératrices selon l'aire sont tellement compliquées que l'on ne peut espérer calculer facilement le comportement asymptotique des nombres  $I_{k,n}$  de piles de sable appartenant à  $IPM(n, k)$ . De ce fait, nous allons donner une échelle de fonction, i.e. nous allons trouver une suite  $\{p_k(n)\}_{k \geq 2}$  telle que  $p_{k+1}(n) \leq I_{k,n} \leq p_{k+2}(n)$ . Le résultat est le suivant :

**Théorème 7** *Le nombre  $I_{k,n}$  de piles de sable appartenant à  $IPM(n, k)$  peut être encadré par*

$$C_{k+1} n^{-3/4} \exp(B_{k+1} n^{1/2}) O(1 + n^{-1/4}) \leq I_k(n) \leq C_{k+2} n^{-3/4} \exp(B_{k+2} n^{1/2}) O(1 + n^{-1/4}),$$

avec  $C_k = ((k-1)/(6k^3))^{1/4}/2$  et  $B_k = 2\pi\sqrt{(k-1)/(6k)}$ .

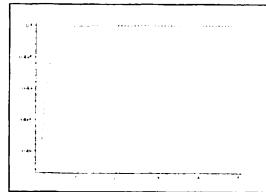
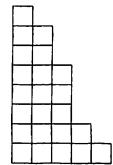
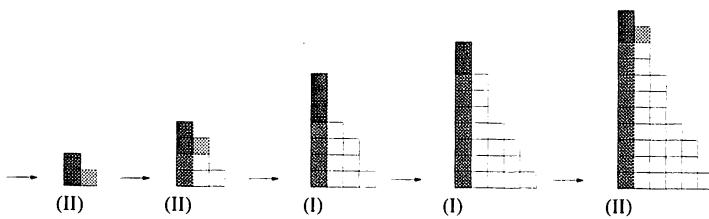
Pour prouver ce théorème nous énonçons tout d'abord le lemme suivant. Pour  $k \geq 2$ , soit  $p_k(n) = [q^n] \prod_{i \geq 1} \frac{1-q^{ki}}{1-q^i}$ , où  $[q^n]f(q)$  est le coefficient de  $q^n$  dans  $f(q)$ .

**Lemme 1** *Quand  $n \rightarrow \infty$ ,*

$$p_k(n) = C_k n^{-3/4} \exp(B_k n^{1/2}) O(1 + n^{-1/4}),$$

avec  $C_k = ((k-1)/(6k^3))^{1/4}/2$  et  $B_k = 2\pi\sqrt{(k-1)/(6k)}$ .

**Preuve.** Nous voulons utiliser le théorème de Meinardus (théorème 6.2 [2]). Montrons tout d'abord que nous avons les conditions requises pour appliquer le théorème. Soit  $f_k(\tau) = \prod_{n=1}^{\infty} (1 - q^n)^{-a_{n,k}} = 1 + \sum_{n=1}^{\infty} p_k(n)q^n$ , où  $q = e^{-\tau}$ ,  $|q| < 1$  et  $a_{n,k} = \begin{cases} 0 & \text{si } n \equiv 0 \pmod{k} \\ 1 & \text{sinon} \end{cases}$ . Nous devons considérer la série de Dirichlet  $D_k(s) = \sum_{n=1}^{\infty} \frac{a_{n,k}}{n^s} = (1 - k^{-s})\zeta(s)$ , ( $s = \sigma + it$ ). Comme  $\zeta(s)$  est convergente pour  $\sigma > \alpha$  avec  $\alpha = 1$ ,  $D_k(s)$  est convergente pour  $\sigma > \alpha$ . De plus, comme  $\zeta(s)$  a un prolongement analytique dans la région  $\sigma \geq -C_0$  avec  $C_0 = 1/2$  et dans cette région,  $\zeta(s)$  est analytique sauf en  $s = 1$  qui est un pôle d'ordre 1 et de résidu  $A = 1$ ,  $D_k(s)$  a un prolongement analytique dans la région  $\sigma \geq -C_0$  et  $D_k(s)$  est analytique dans cette région sauf en  $s = 1$  qui est un pôle d'ordre 1 et de résidu  $A = (k-1)/k$ . Ensuite, il nous faut montrer que  $D_k(s) = O(|t|^{\mathcal{C}_1})$  uniformément dans  $\sigma \geq -C_0$  quand  $|t| \rightarrow \infty$  et avec  $\mathcal{C}_1$  réel positif fixé. En effet, ceci est vrai puisque  $\zeta(s) = O(|t|^{\mathcal{D}_1})$  uniformément dans  $\sigma \geq -1/2$  quand  $|t| \rightarrow \infty$  avec  $\mathcal{D}_1$  réel positif fixé. Comme  $|1 - k^{-s}| \leq 1 + |k^{-s}| = 1 + k^{-\sigma} \leq 1 + k^{1/2}$ , et  $k$  est une constante, on obtient le résultat. Nous utilisons aussi la fonction  $g_k(\tau) = \sum_{n=1}^{\infty} a_{n,k}q^n = \frac{q}{1-q} - \frac{q^k}{1-q^k}$ ,  $q = e^{-\tau}$ . Nous devons nous assurer que si  $\tau = y + 2i\pi x$  ( $x, y$  réels) et  $\arg(\tau) > \pi/4$ ,  $|x| \leq 1/2$ , pour  $y$  suffisamment petit, il existe  $\epsilon > 0$  et  $\mathcal{C}_2 = \mathcal{C}_2(\epsilon)$  (réel positif), tels que  $\Re(g_k(\tau)) - g_k(y) \leq -\mathcal{C}_2 y^{-\epsilon}$ .

FIG. 6: *Évolution de la plus petite racine*FIG. 7: *Une pile  
L(2)*FIG. 8: *Construction d'une pile de sable dans L(2)*

Soit  $f(\tau) = \frac{e^{-\tau}}{1-e^{-\tau}}$ . Alors, pour  $y$  suffisamment petit, il est simple de montrer que  $\Re(f(\tau)) - f(y) \leq -\frac{1+\beta}{y}$ , pour  $y$  suffisamment petit et pour tout  $\beta > 0$ .

Maintenant nous devons borner  $\Re(g_k(\tau)) - g_k(y) = \Re(f(\tau)) - f(y) - (\Re(f(k\tau)) - f(ky))$ . De même, on peut montrer que  $f(ky) - \Re(f(k\tau)) \leq \frac{1}{ky} \leq \frac{1}{2y}$  pour tout  $k \geq 2$  et  $y$  suffisamment petit. Par conséquent  $\Re(g_k(\tau)) - g_k(y) \leq (-1+\beta)/(2y)$ , pour  $y$  suffisamment petit pour tout  $k \geq 2$  et pour tout  $\beta > 0$ . Il suffit donc de prendre  $C_2 = (1+\beta)/2$  avec  $\beta > 0$  et  $\epsilon = 1$ . Maintenant nous avons les conditions requises pour appliquer le théorème de Meinardus et le lemme en découle directement.  $\square$

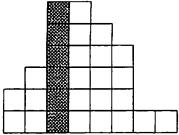
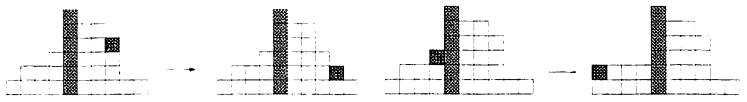
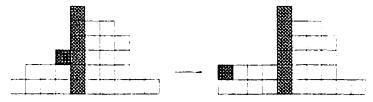
**Preuve du théorème.** Le nombre  $p_k(n)$  est le nombre de partages où les parts peuvent être répétées au plus  $k - 1$  fois. Par conséquent, il est facile de voir à partir de la caractérisation des piles de sable appartenant à  $IPM(n, k)$  que,  $p_{k+1}(n) \leq i_k(n) \leq p_{k+2}(n)$ . Nous n'avons alors qu'à appliquer le lemme.  $\square$

### 3 Piles de sable dans $L(\theta)$

**Proposition 2** [13] *Une pile de sable appartenant à  $L(\theta)$  est un partage  $\pi = (\pi_1, \pi_2, \dots, \pi_l)$  telle que*

- $\pi'_i - \pi'_{i+1} \geq \theta - 1$ , pour  $1 \leq i < \pi_1$  ;
- pour tout  $i < j$  avec  $\pi'_i - \pi'_{i+1} = \theta - 1$  et  $\pi'_j - \pi'_{j+1} = \theta - 1$ , il existe  $z$  avec  $i < z < j$  tel que  $\pi'_z - \pi'_{z+1} \geq \theta$ .

Voir figure 7 pour exemple de partage appartenant à  $L(\theta)$ . Nous voulons compter les partages appartenant à  $L(\theta)$  pour tout  $\theta$  entier positif selon leur aire (variable  $q$ ) et

FIG. 9: *Frobenius*FIG. 10: *Règle verticale*FIG. 11: *Règle horizontale*

leur hauteur (variable  $x$ ). Nous notons la série génératrice correspondante  $\mathcal{L}_\theta(q, x)$ .

**Lemme 2**  $\mathcal{L}_\theta(q, x)$  vérifie la  $q$ -équation

$$\mathcal{L}_\theta(q, x) = \frac{1 - (xq)^\theta}{1 - xq} + \left( \frac{(xq)^\theta}{1 - xq} + x^\theta q^{\theta+1} \right) \mathcal{L}_\theta(q, xq). \quad (6)$$

**Preuve.** Une pile de sable dans  $L(\theta)$  est une colonne de hauteur  $i$ ,  $0 \leq i \leq \theta - 1$ , ou un partage où la plus haute colonne est dupliquée et augmentée d'au moins  $\theta - 1$  ou un partage où la plus haute colonne est augmentée de 1, dupliquée et augmentée de  $\theta - 1$ .  $\square$

La construction de la pile de sable  $\pi = (6, 5, 5, 4, 3, 3, 3, 2, 2, 2, 1)$  appartenant à  $L(2)$  est illustrée sur la figure 8. Par conséquent, si nous itérons la  $q$ -équation précédente, nous trouvons

**Théorème 8** La série génératrice  $\mathcal{L}_\theta(q, x)$  est

$$\frac{1 - (xq)^{\theta-1}}{1 - xq} + \sum_{n \geq 1} x^{(\theta-1)n} q^{\theta n(n+1)/2} \frac{1 - (xq^{n+1})^\theta}{1 - xq^{n+1}} \prod_{i=1}^n \left( q + \frac{1}{1 - xq^i} \right).$$

**Preuve.** Directe à partir de (6). Notons que l'on peut aussi obtenir une preuve bijective de ce résultat [6].  $\square$

Notons que comme pour  $IPM(k)$ , nous pouvons encadrer le nombre  $l_{n,\theta}$  de partages dans  $L(n, \theta)$  puisque

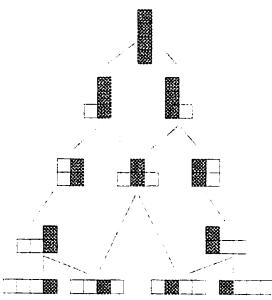
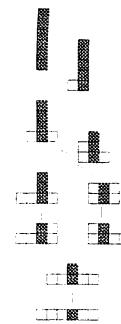
$$p_{\geq \theta-1}(n) \leq l_{n,\theta} \leq p_{\geq \theta}(n),$$

où  $p_{>\theta}(n)$  est le nombre de partages  $\pi$  de  $n$  tels que  $\pi_i - \pi_{i+1} \geq \theta$ . Des formes simples de produits infinis des séries génératrices associées ne sont connues que pour  $\theta = 0, 1, 2$ . De ce fait, nous ne pouvons encadrer facilement  $l_{n,\theta}$  pour tout  $\theta$ .

## 4 Piles de sable de Frobenius

La généralisation la plus naturelle des piles de sable est de laisser les grains tomber plus seulement à droite mais aussi à gauche. Nous appellerons ce nouvel objet *pile de sable de Frobenius*.

**Définition 3** Soit  $j$  un entier. Une pile de sable de Frobenius  $a$  est un couple formé d'un indice pivot  $1 \leq p(a) \leq j$  et d'une suite d'entiers  $(a_1, a_2, \dots, a_j)$  telle que  $0 < a_1 \leq a_2 \leq \dots \leq a_{p(a)} \geq a_{p(a)+1} \geq \dots \geq a_j > 0$ .

FIG. 12:  $L_F(4)$ FIG. 13:  $L_{FB}(6)$ 

Remarquons qu'une pile de sable de Frobenius n'est autre qu'un partage de Frobenius ou  $F$ -partage [10, 3]. Rappelons qu'un partage de Frobenius de l'entier  $n$  est un 3-uplet formé d'un entier  $k$  et de deux suites  $(b_1, \dots, b_k)$  et  $(c_1, \dots, c_k)$  tels que  $b_1 \geq \dots \geq b_k \geq 0$ ,  $c_1 \geq \dots \geq c_k \geq 0$  et  $k + \sum_{i=1}^k (b_i + c_i) = n$ . La bijection entre les deux notations est triviale. Par exemple, sur la figure 9, on a la pile de sable de Frobenius  $(3, (2, 3, 6, 6, 5, 4, 1, 1))$  et le  $F$ -partage  $(6, (2, 2, 1, 0, 0, 0), (5, 3, 3, 3, 2, 1))$ .

Comme pour  $L_B$ , nous définissons le mouvement d'un grain de sable par deux règles : la règle verticale et la règle horizontale illustrées en figures 10 et 11.

**Définition 4** Soit  $a = (p(a), (a_1, a_2, \dots, a_l))$  et  $b = (p(b), (b_1, b_2, \dots, b_m))$  deux piles de sable de Frobenius. Alors  $a \geq_F b$  si et seulement si pour tout  $i, j \geq 0$

$$\sum_{l=p(a)-i}^{p(a)+j} a_l \geq \sum_{l=p(b)-i}^{p(b)+j} b_l.$$

Le modèle que nous étudions est l'ensemble des  $F$ -partages obtenues à partir de  $(n)$  en utilisant les règles horizontale et verticale. Comme pour les partages ordinaires [5], il est simple de montrer que l'on obtient toutes les  $F$ -partages de l'entier  $n$ . Soit  $L_F(n)$  l'ensemble des partages de Frobenius muni de l'ordre  $\geq_F$ .

**Proposition 3**  $L_B(n)$  est un sous-ordre de  $L_F(n)$ .

**Preuve.** Les partages sont des  $F$ -partages pour lesquelles la suite  $b$  n'a que des zéros.  $\square$

**Définition 5** Un demi-treillis est un ordre  $(P, \leq)$  tel que pour tout couple  $(x, y)$  de  $P$ , la borne supérieure existe. De plus si la borne inférieure n'existe pas, alors il n'existe pas d'éléments  $z$  tel que  $z \leq x$  et  $z \leq y$ .

La proposition suivante se prouve aisément :

**Proposition 4**  $L_F(n)$  est un demi-treillis d'élément maximal  $((n), 1)$  et d'éléments minimaux  $((1, 1, \dots, 1), j)$ ,  $1 \leq j \leq n$ .

$L_F(4)$  est représenté sur la figure 12. Pour retrouver un treillis, il suffit de forcer les nombres de grains à droite et à gauche de la colonne à être presque égaux :

**Définition 6** Une pile de sable de Frobenius équilibrée est une pile de sable de Frobenius avec  $0 \leq \sum_{i=1}^{p(a)-1} a_i - \sum_{i=p(a)+1}^l a_i \leq 1$ .

Nous exposons quelques uns des résultats que l'on peut obtenir :

**Lemme 3** *L'ensemble des piles de sable équilibrés  $L_{BF}(n)$  muni de l'ordre  $\geq_F$  est un treillis d'élément maximal  $(1, (n))$  et d'élément minimal  $(\lceil \frac{n+1}{2} \rceil, (1, 1, \dots, 1))$ .*

Par exemple, le treillis  $L_{BF}(6)$  est présenté sur la figure 13. Maintenant que nous avons établi la structure de  $L_F(n)$ , nous allons étudier la longueur des chaînes maximales. Soient  $l$  et  $l'$  tels que  $n = \frac{1}{2}l(l+1) + ll'$  avec  $0 \leq l' \leq l$ .

**Proposition 5** *La longueur maximale d'une chaîne maximale dans  $L_F(n)$  est  $2(\frac{l+1}{3}) + ll' + 1$ .*

**Proposition 6** *Pour  $n \geq 3$ , la longueur minimale d'une chaîne maximale dans  $L_F(n)$  est  $2n - 4$ .*

On peut aussi calculer les bornes supérieure et inférieure de deux éléments dans  $L_F(n)$  puis limiter la largeur des glissements de la règle horizontale. Tout ceci est développé dans [6] et sera exposé dans une version longue de ce résumé étendu.

**Remerciements.** Les auteurs tiennent à remercier J.P. Allouche, M. Latapy, M. Morvan et H. Phan pour leurs propositions constructives lors de l'élaboration de ce travail.

## Références

- [1] R. Anderson, L. Lovász, P. Shor, J. Spencer, E. Tardos, and Winograd S. Disks, ball and walls: analysis of a combinatorial game. *Amer. Math. Monthly*, 96:481–493, 1989.
- [2] G. E. Andrews. *The theory of partitions*, volume 2. Addison-Wesley, 1976.
- [3] G. E. Andrews. Generalized Frobenius partitions. *Mem. Amer. Math. Soc.*, 49(301):iv+44, 1984.
- [4] P. Bak, C. Tang, and K. Wiesenfeld. Self organized criticality. *Phys. rev. A*, 38:364–374, 1988.
- [5] T. Brylawski. The lattice of integer partitions. *Discrete Math.*, 6:201–219, 1973.
- [6] S. Corteel. *Probèmes énumératifs issus de l'Informatique, de la Physique et de la Combinatoire*. PhD thesis, Université Paris-Sud, 2000.
- [7] E. Goles and M. A. Kiwi. Games on line graphs and sand piles. *Theoret. Comput. Sci.*, 115(2):321–349, 1993.
- [8] E. Goles, M. Morvan, and H. D. Phan. Sand piles and order structure of integer partitions. *To appear in Theoretical Computer Science*, 1999.
- [9] E. Goles, M. Morvan, and H.D. Phan. About the dynamics of some systems based on integer partitions and compositions. *To appear in LNCS issue, proceedings of FPSAC'2000*, 2000.
- [10] G. Kreweras. Sur les extensions linéaires d'une famille particulière d'ordres partiels. *Discrete Math.*, 27(3):279–295, 1979.
- [11] M. Latapy, R. Mantaci, M. Morvan, and H. D. Phan. Structure of some sand piles models. *Submitted*, 1999.
- [12] M. Latapy and H. D. Phan. The lattice of integer partitions and its infinite extension. *To appear in LNCS special issue Proceedings of ORDAL'99*, 1999.
- [13] H. D. Phan. *Structures ordonnées et dynamiques de piles de sable*. PhD thesis, Université Paris 7, 1998.
- [14] J. Spencer. Balancing vectors in the max norm. *Combinatorica*, 6:55–65, 1986.

# Un point de vue symbolique sur la logique temporelle linéaire

Jean-Michel Couvreur

LaBRI, Université de Bordeaux I, Talence, France  
 couvreur@labri.u-bordeaux.fr

## Abstract

In this work, we revise the symbolic Büchi automata construction for linear temporal logic. Our main goal was educational: design a technique easy to understand, manage past operators and construct small automata for small formulas. We submit a technique which answers to this three objectives and which is more efficient than the previous symbolic constructions. We generalize our result to an extended temporal logic. This allows us to handle the complementation problem for Büchi automata in an optimal way.

## Résumé

Dans ce travail, nous proposons de réviser les méthodes de construction symbolique d'un automate pour une formule de logique temporelle linéaire. Notre objectif initial était pédagogique : développer une technique facile à comprendre intégrant les opérateurs du passé et construisant de petits automates pour de petites formules. La technique que nous proposons répond à ces trois objectifs et est plus efficace que les techniques symboliques existantes. Nous avons généralisé notre résultat à une logique temporelle linéaire étendue. Celle-ci nous permet de traiter le problème du complément d'un automate de Büchi de manière optimum.

## 1 Introduction

La logique temporelle est un langage puissant permettant de décrire des propriétés de sûreté, d'équité et de vivacité de systèmes. Il est utilisé comme langage de spécification dans des outils tel que SPIN [4] et SMV [6]. Cependant, vérifier qu'un système respecte une telle spécification est PSPACE-complet [8]. En pratique, les techniques de vérification sont confrontées à un problème d'explosion combinatoire du nombre d'états du système et de celui de l'automate codant la formule. De nombreuses techniques ont été élaborées pour faire face à ce problème d'explosion. Nous pouvons noter les techniques de vérification à la volée combinées avec des techniques de réduction *ordre partiel* (SPIN [4]). La représentation symbolique par les diagrammes de décisions permet de coder un système et l'automate d'une formule de manière concise et ainsi de repousser les limites de la vérification (SMV [6]). Chacune de ces méthodes ont leurs succès sur des systèmes industriels prouvant leur bien-fondé.

Dans ce travail, nous proposons de réviser les méthodes de construction symbolique d'un automate pour une formule de logique temporelle linéaire. Notre objectif initial était pédagogique : développer une technique facile à comprendre intégrant les opérateurs du passé et construisant de petits automates pour de petites formules. La

technique que nous proposons répond à ces trois objectifs et est plus efficace que les techniques symboliques existantes [1, 6]. Nous avons généralisé notre résultat à une logique temporelle linéaire étendue. Celle-ci nous permet de traiter le problème du complément d'un automate de Büchi de manière optimum.

## 2 Préliminaires

### 2.1 Logique temporelle linéaire étendue

La logique temporelle linéaire permet d'énoncer des propriétés sur les comportements d'un système. Elle permet de parler de l'évolution d'un système au cours du temps en examinant la suite des états que prend ce système. Les propriétés élémentaires que vérifient ces états jouent un rôle fondamental. Elles introduisent les premières formules d'une logique temporelle : "la propriété  $p$  est vérifiée à l'instant  $t$ ", "la propriété  $p$  est vérifiée dans un instant après  $t$ " et "la propriété  $p$  est vérifiée dans un instant avant  $t$ " sont des formules de logique temporelle linéaire. Pour un ensemble  $AP$  de propositions élémentaires, les modèles de la logique temporelle sont les mots infinis sur l'ensemble  $2^{AP}$  des valuations de  $AP$  (i.e application de  $AP$  dans  $\{0, 1\}$ ). À partir de ces hypothèses, le temps est discret, a un instant initial, est infini et dans un instant donné seulement les valeurs des propositions élémentaires sont examinées.

Nous considérons une logique temporelle linéaire étendue (LTLE) dans laquelle les termes sont construits à partir de propositions élémentaires et d'automates finis étiquetés par des formules. Le critère de satisfaction d'une formule du type *automate* est basé sur la reconnaissance de mots finis ou infinis par des chemins dans l'automate. Cette logique est similaire aux logiques présentées dans [11, 12]. Une des particularités de notre logique est qu'un automate  $A$  permet de définir quatre types de formules : formule finitaire du futur  $A_{F+}$ , formule infinitaire du futur  $A_{L+}$ , formule finitaire du passé  $A_{F-}$ , formule infinitaire du passé  $A_{L-}$ . Les formules du futur sont évaluées pour un mot  $x_0 \cdot x_1 \cdot x_2 \cdots$  et pour un instant  $i$  en reconnaissant le suffixe  $x_i \cdot x_{i+1} \cdot x_{i+2} \cdots$  dans l'automate, alors que pour les formules passées, on considère le préfixe fini renversé  $x_i \cdot x_{i-1} \cdot x_{i-2} \cdots x_0$ . Les formules finitaires et infinitaires se distinguent par leurs critères d'acceptation : une formule finitaire est reconnue par un chemin fini dans l'automate terminant dans un état dit d'acceptation alors qu'une formule infinitaire peut être reconnue par un chemin fini ou infini. La syntaxe d'une formule LTLE est donnée par:

1. Toute proposition  $p \in AP$  est une formule LTLE.
2. Si  $f$  et  $g$  sont des formules LTLE, alors  $\neg f$ ,  $f \wedge g$  sont des formules LTLE.
3. Pour tout automate non déterministe étiqueté par des formules LTLE  $A = \langle S, \rightarrow, s_0, F \rangle$  où  $S$  est un ensemble d'états,  $\rightarrow$  une relation de transitions ,  $s_0$  est un état initial,  $F \subseteq S$  est un ensemble d'états d'acceptations,  $A_{F+}$ ,  $A_{L+}$ ,  $A_{F-}$ ,  $A_{L-}$  sont des formules LTLE.

Nous utilisons la notation standard  $\sigma, i \models f$  pour représenter la satisfaction d'une formule LTLE pour un mot infini  $\sigma$  et un instant  $i$ . L'évaluation de cette relation pour l'instant 0 définit la satisfaction d'une formule pour un mot :  $\sigma \models f \equiv \sigma, 0 \models f$ . La relation  $\models$  est défini par induction de la manière suivante:

1.  $\sigma, i \models p$ ssi  $p \in x_i$  pour  $p \in AP$ ;
2.  $\sigma, i \models \neg f$ ssi  $\neg(\sigma, i \models f)$ ;
3.  $\sigma, i \models f \wedge g$ ssi  $\sigma, i \models f$  et  $\sigma, i \models g$ ;

4.  $\sigma, i \models A_{F+}$  ssi il existe un chemin fini acceptant  $\nu = s_0 \xrightarrow{f_0} s_1 \dots s_{n-1} \xrightarrow{f_{n-1}} s_n$  dans  $A$  tel que  $\forall k < n : \sigma, i+k \models f_k$  and  $s_n \in F$ ;
5.  $\sigma, i \models A_{L+}$  ssi  $\sigma \models A_{F+}$  ou il existe un chemin infini acceptant  $\nu = s_0 \xrightarrow{f_0} s_1 \dots$  dans  $A$  tel que  $\forall k : \sigma, i+k \models f_k$ ,
6.  $\sigma, i \models A_{F-}$  ssi il existe un chemin fini acceptant  $\nu = s_0 \xrightarrow{f_0} s_1 \dots s_{n-1} \xrightarrow{f_{n-1}} s_n$  dans  $A$  tel que  $\forall k < n : \sigma, i-k \models f_k$  and  $s_n \in F$ ;
7.  $\sigma, i \models A_{L-}$  ssi  $\sigma, i \models A_{F-}$  ou il existe un chemin fini acceptant  $\nu = s_0 \xrightarrow{f_0} s_1 \dots s_i$  de longueur  $i$  dans  $A$  tel que  $\forall k : \sigma, i-k \models f_k$ .

On peut aussi définir de manière équivalente LTLE à partir d'opérateurs temporels de la forme  $A_{S\pm}(f_1, \dots, f_n)$  (avec  $S = F, L$ ) où l'automate fini  $A$  est étiqueté par des formules propositionnelles sur  $AP \cup \{f_1, \dots, f_n\}$ . Les opérateurs classiques de logique temporelle linéaire "Next" et "Until", ainsi que leurs versions sur le passé, peuvent être redéfinis en LTLE par :

- $fUg = A_{F+}(f, g)$ ,  $fU^-g = A_{F-}(f, g)$  avec  $A = \langle \{s_1, s_2\}, \{s_1 \xrightarrow{f} s_1, s_1 \xrightarrow{g} s_2\}, s_1, \{s_2\} \rangle$ ;
- $O(f) = A_{F+}(f)$ ,  $O^-(f) = A_{F-}(f)$  avec  $A = \langle \{s_1, s_2, s_3\}, \{s_1 \xrightarrow{1} s_2, s_2 \xrightarrow{f} s_3\}, s_1, \{s_3\} \rangle$ ;

## 2.2 Automates de Büchi à transitions

Le principe des méthodes de vérification automatique est d'associer à toute formule un automate fini acceptant les mêmes mots. L'acceptation d'un mot par ces automates repose sur l'existence d'un chemin infini reconnaissant le mot tel que l'ensemble des états parcourus infiniment souvent vérifie un certain critère. Un critère simple est celui des automates de Büchi : le chemin doit passer infiniment souvent par au moins un état d'acceptation. Dans le cadre de notre étude, nous utilisons les automates de Büchi à transitions. Ils possèdent un ensemble de conditions d'acceptations et chaque condition d'acceptation définit un ensemble de transitions. Un chemin est acceptant si il passe infiniment souvent par au moins une transition de chaque condition d'acceptation. Formellement, un automate de Büchi à transition possède les composantes suivantes :

- $S$  est un ensemble fini d'états;
- $\rightarrow \subseteq S \times 2^{AP} \times S$  est une relation de transitions
- $Acc$  est un ensemble de conditions d'acceptation :  $\forall a \in Acc, a \subseteq \rightarrow$ ;
- $S_0$  est un ensemble d'états initiaux.

Un mot infini  $\sigma = x_0.x_1.x_2 \dots$  sur l'alphabet  $2^{AP}$  est accepté par un automate de Büchi à transitions si et seulement si il existe un chemin infini dans  $A$

$$\rho = s_0 \xrightarrow{x_0} s_1 \xrightarrow{x_1} s_2 \xrightarrow{x_2} \dots$$

tel que  $\forall a \in Acc, \forall i \geq 0, \exists j \geq i : \langle s_j, x_j, s_{j+1} \rangle \in a$ .

Un automate de Büchi à transitions peut être défini symboliquement en identifiant les états par des vecteurs de booléens. Cela offre un moyen simple et concis de les construire et de les définir. Considérons un tableau de variables booléennes  $Now$  de la taille des vecteurs de booléens. Un état associe une valeur booléenne à chaque variable booléenne. Un ensemble d'états est alors représenté par une formule booléenne des

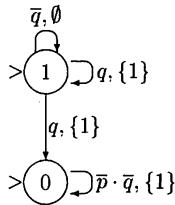


Figure 1: Représentation explicite d'un automate de Büchi à transitions

variables du tableau  $Now$ , cette formule rendant vrai pour les états de l'ensemble. De la même manière, une transition est un vecteur de booléens codant l'état source, l'état destination et les valeurs des propositions élémentaires. En utilisant un tableau  $Next$  pour identifier l'état destination, une transition associe des valeurs à chaque variable booléenne des tableaux  $Now$  et  $Next$ , ainsi qu'à toutes les propositions élémentaires. Une relation de transition et une condition d'acceptation peuvent être représentées par des formules booléennes. Nous utilisons les notations suivantes pour définir symboliquement un automate :

- $Now$  et  $Next$  sont des tableaux de variables booléennes indiquées par un ensemble fini  $VAR$ ,
- $S$  est une formule booléenne des variables du tableau  $Now$ . Elle identifie l'ensemble des états
- $R$  est une formule booléenne des variables de  $AP$ ,  $Now$  et  $Next$ . Elle représente la relation de transition.
- $Acc$  est un tableau de formules booléennes des variables de  $AP$ ,  $Now$  et  $Next$ , indiquées par un ensemble  $ACC$ . Il code les conditions d'acceptation.
- $S_0$  est une formule booléenne des variables du tableau  $Now$ . Elle identifie l'ensemble des états initiaux.

**Exemple 1.** Considérons l'automate donné par sa représentation symbolique :

$$\begin{aligned}
 AP &= \{p, q\} \\
 VAR &= \{1\} \\
 ACC &= \{1\} \\
 S &= 1 \\
 S_0 &= 1 \\
 R &= (\bar{p} + Now[1]) \cdot (Now[1] \Leftrightarrow q + Next[1]) \\
 Acc[1] &= \neg Now[1] + q
 \end{aligned}$$

Nous pouvons obtenir une représentation explicite de l'automate en évaluant dans un premier temps la relation de transitions pour toutes les valeurs des états sources et destinations :  $0 \xrightarrow{\bar{p}, \bar{q}} 0$ ,  $1 \xrightarrow{q} 0$  et  $1 \xrightarrow{1} 1$ . Dans un deuxième temps, on décompose les transitions pour faire apparaître celles qui sont dans l'ensemble  $Acc[1]$ .

### 3 Construction d'un automate pour une formule LTL

Le principe de construction d'un automate pour une formule LTL est dû essentiellement à Lichtenstein, Pnueli, Vardi et Wolper [5, 10]. Une approche symbolique de ces construc-

Formule	VAR	$S_0$
$f$	$f$	$Now[f]$
$O(g)$	$g$	
$gUh$	$gUh$	
$O^-(g)$	$O^-(g)$	$\neg Now[O^-(g)]$
$gU^-h$	$O^-(gU^-h)$	$\neg Now[O^-(gU^-h)]$

Table 1: Définition du vecteur d'état et des états initiaux

Formule	$R$	$Acc$
$f$	$Now[f] \Leftrightarrow f$	
$O(g)$	$Now[g] \Leftrightarrow g$	
$gUh$	$Now[gUh] \Leftrightarrow h + g \cdot Next[gUh]$	$\neg Now[gUh] + h$
$O^-(g)$	$Next[O^-(g)] \Leftrightarrow g$	
$gU^-h$	$Next[O^-(gU^-h)] \Leftrightarrow h + g \cdot Now[O^-(gU^-h)]$	

Table 2: Relation de transitions et conditions d'acceptation

tions peut être trouvée dans [1, 6]. Notre méthode ne diffère que très légèrement des constructions précédentes. La différence vient du type d'automate que l'on construit. Ces techniques sont basées sur les deux assertions suivantes:

$$\begin{aligned} gUh &\equiv h + g \cdot O(gUh) \\ gU^-h &\equiv h + g \cdot O^-(gU^-h) \end{aligned}$$

La valeur d'une formule LTL à l'instant 0 pour un mot infini donné dépend uniquement des valeurs de ces sous-formules temporelles  $O(g)$ ,  $gUh$ ,  $O^-(g)$  et  $gU^-h$  pour tous les instants du mot ainsi que des valeurs des propositions élémentaires. Leurs valeurs sont régies par les assertions précédentes. Dans notre construction symbolique, chaque sous-formule est représentée par un bit dans le vecteur d'état et les assertions sont considérées comme des contraintes de la relation de transitions. Afin que ces assertions ne fassent référence qu'à l'instant présent et suivant, nous associons aux bits des formules  $O(g)$  et  $gU^-h$  les formules  $g$  et  $O^-(gU^-f)$ . Un bit spécial est associé à la formule initiale  $f$  dont l'objet est d'identifier les états initiaux. Nous ajoutons comme contraintes que les formules du type  $O^-(g)$  et  $O^-(gU^-h)$  ne peuvent être vérifiée à un instant 0. Nous associons en plus une condition d'acceptation pour toute sous-formule  $gUh$ . En effet une formule  $gUh$  est vérifiée si elle respecte son assertion et si la formule  $h$  est vérifiée dans un instant futur. Notre construction peut se résumer par les deux tables 1, 2 dans laquelle la formule  $f$  représente la formule initiale.

Les formules  $f$ ,  $g$  et  $h$  apparaissant dans les contraintes de la relation de transitions et les conditions d'acceptation doivent être considérées comme des formules propositionnelles sur les propositions élémentaires et les variables des tableaux  $Now$  et  $Next$ . Il suffit d'identifier leurs sous-formules du type  $O(l)$ ,  $l_1Ul_2$ ,  $O^-(l)$  et  $l_1U^-l_2$  par les variables  $Next[l]$ ,  $Now[l_1Ul_2]$ ,  $Now[O^-(l)]$  et  $Next[O^-(l_1Ul_2)]$ .

**Exemple 2.** Construction d'un automate pour la formule  $f = \bar{p}U(p \cdot (\bar{q}U^-q))$

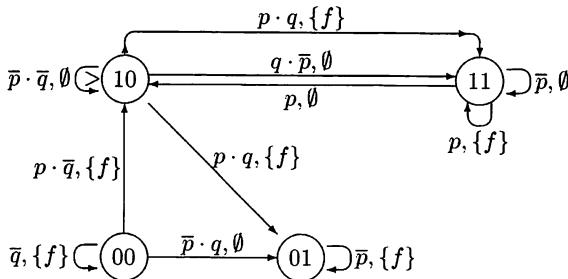


Figure 2: Automate de Büchi à transitions pour  $f = \bar{p}U(p \cdot (\bar{q}U^- q))$

Cette formule accepte les mots tel que la propriété  $q$  est vraie avant la propriété  $p$ . Les états de l'automate sont des vecteurs de bits à deux dimensions :  $VAR = \{f, g\}$  avec  $g = O^-(\bar{q}U^- q)$ . L'ensemble des conditions d'acceptation est réduit à  $\{f\}$ . En effet, la formule ne contient qu'un opérateur du type  $U$ . En appliquant les règles de construction d'un automate, nous obtenons :

$$\begin{aligned} S &= 1 \\ S_0 &= Now[f] \cdot \neg Now[g] \\ R &= (Now[f] \Leftrightarrow p \cdot Next[g] + \bar{p} \cdot Next[f]) \\ &\quad \cdot (Next[g] \Leftrightarrow q + \bar{q} \cdot Now[g]) \\ Acc[f] &= \neg Now[f] + p \cdot Next[g] \end{aligned}$$

La représentation explicite de cet automate est donnée figure 2. Nous pouvons noter que l'état 00 peut être éliminé de l'automate car il est non accessible à partir de l'état initial. Cependant, notre construction code la formule et la négation de formule : en prenant l'état 00 comme état initial, l'automate code  $\neg f$ .  $\square$

**Exemple 3.** Construction d'un automate pour la formule  $f = \Box(p \Rightarrow \Diamond q)$  :

Les opérateurs temporels  $\Diamond$  et  $\Box$  sont définis par  $\Diamond(g) = 1Ug$  et  $\Box(g) = \neg\Diamond(\neg g)$ .  $\Diamond(g)$  exprime que  $g$  sera vérifié au moins à un instant et  $\Box(g)$  exprime l'invariant "g est vérifié à tout instant". Dans le cas d'une formule invariante  $\Box(g)$ , nous pouvons simplifier la construction en n'associant pas  $\Box(g)$  à un bit du vecteur d'états, en ajoutant  $g$  comme contrainte de la relation de transition, et en prenant tous les états pour des états initiaux. Ainsi pour la formule  $f = \Box(p \Rightarrow \Diamond q)$ , nous retrouvons l'automate de l'exemple 1 (Figure 1). L'unique bit du vecteur d'états est associé à la formule  $\Diamond q$ .  $\square$

La justification de notre construction symbolique est simple. Un chemin infini acceptant dans un automate décrit un mot infini vérifiant à chaque instant les formules définies par le vecteur de bit de l'état correspondant. En particulier, le mot infini doit vérifier la formule initiale  $f$ . Cette preuve peut être obtenue par récurrence sur la taille des formules associées aux états. Réciproquement, à un mot infini vérifiant  $f$  correspond un chemin infini dans l'automate tel qu'à tout instant le mot vérifie les mêmes formules que l'état correspondant.

La table 3 compare notre construction avec les principales techniques de la littérature en se basant sur un ensemble de formules proposées dans [3]. La première colonne reprend les résultats données dans [3] pour la première construction développée par Vardi et Wolper [10]. Nous donnons comme résultat le nombre d'états accessibles de l'automate obtenu. Nous distinguons ensuite les constructions locales des construc-

Formules	Première $a \subseteq S$ [10]	Locale		Symbolique	
		$a \subseteq S$ [3]	$a \subseteq \rightarrow$ [2]	$a \subseteq S$ [1]	$a \subseteq \rightarrow$
$pUq$	8	3	2	8	2
$pU(qUs)$	26	4	3	32	3
$\neg((pU(qUs))$	26	7	3	32	3
$GFp \Rightarrow GFq$	114	9	5	28	7
$FpUGq$	56	8	4	32	3
$GpUq$	13	5	4	16	4
$\neg(FFp \Leftrightarrow Fp)$	-	22	3	1	1

Table 3: Comparaison des techniques de construction d'un automate pour LTL

tions symboliques : ces dernières définissent directement l'automate par des contraintes symboliques alors que les premières calculent les états de proche en proche en cherchant à limiter le nombre d'états. Elles se distinguent aussi par le type des conditions d'acceptation : soit elles sont définies par un ensemble d'états, soit par un ensemble de transitions. Bien que ces formules soient minuscules, nous pouvons conclure qu'il est plus efficace de construire des automates de Büchi à transitions. Nous pouvons aussi noter que notre nouvelle technique concurrence les méthodes de construction locale. Ce dernier point constitue une petite surprise dans le sens où cette technique n'a pas été développée pour minimiser le nombre d'états mais pour améliorer et simplifier les techniques de vérification symbolique proposées dans [1].

## 4 Construction d'un automate pour une formule LTLE

La première construction d'un automate pour une formule LTLE est due à Vardi et Wolper [11]. Cette construction se limite aux formules du futur. Nous proposons une construction symbolique simple et plus efficace que la précédente. Elle se base sur les assertions induites par les formules du type automate :

$$\begin{aligned} A_{S+}(s) &\equiv \sum_{s \xrightarrow{g} r: r \notin F} g \cdot O(A_{S+}(r)) + \sum_{s \xrightarrow{g} r: r \in F} g \\ A_{S-}(s) &\equiv \sum_{s \xrightarrow{g} r: r \notin F} g \cdot O^-(A_{S-}(r)) + \sum_{s \xrightarrow{g} r: r \in F} g \end{aligned}$$

Les formules  $A_{S+}(s)$  et  $A_{S-}(s)$  représentent des formules finitaires ou infinitaires du futur et du passé dans lequel l'automate  $A$  a  $s$  comme état initial. La valeur d'une formule LTLE à l'instant 0 pour un mot infini dépend des valeurs des sous-formules  $A_{S+}(s)$  et  $A_{S-}(s)$  pour tous les instants du mot. Leurs valeurs sont régies par les assertions précédentes. Comme pour la construction symbolique d'une formule LTL, chacune de ces sous-formules correspond à un bit dans le vecteur d'état et les assertions définissent des contraintes de la relation de transitions. Pour les formules du passé, la valeur du bit associée à  $A_{S-}(s)$  est celui de la formule  $O^-(A_{S-}(s))$ . Pour traiter les formules du futur, nous introduisons un nouveau bit  $A_{S+}^\alpha(s)$  pour chaque sous-formule  $A_{S+}(s)$ . Il est utilisé pour *marquer* les bits valant 1 pour les formules finitaires  $A_{F+}(s)$  et 0 pour les formules infinitaires  $A_{L+}(s)$ . Leurs valeurs ne respectent les assertions que lorsqu'elles correspondent à un bit marqué. Ce procédé permet de suivre l'évolution d'un sous-ensemble de formules  $A_{F+}(s)$  (resp.  $A_{L+}(s)$ ) valant 1 (resp. 0) pour chaque formule  $A_{F+}$  (resp.  $A_{L+}$ ). Quand une transition permet de prouver localement la validité de cet ensemble de sous-formules, nous marquons dans l'état suivant toutes

Formule	VAR	S	$S_0$
$f$	$f$		$Now[f]$
$A_{F+}$	$A_{F+}(s), A_{F+}^\alpha(s)$	$Now[A_{F+}^\alpha(s)] \Rightarrow Now[A_{F+}(s)]$	
$A_{L+}$	$A_{L+}(s), A_{L+}^\alpha(s)$	$\neg Now[A_{L+}^\alpha(s)] \Rightarrow \neg Now[A_{L+}(s)]$	
$A_{F-}$	$O^-(A_{F-}(s))$		$\neg Now[O^-(A_{F-}(s))]$
$A_{L-}$	$O^-(A_{L-}(s))$		$\neg Now[O^-(A_{L-}(s))]$

Table 4: Définition du vecteur d'état, des états de l'automate et des états initiaux

Formule	Acc
$A_{F+}$	$\prod_{s \notin F} (\neg Now[A_{F+}^\alpha(s)] + \sum_{(g,r): r \in F \wedge s \xrightarrow{g} r} g)$
$A_{L+}$	$\prod_{s \notin F} (Now[A_{L+}^\alpha(s)] + \neg \sum_{(g,r): r \in S \wedge s \xrightarrow{g} r} g)$

Table 5: Définition des conditions d'acceptation

les formules  $A_{S+}(s)$ . Ces transitions définissent les conditions d'acceptations de notre automate ; il en existe une pour chaque sous-formule  $A_{S+}$ . Notre construction peut se résumer par les tables 4, 5 et 6.

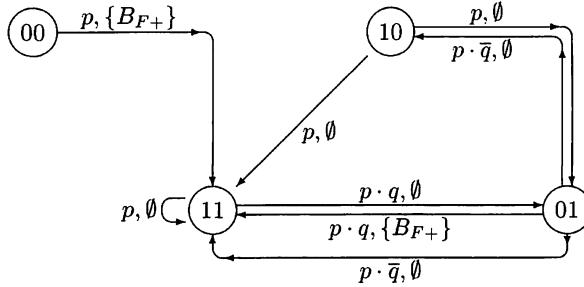
La justification de notre construction est plus difficile que pour celle de LTL. Dans un sens, il est simple de prouver qu'un chemin infini acceptant dans un automate décrit un mot infini vérifiant à chaque instant les formules définies par le vecteur de bits de l'état correspondant. Réciproquement, à un mot infini vérifiant  $f$  correspond plusieurs chemins infinis dans l'automate tel qu'à tout instant le mot vérifie les mêmes formules que l'état correspondant. Nous pouvons cependant construire un chemin acceptant de proche en proche en effectuant le lien entre les formules marquées et des chemins dans les automates des formules  $A_{S+}$ .

**Exemple 4.** Construction d'un automate pour la formule  $\square(B_{F+})$

$$\text{avec } B = \langle \{s_1, s_2, s_3\}, \{s_1 \xrightarrow{p} s_2, s_2 \xrightarrow{p} s_1, s_2 \xrightarrow{q} s_3\}, s_1, \{s_3\} \rangle;$$

Formule	R
$f$	$Now[f] \Leftrightarrow f$
$A_{F+}$	$Now[A_{F+}(s)] \Leftrightarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Next[A_{F+}(r)] + \sum_{s \xrightarrow{g}: r: r \in F} g$
	$Now[A_{F+}^\alpha(s)] \Rightarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Next[A_{F+}^\alpha(r)] + \sum_{s \xrightarrow{g}: r: r \in F} g$
	$Acc(A_{F+}) \Rightarrow \prod_{s \in S} Next[A_{F+}^\alpha(s)] \Leftrightarrow Next[A_{F+}(s)]$
$A_{L+}$	$Now[A_{L+}(s)] \Leftrightarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Next[A_{L+}(r)] + \sum_{s \xrightarrow{g}: r: r \in F} g$
	$Now[A_{L+}^\alpha(s)] \Leftarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Next[A_{L+}^\alpha(r)] + \sum_{s \xrightarrow{g}: r: r \in F} g$
	$Acc(A_{L+}) \Rightarrow \prod_{s \in S} Next[A_{L+}^\alpha(s)] \Leftrightarrow Next[A_{L+}(s)]$
$A_{F-}$	$Next[O^-(A_{F-}(s))] \Leftrightarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Now[O^-(A_{F-}(r))] + \sum_{s \xrightarrow{g}: r: r \in F} g$
$A_{L-}$	$Next[O^-(A_{L-}(s))] \Leftrightarrow \sum_{s \xrightarrow{g}: r: r \notin F} g \cdot Now[O^-(A_{L-}(r))] + \sum_{s \xrightarrow{g}: r: r \in F} g$

Table 6: Définition de la relation de transitions

Figure 3: Automate de Büchi à transitions pour  $\square(B_{F+})$ 

Comme notre formule est du type invariant, nous simplifions notre construction en ajoutant  $Now[B_{F+}(s_1)]$  comme contrainte à la relation de transitions et en prenant tous les états comme états initiaux.

Les états de l'automate sont des vecteurs de bits à quatre dimensions :  $VAR = \{B_{F+}(s_1), B_{F+}^\alpha(s_1), B_{F+}(s_2), B_{F+}^\alpha(s_2)\}$ . L'ensemble des conditions d'acceptation est réduit à  $\{B_{F+}\}$ . En appliquant nos règles de construction, nous obtenons :

$$\begin{aligned}
 S &= (Now[B_{F+}^\alpha(s_1)] \Rightarrow Now[B_{F+}(s_1)]). (Now[B_{F+}^\alpha(s_2)] \Rightarrow Now[B_{F+}(s_2)]) \\
 S_0 &= 1 \\
 Acc(B_{F+}) &= \neg Now[B_{F+}(s_1)] \cdot (\neg Now[B_{F+}(s_2)] + q) \\
 R &= Now[B_{F+}(s_1)] \\
 &\cdot (Now[B_{F+}(s_1)] \Leftrightarrow p \cdot Next[B_{F+}(s_2)]) \\
 &\cdot (Now[B_{F+}^\alpha(s_1)] \Rightarrow p \cdot Next[B_{F+}^\alpha(s_2)]) \\
 &\cdot (Now[B_{F+}(s_2)] \Leftrightarrow p \cdot Next[B_{F+}(s_1)] + q) \\
 &\cdot (Now[B_{F+}^\alpha(s_2)] \Rightarrow p \cdot Next[B_{F+}^\alpha(s_1)] + q) \\
 &\cdot (Acc(B_{F+}) \Rightarrow (Now[B_{F+}(s_1)] \Leftrightarrow Now[B_{F+}^\alpha(s_1)]). \\
 &\quad (Now[B_{F+}(s_2)] \Leftrightarrow Now[B_{F+}^\alpha(s_2)]))
 \end{aligned}$$

La représentation explicite de cet automate est donnée figure 3. Nous n'avons représenté dans les états que les valeurs des bits associées aux indices  $B_{F+}^\alpha(s_1)$  et  $B_{F+}^\alpha(s_2)$ . En effet les deux autres bits valent toujours 1.  $\square$

Une application directe de notre technique est la construction de la négation d'un automate de Büchi. Récemment C. Löding [9] a développé une méthode de transformation d'un automate de Büchi vers un autre type d'automate : les automates faibles alternants. Cette transformation s'adapte parfaitement à notre problème. En effet un automate de Büchi à  $n$  états  $A = \langle S, \rightarrow, F, S_0 \rangle$  s'écrit  $A_{2nL+}$  en LTLE à partir des  $2n + 1$  formules  $A_{0L+}, A_{1F+}, A_{2L+}, A_{3F+} \dots A_{2nL+}$ . Ces automates se déduisent de l'automate  $A$  de la manière suivante :

- Pour l'indice 0, l'automate est simplement défini par  $A_0 = \langle S, \rightarrow, \emptyset, S_0 \rangle$ ,
- Pour les indices pairs autre que 0, l'automate est donné par  $A_{2*i} = \langle S, \rightarrow_{2*i}, \emptyset, S_0 \rangle$  où chaque transition  $\langle s, x, r \rangle$  de  $A$  produit une transition  $\langle s, x \cdot A_{2i-1L+}, r \rangle$  dans  $A_{2*i}$ ,
- Pour les indices impairs, l'automate est donné  $A_{2*i-1} = \langle S \cup \{0\}, \rightarrow_{2*i-1}, \{0\}, S_0 \rangle$  où chaque transition  $\langle s, x, r \rangle$  de  $A$  dont la destination est acceptante ( $r \in F$ ) produit une transition  $\langle s, x \cdot A_{2i-2F+}, 0 \rangle$  dans  $A_{2*i-1}$ , les autres donnent la même transition  $\langle s, x, r \rangle$  dans  $A_{2*i-1}$ .

Notre construction déduit directement pour la formule  $\neg A_{2nL+}$  un automate de Büchi à transitions avec au maximum  $3^{n \cdot (2n+1)}$  états. Nous pouvons réduire le nombre d'états en  $2^{O(n \log n)}$  en remarquant que  $A_{0L+}(s) \Leftarrow A_{1F+}(s) \Leftarrow A_{2L+}(s) \Leftarrow A_{3F+}(s) \Leftarrow \dots \Leftarrow A_{2nL+}(s)$  et en imposant que des formules d'automates différents ne peuvent pas être marquées dans un même état. La transformation d'un automate à transitions en un automate de Büchi laisse le nombre d'états en  $2^{O(n \log n)}$ . Ce coût est essentiellement égal à la borne inférieure du nombre d'états établie par M. Michel et a déjà été atteint par S. Safra [7] par une technique de déterminisation d'automates. Nous obtenons par une méthode symbolique une construction essentiellement optimale de la négation d'un automate de Büchi.

## References

- [1] E. Clarke, O. Grumberg, and D. Long. Verification tools for finite-state concurrent systems. *Lecture Notes in Computer Science*, 803, 1994.
- [2] J.-M. Couvreur. On-the-fly verification of linear temporal logic. In *FM'99*, volume 1708 of *Lecture Notes in Computer Science*, pages 253–271, 1999.
- [3] R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *PSTV95, Protocole Specification Testing and Verification*, pages 3–18, 1995.
- [4] G. J. Holzmann. *Design and Validation of Computer Protocols*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [5] O. Lichtenstein and A. Pnueli. Checking the finite-state concurrent programs satisfy their linear specifications. In *popl85*, pages 97–107, 1985.
- [6] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publisher, Boston, MA, 1994.
- [7] S. Safra. On the complexity of  $\omega$ -automata. In *29th Ann. Symp. on Foundations of Computer Science*, IEEE Computer Society, pages 319–327, 1988.
- [8] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logic. *Journal of the Association for Computing Machinery*, 32(3):733–749, July 1985.
- [9] W. Thomas. Complementation of Büchi automata revisited. *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa, Springer-Verlag*, pages 109–122, 1999.
- [10] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *1th Symposium on Logic in Computer Science*, pages 322–331, 1986.
- [11] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.
- [12] P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.

# A BIJECTION BETWEEN DIAGONALLY CONVEX DIRECTED POLYOMINOES AND EVEN TREES, AND ITS APPLICATIONS

EMERIC DEUTSCH, SVJETLAN FERETIĆ, AND MARC NOY

## Abstract

We present a simple bijection between diagonally convex directed (DCD) polyominoes with  $n$  diagonals and plane trees with  $2n$  edges in which every vertex has even degree (even trees), which specializes to a bijection between parallelogram polyominoes and binary trees. Next we consider a natural definition of symmetry for DCD-polyominoes, even trees, ternary trees, and non-crossing trees, and show that the number of symmetric objects of a given size is the same in all four cases.

## Résumé

On présente une bijection simple entre polyominos dirigés diagonalement convexes (DCD) avec  $n$  diagonales et arbres planaires avec  $2n$  arêtes dont les sommets sont de degré pair (arbres pairs); par restriction on obtient une bijection entre polyominos parallélogrammes et arbres binaires. On considère aussi une définition naturelle de symétrie parmi les polyominos DCD, arbres pairs, arbres ternaires et arbres sans croisement, et on prouve que le nombre d'objets symétriques d'une taille donnée est le même dans les quatre cas.

## 1. INTRODUCTION

The Catalan numbers (M1459)<sup>1</sup>, defined by  $\frac{1}{n+1} \binom{2n}{n}$  for  $n = 0, 1, 2, \dots$ , are probably the most frequently occurring combinatorial numbers after the binomial coefficients. For example, in [17] there are given 66 combinatorial structures that are counted by these numbers.

The next sequence in the hierarchy of generalized Catalan numbers is 1, 1, 3, 12, 55, 273, 1428, ..., defined by  $\frac{1}{2n+1} \binom{3n}{n}$ , (M2926). This sequence also appears in many different contexts. In addition to the four instances that we will consider in more detail, the sequence counts

- (i) paths in the  $(x, y)$  plane from  $(0, 0)$  to  $(2n, n)$  with steps  $(1, 0)$  and  $(0, 1)$ , that never pass above the line  $y = \frac{1}{2}x$  [11];
- (ii) paths in the  $(x, y)$  plane from  $(0, 0)$  to  $(3n, 0)$  with steps  $(1, 1)$  and  $(1, -2)$ , that never pass below the  $x$ -axis [17], p. 169;
- (iii) certain two-line arrays (see [3]);
- (iv) tree-involutions (see [7]; [17], p. 138);
- (v) ways of decomposing the circular permutation  $(1, 2, \dots, n)$  into a product of  $n - 1$  transpositions (see [7]; [8]; [12]; [17], p. 93);
- (vi) ways of associating  $n$  applications of a given nonassociative ternary operation (see [4]; [11]; [17], p. 169);

---

1991 *Mathematics Subject Classification.* 05A15.

*Key words and phrases.* Ternary trees, non-crossing trees, diagonally convex directed polyominoes, plane trees.

<sup>1</sup>These numbers are identifiers of the sequences in *The Encyclopedia of Integer Sequences* [16].

(vii) ways of dissecting a convex  $2n + 2$ -gon into  $n$  quadrilaterals by drawing  $n - 1$  diagonals, no two of which intersect in its interior (see [11]; [17], p. 169).

Next we mention the four families of objects that we will consider.

*Ternary trees.* These are ordered trees in which each node has outdegree 0 or 3 (see [4], [5], [7], [9], [14]; [15], p. 127; [17], p. 92; [18]). The *size* of a ternary tree is defined to be the number of its internal nodes. In Fig. 1a we show a ternary tree of size 4. In all drawings of trees throughout the paper, the root is always the topmost vertex.

*Non-crossing trees.* These are trees drawn on the plane having as vertices the vertices of a convex polygon and whose edges do not cross (see [6]; [7]; [10]; [13]; [17], p. 93). We consider them *rooted*, i. e. one vertex is distinguished. The *size* of a non-crossing tree is defined to be the number of its edges. In Fig. 1b we show a non-crossing tree of size 5

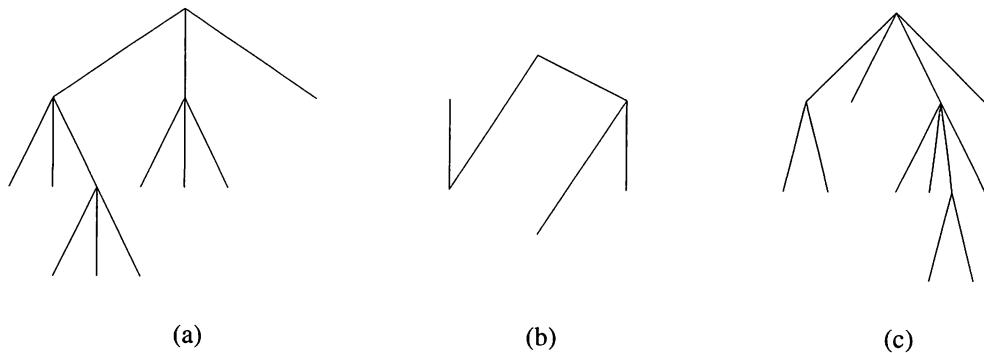


FIGURE 1. a) A ternary tree; b) a non-crossing tree; c) an even tree.

*Even trees.* These are ordered trees in which each node has even outdegree (see [18]). The *size* of an even tree is defined to be half of the number of edges. In Fig. 1c we show an even tree of size 6.

In each of the above three examples, making use of the simple way in which set operations on combinatorial objects translate into operations on their counting generating functions (see, for example, [15]), one can derive at once that the generating function  $g = g(z)$  satisfies the equation

$$g = 1 + zg^3,$$

where  $z$  marks the number of internal nodes, the number of edges, and the number of edge pairs, respectively. From this equation, applying the Lagrange inversion formula, a straightforward computation gives

$$(1) \quad g(z) = \sum_{n=0}^{\infty} \frac{1}{2n+1} \binom{3n}{n} z^n,$$

so that in each case the number of objects of size  $n$  is equal to  $\frac{1}{2n+1} \binom{3n}{n}$ . We also obtain

$$(2) \quad g^2(z) = \sum_{n=0}^{\infty} \frac{1}{n+1} \binom{3n+1}{n} z^n,$$

which will be needed later.

*Diagonally convex directed polyominoes* (DCD-polyominoes). A *polyomino* is a finite connected union of elementary cells (i. e. squares  $[i, i+1] \times [j, j+1] \subseteq \mathbb{R} \times \mathbb{R}$  with  $i$  and  $j$  integers), whose interior is also connected. Polyominoes are defined up to translation. A

polyomino  $P$  is said to be *directed* if every cell of  $P$  can be reached from a distinguished cell, called the *root*, by a path which is contained in  $P$  and uses only North and East steps. We will consider that the cell  $[0, 1] \times [0, 1]$  is the root. A directed polyomino is said to be *diagonally convex* if all cells with centers on a line of slope  $-1$  form a continuous chain of diagonal neighbors. These continuous chains of cells are the *diagonals* of the diagonally convex directed polyomino. The *size* of a DCD-polyomino is defined to be the number of its diagonals. DCD-polyominoes have been investigated in [1], [2], [5], [9], and [14]. For a diagonal  $D$  of a DCD-polyomino we define its *shadow* as the union of the cells that are either upper or right neighbors of the cells of  $D$  (they extend diagonally South-East from the upper neighbor of the top cell of  $D$  to the right neighbor of the bottom cell of  $D$ ). Let  $D_1, D_2, \dots, D_n$  be the diagonals of a DCD-polyomino of size  $n$ , numbered from the root upwards in the NE direction. Clearly,  $D_1$  consists only of one cell (the root). Due to the fact that the polyomino is directed, each  $D_i$  ( $2 \leq i \leq n$ ) lies in the shadow of  $D_{i-1}$ . A DCD-polyomino of size  $n$  can be coded by two sequences  $(a_1, a_2, \dots, a_n)$  and  $(b_1, b_2, \dots, b_n)$ , each consisting of  $n$  nonnegative integers. The number  $a_i$  ( $b_i$ ) is called the *i*th *upper loss* (*lower loss*) and is defined, for  $i \in \{2, 3, \dots, n\}$ , as the number of cells in the shadow of  $D_{i-1}$  and situated above (below) the cells of  $D_i$ , while  $a_1 = b_1 = 0$ . The encoding of DCD-polyominoes by these numbers has been introduced in [9].

To illustrate the above-defined concepts, we consider the DCD-polyomino  $P$  of Fig. 2 (its contour is drawn with a heavy line).  $P$  has size 7, the cells of the seven diagonals being numbered accordingly. The shaded cells form the shadow of the fourth diagonal. The upper losses  $a_i$ , the lower losses  $b_i$ , and the diagonal lengths  $d_i$  of  $P$  are given in Table 1.

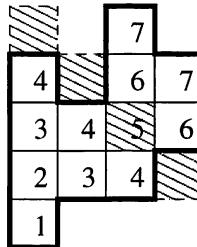


FIGURE 2. A DCD-polyomino of size 7.

$i$	1	2	3	4	5	6	7
$a_i$	0	0	0	0	2	0	0
$b_i$	0	1	0	0	1	0	1
$d_i$	1	1	2	3	1	2	2

TABLE 1

The rest of the paper is organized as follows. In Section 2 we establish a simple bijection between DCD-polyominoes of size  $n$  and even trees of size  $n$ . This bijection has several interesting features that we discuss in detail. In Section 3 we consider a natural definition of symmetry for the four families of objects under study, and show that the number of symmetric objects of size  $n$  is the same in all four cases.

## 2. A BIJECTION BETWEEN DCD-POLYOMINOES AND EVEN TREES

Bijections between diagonally convex directed polyominoes and other combinatorial structures have been defined before. More precisely, rather complex bijections between DCD-polyominoes and ternary trees are given in [5] and [14] and a bijection between DCD-polyominoes with  $n$  diagonals and paths in the  $(x, y)$  plane from  $(0, 0)$  to  $(2n, n)$  with steps  $(1, 0)$  and  $(0, 1)$ , and never passing above the line  $y = \frac{1}{2}x$  is given in [9]. This latter, by composition, yields another bijection between DCD-polyominoes and ternary trees.

In this section we present a new simple bijection between DCD-polyominoes of size  $n$  and even trees of size  $n$ . This bijection, as will be seen later, has the desired feature of commuting with naturally defined symmetries on DCD-polyominoes and on even trees. In addition, (i) under this bijection, the statistics “one-half of the perimeter of a DCD-polyomino” and “number of leaves of an even tree” correspond to each other, and (ii) the restriction of this bijection to parallelogram polyominoes is a bijection between the latter and binary trees.

We consider a diagonally convex directed polyomino of size  $n$ , with upper losses  $a_1, a_2, \dots, a_n$ , lower losses  $b_1, b_2, \dots, b_n$ , and diagonal lengths

$$(3) \quad d_i = i - \sum_{j=1}^i (a_j + b_j) > 0 \quad (i = 1, 2, \dots, n),$$

(see [9], Prop. 1). It is immediate that

$$(4) \quad d_{i-1} = a_i + b_i + d_i - 1 \quad (i = 1, 2, \dots, n; \quad d_0 = 0).$$

The even tree associated to this diagonally convex directed polyomino is constructed in the following manner. We start with a root of degree  $2d_n$ , i. e. we hang  $2d_n$  leaves on the root. We label the leftmost leaf by  $A_1$ , the rightmost one by  $B_1$ , and we hang on them  $2a_n$  and  $2b_n$  leaves, respectively. We have now  $2d_n + 2a_n + 2b_n - 2 = 2d_{n-1}$  unlabeled leaves. Note that  $2d_{n-1} \geq 2$  if  $n > 1$ . We traverse the tree in left (right) preorder<sup>2</sup> and we label the first unlabeled leaf by  $A_2$  ( $B_2$ ). On  $A_2$  and  $B_2$  we hang  $2a_{n-1}$  and  $2b_{n-1}$  leaves, respectively. Now we have  $2d_{n-1} + 2a_{n-1} + 2b_{n-1} - 2 = 2d_{n-2}$  unlabeled leaves. Again  $2d_{n-2} \geq 2$  if  $n > 2$ . We continue in this manner and after the  $n$ th labeling (with  $A_n$  and  $B_n$ ) we will have no unlabeled leaves ( $2d_1 + 2a_1 + 2b_1 - 2 = 0$ ). The obtained even tree is defined to be the image of the given diagonally convex directed polyomino. Making use of (3), one can see easily that the even tree has  $2d_n + 2 \sum_{j=1}^n (a_j + b_j) = 2n$  edges, i. e. its size is  $n$ . In Fig. 3 we show the successive steps for obtaining the even tree corresponding to the DCD-polyomino of Fig. 2.

To obtain the inverse mapping, it is easy to retrace the steps of the above construction. Namely, omitting the root of a given even tree, we label its nodes alternatively in left and right preorder, by  $A_1, B_1, A_2, B_2, \dots, A_n, B_n$ . For each  $i \in \{1, 2, \dots, n\}$  we define

$$a_i = \frac{1}{2} \text{outdegree of } A_{n+1-i}, \quad b_i = \frac{1}{2} \text{outdegree of } B_{n+1-i}.$$

Now, the inverse image of the given even tree is the diagonally convex directed polyomino defined by the sequences of upper losses  $a_1, a_2, \dots, a_n$  and lower losses  $b_1, b_2, \dots, b_n$ .

Next we show that the semiperimeter of a DCD-polyomino is equal to the number of leaves of the corresponding even tree. It is known (Proposition 1 in [9]) and easy to prove that the semiperimeter of a DCD-polyomino is equal to the number of its zero

---

<sup>2</sup>The usual preorder traversal is called here *left preorder traversal* and then, *right preorder traversal* has the obvious meaning.

## STEP

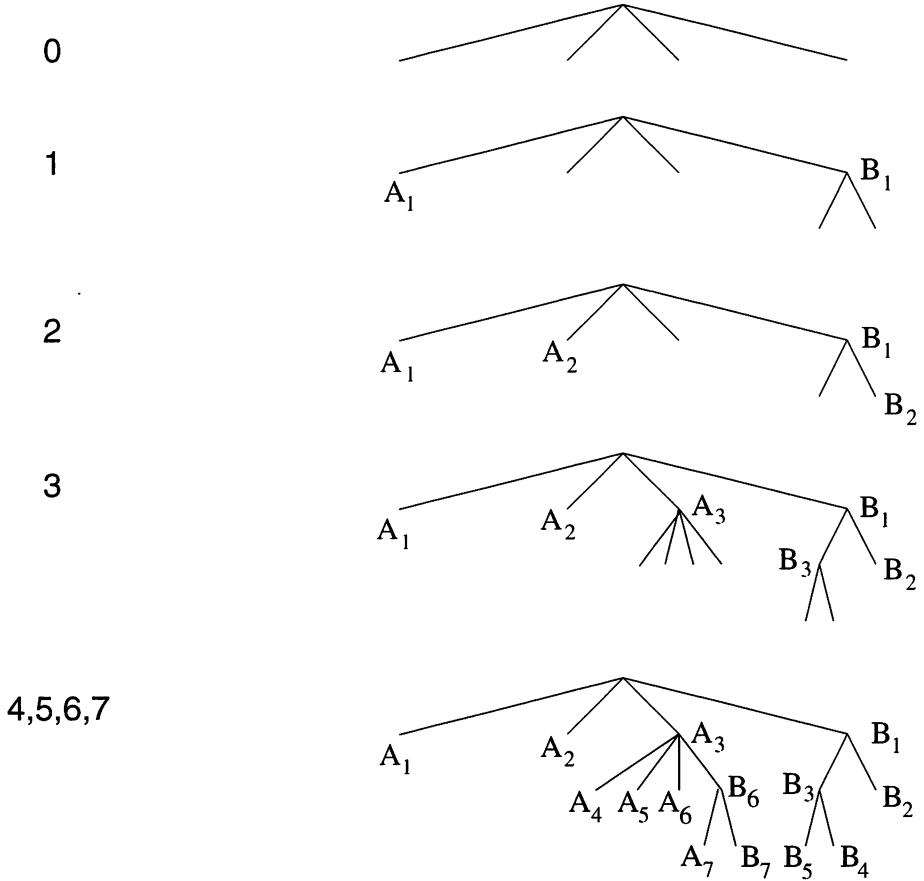


FIGURE 3. Illustrating the bijection.

losses. However, from the definition of our bijection it follows that each zero loss of a DCD-polyomino contributes a leaf to the corresponding even tree.

Another statistics correspondence follows at once from the definition of the bijection. Namely, the length of the last diagonal of a DCD-polyomino (called “pointure” in [14]) is equal to one-half of the root degree of the corresponding even tree.

Finally, a *parallelogram polyomino* is a polyomino bounded by two lattice paths<sup>3</sup> which start together and end together but never meet in between. Equivalently, in a way more convenient for our purpose, a parallelogram polyomino is a DCD-polyomino such that (i) all its losses are equal to 0 or 1 and (ii) the length of the last diagonal is 1. Now, given a parallelogram polyomino, from the definition of the bijection it follows that the corresponding even tree is a binary tree (since we keep hanging either 0 or 2 leaves on the existing nodes). Hence we obtain a bijection between parallelogram polyominoes of semiperimeter  $n$  and binary trees with  $n$  internal nodes.

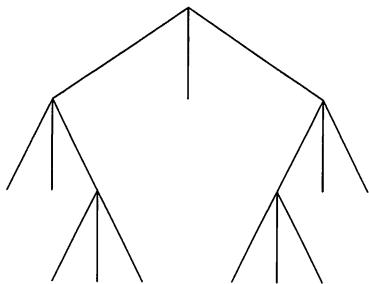
<sup>3</sup>By a *lattice path* we mean a path with steps  $(1, 0)$  and  $(0, 1)$ .

## 3. SYMMETRY

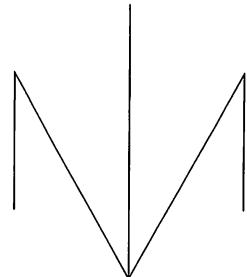
In each of the four families of combinatorial objects that we investigate, we can consider a natural definition of symmetry.

*Ternary trees.* For a ternary tree  $\tau$ , we define its reflection  $\tau^r$  by interchanging recursively left and right children at every node. Then we say that  $\tau$  is *symmetric* if  $\tau = \tau^r$  (see Fig. 4a).

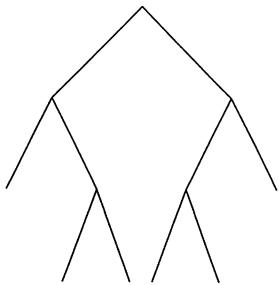
*Non-crossing trees.* For a non-crossing tree  $T$ , we define its reflection  $T^r$  with respect to a bisector through the root. Then we say that  $T$  is *symmetric* if  $T^r = T$  (see Fig. 4b).



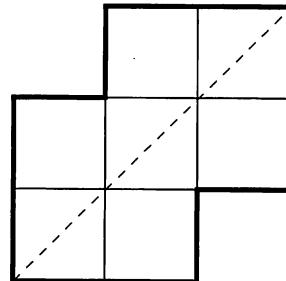
(a)



(b)



(c)



(d)

FIGURE 4. Symmetric objects.

*Even trees.* Reflection and symmetry are defined as for ternary trees (see Fig. 4c for a symmetric even tree).

*Diagonally convex directed polyominoes.* For a DCD-polyomino  $P$ , we define its reflection  $P^r$  with respect to the bisector of the first quadrant. Then we say that  $P$  is *symmetric* if  $P^r = P$  (see Fig. 4d).

**Theorem 1.** *The number of symmetric objects of size  $n$  is the same in all cases. Moreover, it is given by*

$$(5) \quad s_n = \begin{cases} \frac{1}{n+1} \binom{3n/2}{n/2} & \text{if } n \text{ is even;} \\ \frac{2}{n+1} \binom{(3n-1)/2}{(n-1)/2} & \text{if } n \text{ is odd.} \end{cases}$$

This can be shown by using generating functions. If  $s_n$  is the number of symmetric objects of size  $n$ , in each case one shows that

$$(6) \quad S(z) = \sum_{n=0}^{\infty} s_n z^n = g(z^2) + z(g(z^2))^2,$$

where

$$(7) \quad g(z) = 1 + zg(z)^3.$$

For instance, in the case of ternary trees, let  $\tau_1, \tau_2, \tau_3$  be the left, middle and right ternary trees attached to the root of a ternary tree  $\tau$ . If  $\tau$  is symmetric, then  $\tau_2$  is symmetric,  $\tau_1$  is any ternary tree, and  $\tau_3$  is its reflection. Hence

$$S(z) = 1 + zS(z)g(z^2),$$

from where

$$S(z) = \frac{1}{1 - zg(z^2)} = g(z^2) + z(g(z^2))^2.$$

Now, making use of relations (1) and (2), we obtain easily the expressions in (5). The sequence defined by (5) is obtained by alternating the terms of the sequences M2926 and M1782, while the latter is the convolution of the former with itself.

However, instead of the individual enumeration of the symmetric objects in each of the four cases, it seems more interesting to define symmetry preserving bijections between the four families of the considered combinatorial objects.

*A bijection between ternary trees and even trees.* To the empty ternary tree we make correspond the empty even tree. Given a ternary tree  $\tau$  with  $\tau_1, \tau_2, \tau_3$  as before, define recursively an even tree  $\tau'$  as indicated in Fig. 5.

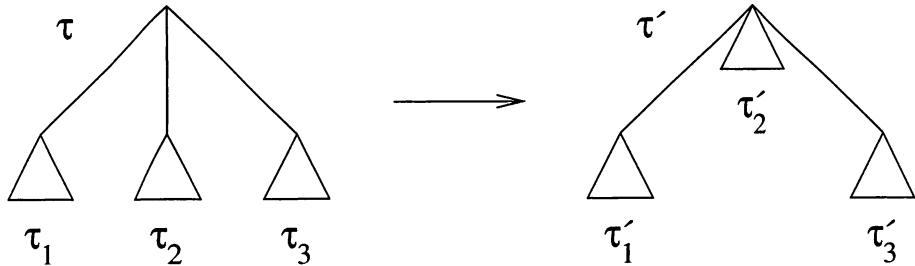


FIGURE 5. From ternary trees to even trees.

It follows easily by induction that through this bijection (i) the size is preserved, (ii) the numbers of left and right leaves of a ternary tree sum up to the number of leaves of the corresponding even tree, and (iii) the length of the central path of the ternary tree is equal to half of the degree of the root of the corresponding even tree. In addition, symmetry is preserved since, clearly,  $(\tau^r)' = (\tau')^r$ .

*A bijection between even trees and non-crossing trees.* We consider a non-crossing tree as an ordered sequence of butterflies attached to the root, where a butterfly is an ordered pair of non-crossing trees (the *wings*) with the roots identified. To the empty even tree we make correspond the empty non-crossing tree. Let  $\varepsilon$  be an even tree, and let  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{2d}$  be the  $2d$  subtrees attached to its root. Then we define recursively a non-crossing tree  $\varepsilon'$  as indicated in Fig. 6.

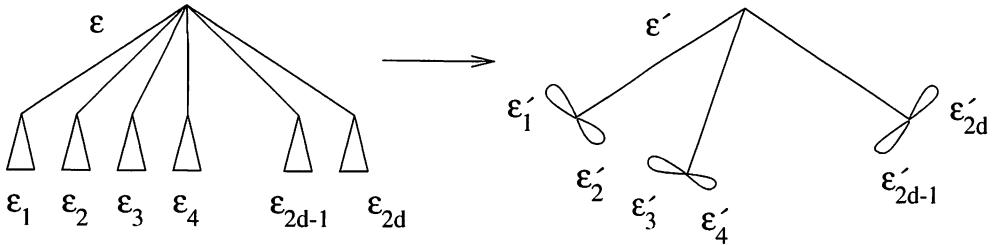


FIGURE 6. From even trees to non-crossing trees.

It follows easily by induction that (i) the size is preserved, (ii) the number of leaves of the even tree is equal to the number of empty wings of the corresponding non-crossing tree, and (iii) half of the degree of the root of an even tree is equal to the degree of the root of the corresponding non-crossing tree. In addition, symmetry is preserved since, clearly,  $(\varepsilon^r)' = (\varepsilon')^r$ .

Finally, a bijection  $P \rightarrow P'$  between DCD-polyominoes and even trees has been defined in Section 2. This bijection preserves symmetry since  $(P^r)' = (P')^r$ . Indeed, the upper (lower) losses of  $P$  are the lower (upper) losses of  $P^r$  and then, by the definition of the bijection, the even tree  $(P^r)'$  is the reflection of the even tree  $P'$ .

*Remark.* From (5) it follows that we have just found four more manifestations of the sequence  $\frac{1}{2n+1} \binom{3n}{n}$ : (i) symmetric ternary trees with  $2n$  internal nodes; (ii) symmetric non-crossing trees with  $2n$  edges; (iii) symmetric even trees with  $4n$  edges; (iv) symmetric DCD-polyominoes with  $2n$  diagonals.

## REFERENCES

- [1] M. Bousquet-Mélou, Percolation models and animals, European J. Combin., 17, 343-369, 1996.
- [2] M. Bousquet-Mélou, Mémoire d'habilitation, LaBRI, Université Bordeaux 1, Rapport No. 1154-96 (1996).
- [3] L. Carlitz, Enumeration of two-line arrays, Fibonacci Quart., 11, 113-130, 1973.
- [4] J. Cigler, Some remarks on Catalan families, European J. Combin., 8, 261-267, 1987.
- [5] M.-P. Delest and J.-M. Fédou, Exact formulas for fully diagonal compact animals, LaBRI, Université Bordeaux 1, Rapport No. 89-06 (1989).
- [6] E. Deutsch and M. Noy, New statistics on non-crossing trees, in *Proc. of the XII Conf. on Formal Power Series and Algebraic Combinatorics*, Moscow, June 2000, to appear in Lecture Notes in Computer Science, Springer.
- [7] S. Dulucq and J.-G. Penaud, Cordes, arbres et permutations, Discrete Math., 117, 89-105, 1993.
- [8] J. A. Eidswick, Short factorizations of permutations into transpositions, Discrete Math., 73, 239-243, 1989.
- [9] S. Feretić and D. Svrtan, Combinatorics of diagonally convex directed polyominoes, Discrete Math., 157, 147-168, 1996.
- [10] P. Flajolet and M. Noy, Analytic combinatorics of non-crossing configurations, Discrete Math., 204, 203-229, 1999.
- [11] P. Hilton and J. Pedersen, Catalan numbers, their generalization, and their uses, Math. Intelligencer, 13, No. 2, 64-75, 1991.
- [12] J. Q. Longyear, A peculiar partition formula, Discrete Math., 78, 115-118, 1989.
- [13] M. Noy, Enumeration of noncrossing trees on a circle, Discrete Math., 180, 301-313, 1998.
- [14] J.-G. Penaud, Animaux dirigés diagonalement convexes et arbres ternaires, LaBRI, Université Bordeaux 1, Rapport No. 90-61, (1990).
- [15] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, Reading, 1996.
- [16] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, 1995.

- [17] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, Cambridge, 1999.
- [18] L. Takács, Enumeration of rooted trees and forests, *Math. Scientist*, 18, 1-10, 1993.

DEPARTMENT OF MATHEMATICS, POLYTECHNIC UNIVERSITY, BROOKLYN, NY 11201  
*E-mail address:* deutsch@magnus.poly.edu

ŠETALIŠTE JOAKIMA RAKOVCA 17, 51000 RIJEKA, CROATIA  
*E-mail address:* sferetic@rijeka.riteh.hr

DEPARTMENT OF APPLIED MATHEMATICS, UNIVERSITAT POLITÈCNICA DE CATALUNYA, PAU GAR-GALLO, 5, 08028 BARCELONA, SPAIN  
*E-mail address:* noy@ma2.upc.es

# A bijection between ordered trees and 2-Motzkin paths and its many consequences

Emeric Deutsch	Louis W. Shapiro
Polytechnic University	Howard University
Brooklyn, NY 11201	Washington, DC 20059
<code>deutsch@magnus.poly.edu</code>	<code>lws@scs.howard.edu</code>

July 13, 2000

## **Abstract**

A new bijection between ordered trees and 2-Motzkin paths is presented, together with its numerous consequences regarding ordered trees as well as other combinatorial structures such as Dyck paths, bushes,  $\{0, 1, 2\}$ -trees, Schröder paths, RNA secondary structures, noncrossing partitions, Fine paths, and Davenport - Schinzel sequences.

## **Résumé**

Une nouvelle bijection entre arbres ordonnés et chemins de Motzkin bicolores est présentée, avec ses nombreuses conséquences en ce qui concerne les arbres ordonnés ainsi que d'autres structures combinatoires telles que chemins de Dyck, buissons, arbres de type  $\{0, 1, 2\}$ , chemins de Schröder, structures secondaires de type RNA, partitions non croisées, chemins de Fine, et enfin suites de Davenport - Schinzel.

## **1 Introduction**

An *ordered tree* is a rooted tree where the order of the subtrees of a vertex is significant. It is well-known that the number of all ordered trees with  $n$  edges is the Catalan number  $C_n = \frac{1}{n+1} \binom{2n}{n}$ . By a 2-Motzkin path we mean paths starting and ending on the horizontal axis but never going below it, with possible steps  $(1, 1)$ ,  $(1, 0)$ , and  $(1, -1)$ , where the level steps  $(1, 0)$  can be either of two kinds: straight and wavy, say. It is less well-known, but easy to show that the number of 2-Motzkin paths with  $n$  steps is given by the Catalan number  $C_{n+1}$ . A bijection between 2-Motzkin paths and Dyck paths (counted by the Catalan numbers) is the following: denoting an up (down) step of a Dyck path or a 2-Motzkin path by  $U$  ( $D$ ), for a given 2-Motzkin path  $\mu$  we replace  $U$  by  $UU$ ,  $D$  by  $DD$ , a straight level step by  $UD$ , and a wavy level step by  $DU$ . Note that the obtained path is a Dyck path except that it can go down to level -1. We place a  $U$  at front and a  $D$  at back in order to fix things up. The Dyck path obtained this way is the image of the 2-Motzkin path  $\mu$ .

In this paper we present a new bijection between ordered trees and 2-Motzkin paths. The bijection is simple but not straightforward. It has numerous consequences regarding ordered trees as well as other combinatorial structures such as Dyck paths, bushes,

$\{0, 1, 2\}$ -trees, Schröder paths, secondary structures, noncrossing partitions, and Fine paths. In the last section we present a new simple bijection between bushes and certain Davenport-Schinzel sequences and then, taking its composition with our main bijection, we obtain another bijection between Motzkin paths and Davenport-Schinzel sequences.

## 2 Tree terminology

As mentioned above, an *ordered tree* is a rooted tree where the order of the subtrees of a vertex is significant. The outdegree of a vertex will be called its *degree*. A vertex of degree zero is called a *leaf*. Contrary to adopted usage, by a *node* we mean a vertex that is neither the root nor a leaf. Thus the vertices of a tree are partitioned into three disjoint sets: the root, the nodes, and the leaves. The nodes of a tree are of two kind: (i) *branch node* if its degree is at least 2 and (ii) *lonely node* if its degree is equal to 1. By a *lonely edge* we mean an edge emanating from a lonely node (edges 1, 3, 10, 15, 18, 19, 21, 24, 26 in Fig. 1). By a *redundant edge* we mean either an edge that is emanating

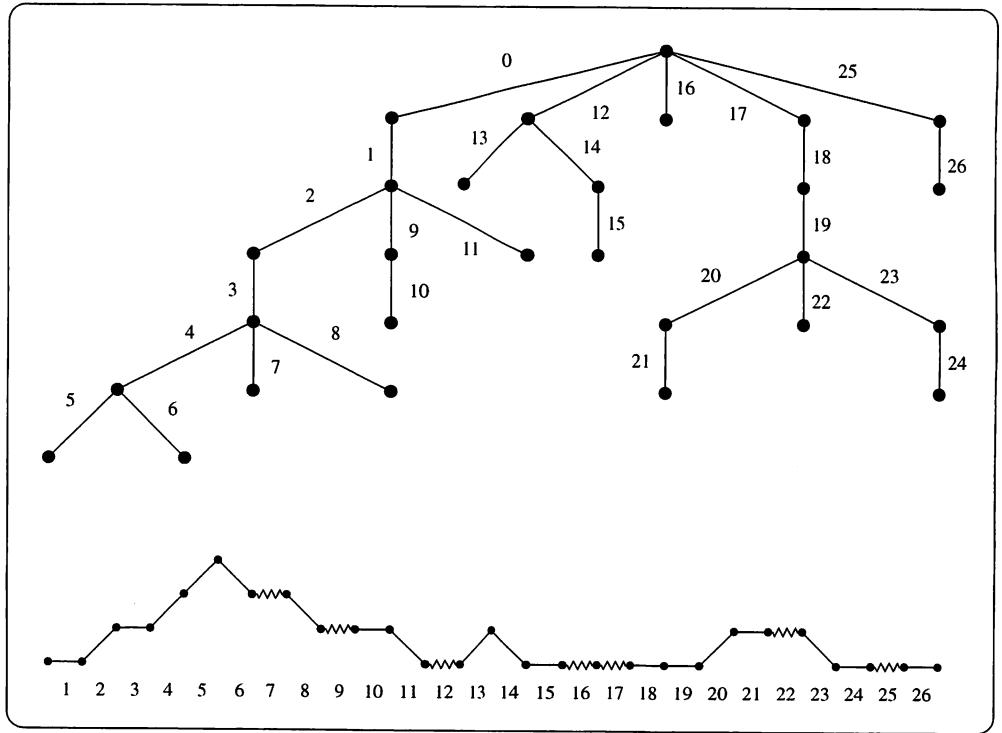


Figure 1: The bijection  $\Phi$

from the root, with the exception of the leftmost edge, or an edge emanating from a branch node, with the exception of the rightmost and leftmost edges (edges 7, 9, 12, 16, 17, 22, 25 in Fig. 1). The leftmost (rightmost) edge emanating from a branch node will be called a *left (right) edge* (edges 2, 4, 5, 13, 20 (6, 8, 11, 14, 23) in Fig. 1). By a *branch* we mean a minimal succession of edges joining either the root with a branch node, or two branch nodes, or a branch node with a leaf. The trees having as roots the

children of the root are called the *subtrees* of the tree. By a *right lonely edge* we mean a lonely edge that is on the rightmost path of some subtree of the tree (edges 1, 15, 18, 19, 24, 26 in Fig. 1). We have the following relation involving some of the newly defined terms:

$$(1) \quad \# \text{ leaves} = 1 + \# \text{ redundant edges} + \# \text{ branch nodes}.$$

### 3 The bijection

Let  $\tau$  be any nonempty ordered tree. Traverse  $\tau$  in preorder and for each edge encountered for the first time (i) do nothing for the first edge; (ii) draw an up step for a left edge; (iii) draw a down step for a right edge; (iv) draw a straight level step for a lonely edge; (v) draw a wavy level step for a redundant edge. It is easy to see that we have obtained a 2-Motzkin path. The described mapping will be denoted by  $\Phi$ . For the inverse mapping, given a 2-Motzkin path, we draw a tree in preorder by the following rule: we start with an edge and then, traversing the 2-Motzkin path from left to right, for each up step we draw a left edge, for each straight level step we draw a lonely edge, for each wavy level step we draw a redundant edge emanating from the appropriate vertex, and for each down step we draw a right edge emanating from the appropriate node. For an example see Fig. 1.

The following correspondences under the bijection  $\Phi$  follow more or less immediately from the definition of the bijection. The last one follows from equality (1).

ordered tree	2-Motzkin path
left edge	up step
right edge	down step
lonely edge	straight level step
right lonely edge	straight level step at level 0
redundant edge	wavy level step
redundant edge emanating from the root	wavy level step at level 0
redundant edge emanating from a node node of degree 2 whose left child is a leaf	wavy level step at level $> 0$
left edge ending in a branch node	peak
right edge ending in a branch node	doublerise, i.e. 2 consecutive up steps
$\# \text{ leaves}$	valley
	$1 + \# \text{ wavy level steps} + \# \text{ up steps}$

### 4 An involution on ordered trees

There is a *trivial involution*  $\pi \rightarrow \pi'$  on the 2-Motzkin paths: interchange the straight and wavy level steps. This involution and our bijection  $\Phi$  induce an involution  $\tau \rightarrow \tau'$  on ordered trees (see example in Fig. 2). This involution on ordered trees can be defined directly, without going through the 2-Motzkin paths. Roughly speaking, traverse the tree in preorder and replace lonely (redundant) edges by redundant (lonely) edges. For trees with 1–3 edges the involution is shown in Fig. 3.

If  $\tau$  is a tree with  $n$  edges and  $\tau'$  is its image under the tree involution, then one shows that

$$(2) \quad \# \text{ leaves}(\tau) + \# \text{ leaves}(\tau') = n + 1.$$

From (2) we have a bijective proof that the number of trees with  $n$  edges and  $k$  leaves is equal to the number of trees with  $n$  edges and  $n + 1 - k$  leaves [2]. An immediate consequence of (2) is

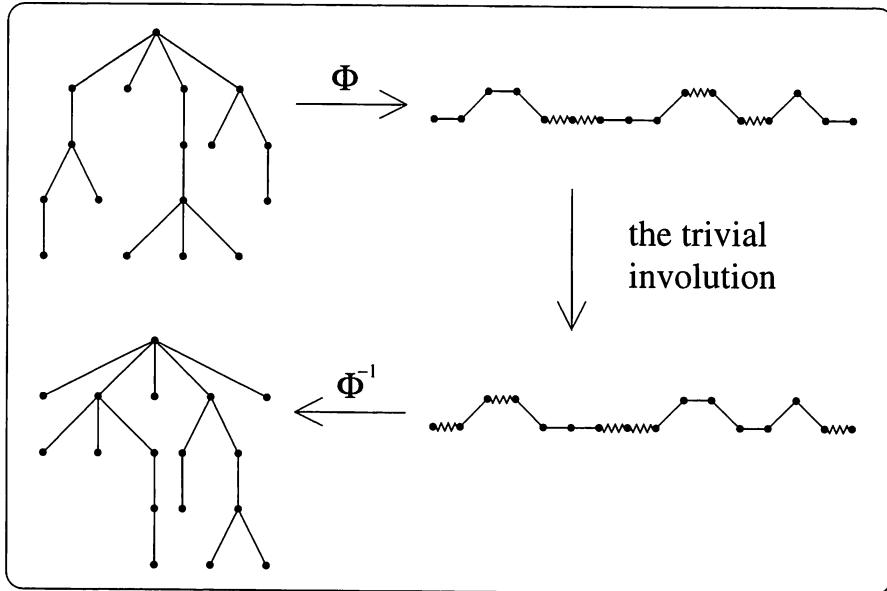


Figure 2:

$$(3) \quad \# \text{ leaves}(\tau) = \# \text{ internal nodes}(\tau'),$$

where, the term “internal node” has the usual meaning (root or node). Thus, we have a bijective proof that the statistics “number of leaves” and “number of internal nodes” are equidistributed, implying, in particular, that in all ordered trees with  $n$  edges, the total number of leaves is equal to the total number of internal nodes.

From trivially equidistributed statistics on 2-Motzkin paths (obtained by interchanging the straight and wavy level steps) we obtain the following nontrivial equidistributions on ordered trees (see the list of correspondences in Section 3): (i) the statistics “number of lonely edges” and “number of redundant edges” are equidistributed; (ii) the statistics “number of right lonely edges” and “number of redundant edges at root level” are equidistributed.

Moreover, the number of ordered trees with  $n$  edges and  $k$  lonely edges (or  $k$  redundant edges) is  $\binom{n-1}{k} M_{n-1-k}$ , where  $M_i$  is a Motzkin number ( $M_0 = 1, M_1 = 1, M_2 = 2, M_3 = 4, \dots$ ). Indeed, if an ordered tree with  $n$  edges has  $k$  redundant edges, then the corresponding 2-Motzkin path has length  $n-1$  and  $k$  wavy level steps. However, it is easy to see that the number of these is  $\binom{n-1}{k} M_{n-1-k}$ .

*Remark.* Making use of various known bijections, we can obtain combinatorial proofs that  $\binom{n-1}{k} M_{n-1-k}$  counts also (i) the number of Dyck paths of length  $2n$  with  $k$  peaks at even level; (ii) the number of Dyck paths of length  $2n$  with  $k$  dud's (i.e. noninitial ascents of length 1); (iii) the number of noncrossing partitions of  $\{1, 2, 3, \dots, n\}$  with  $k$  singleton blocks other than  $\{1\}$ ; (iv) the number of noncrossing partitions of  $\{1, 2, 3, \dots, n\}$  with  $k$  adjacent point pairs in the same block.

As far as the statistics “number of right lonely edges” and “number of redundant edges at root level” are concerned, clearly, they have the same distribution as the statistic “degree of root - 1”.

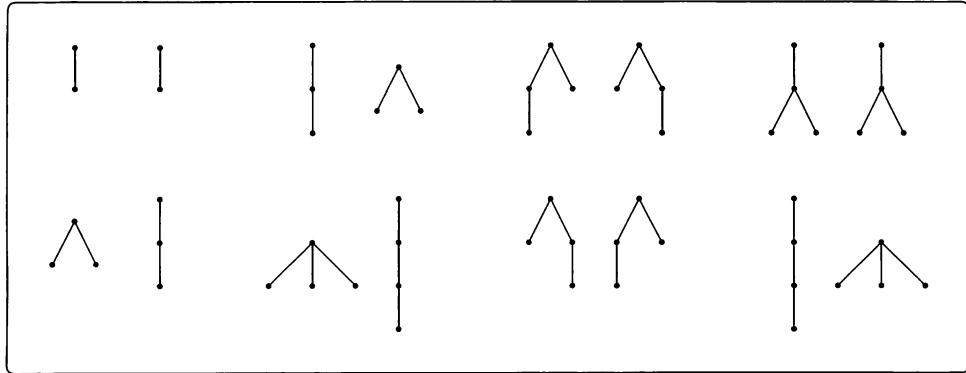


Figure 3:

## 5 Special cases

### 5.1 Planted full binary trees.

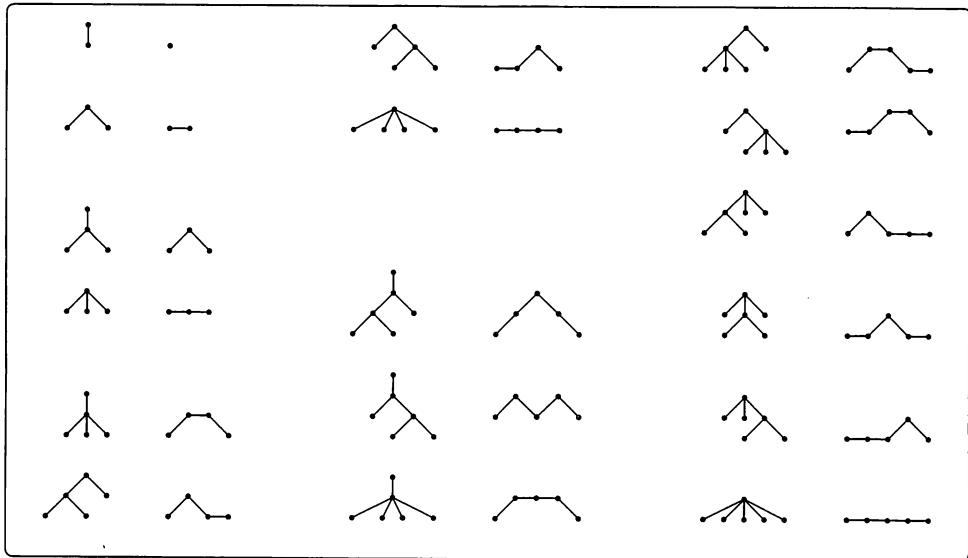
A planted full binary tree has neither lonely nor redundant edges. Consequently, their images under the bijection  $\Phi$  are Dyck paths. This bijection (after removing the “planting stalk”) is nothing but the familiar “glove” bijection (traversing the tree in preorder, to each edge passed on the way down (up) there corresponds a NE (SE) step). The following correspondences between full binary trees and Dyck paths are immediate.

full binary tree	Dyck path
left node	doublerise
right node	valley
left leaf	peak
right leaf (except the last)	doubledescent
level of leftmost leaf	height of first peak
level of rightmost leaf	number of returns

From the obvious equidistribution of left and right statistics on full binary trees we obtain at once the non-obvious but well-known (see for example, [1], [4], [5], [6], [10], [15], [16], [17], [22], [23]) equidistributions of the following pairs of statistics on Dyck paths: (i) “number of doublerises” and “number of valleys”; (ii) “number of peaks” and “1 + number of doublerises”; (iii) “height of first peak” and “number of returns”.

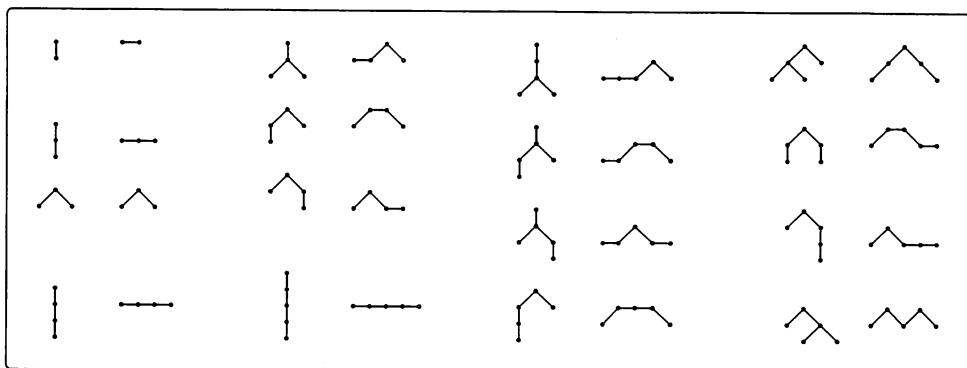
### 5.2 Bushes.

A *bush* is an ordered tree whose nodes have degree at least two [8], [9]. In other words, a bush is a tree with no lonely edges. It follows from here that the restriction of the bijection  $\Phi$  to bushes yields a bijection between bushes and Motzkin paths. Namely, traverse the bush in preorder, do nothing for the first edge, draw an up step for each left edge, draw a level step for each redundant edge, and draw a down step for each right edge. The fact that the number of bushes with a given number of edges is a Motzkin number is well known [8], [9]; this particular bijection may be new (see Fig. 4).

Figure 4: Bijection bushes  $\leftrightarrow$  Motzkin paths

### 5.3 $\{0, 1, 2\}$ -trees.

A  $\{0, 1, 2\}$ -tree is an ordered tree all of whose vertices have degree not exceeding two [8], [9], [12], [20]. In other words, a  $\{0, 1, 2\}$ -tree is a tree which, after a planting stalk is added, has no redundant edges. It follows from here that the restriction of the bijection  $\Phi$  to  $\{0, 1, 2\}$ -trees to which a planting stalk has been added, yields a bijection between  $\{0, 1, 2\}$ -trees and Motzkin paths. Namely, traverse the  $\{0, 1, 2\}$ -tree in preorder, draw an up step for each left edge, draw a level step for each lonely edge, and draw a down step for each right edge. The fact that the number of  $\{0, 1, 2\}$ -trees with a given number of edges is a Motzkin number is well known [8], [9]; this particular bijection may be new (see Fig. 5).

Figure 5: Bijection  $\{0, 1, 2\}$ -trees  $\leftrightarrow$  Motzkin paths (The planting stalk, added to the trees before the bijection, is not shown; nothing corresponds to it under the bijection)

If we apply the bijection  $\Phi$  directly to the  $\{0, 1, 2\}$ -trees, without adding first planting stalks, then we obtain the following new manifestation of the Motzkin numbers: 2-Motzkin paths with at most one wavy level step which, moreover, is at level zero. The  $M_4 = 9$  such paths are given in Fig. 6.

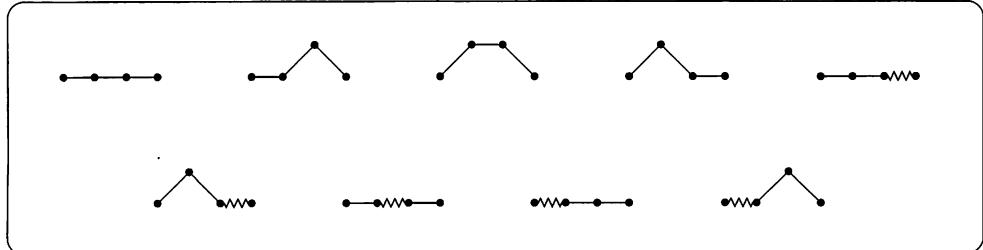


Figure 6: At most one wavy level step which, moreover, is at level zero

*Remark.* The tree involution from Section 5, restricted to bushes and followed by the removal of the edge emanating from the root, yields a bijection between bushes and  $\{0, 1, 2\}$ -trees. Namely, given a bush, traverse it in preorder, replace each redundant edge by a lonely edge, and remove the edge emanating from the root (see Fig. 7).

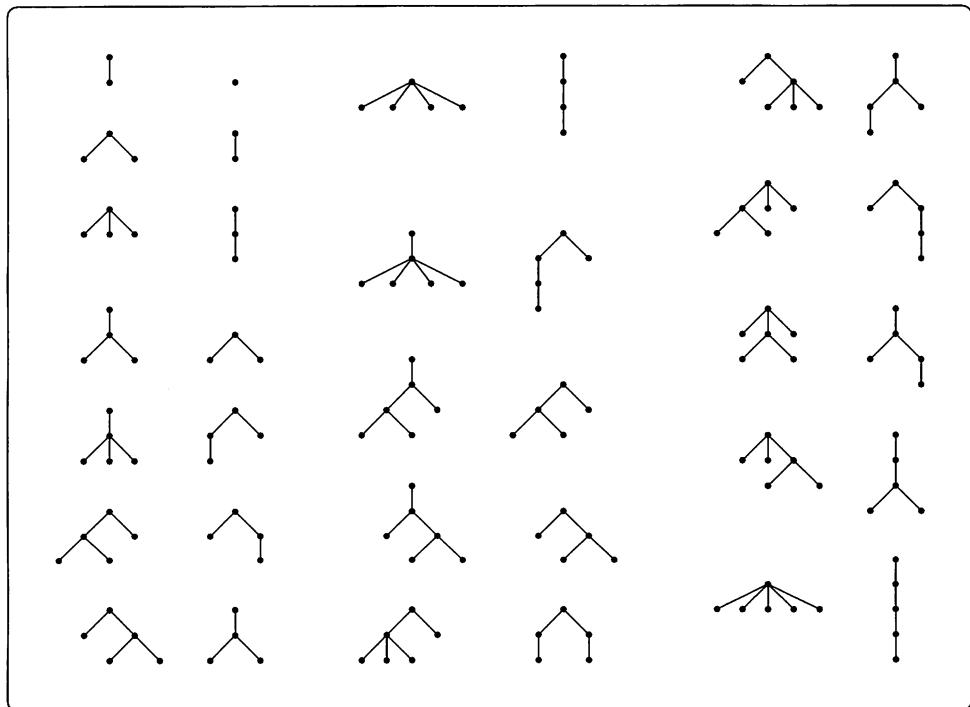


Figure 7: Bijection bushes  $\leftrightarrow \{0, 1, 2\}$ -trees

## 5.4 Schröder paths.

A *Schröder path* of length  $2n$  is a lattice path in the plane from  $(0, 0)$  to  $(2n, 0)$  with steps  $(1, 1)$ ,  $(2, 0)$ , and  $(1, -1)$ , that never go below the horizontal axis. We assume that the level steps of the Schröder paths, viewed as 2-Motzkin paths, are straight and we apply to them the bijection  $\Phi^{-1}$ . We obtain planted trees with nodes of degree at most two and having all branches of odd length. We will call these *Schröder trees*. The Schröder trees with 1, 3, and 5 edges are shown in Fig. 8. Now, after this new

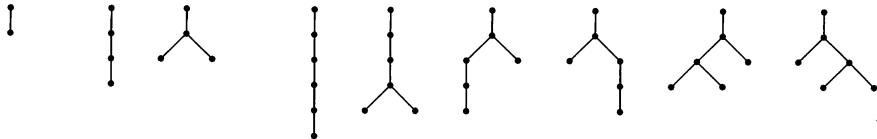


Figure 8: Schröder trees (planted, all nodes have degree  $\leq 2$ , and branches have odd length)

manifestation of the large Schröder numbers has been discovered, we sketch two proofs via generating functions. Let  $G(z)$  be the generating function of the Schröder trees according to number of edges. Each Schröder tree is either a path consisting of an odd number of edges or it is such a path with two Schröder trees hanging at its end. Consequently,  $G = P + PG^2$ , where  $P = z + z^3 + z^5 + \dots = \frac{z}{1-z^2}$ . This equation leads to the known generating function of the Schröder numbers. Alternatively, the Schröder trees can be obtained from the planted full binary trees by replacing edges by paths of odd length. Since the generating function of planted full binary trees is  $zC(z^2)$ , where  $C(z) = \frac{1-\sqrt{1-4z}}{2z}$ , it follows that the generating function of the Schröder trees is  $P(C(P^2))$ , leading again to the desired result.

## 5.5 Secondary structures.

A *secondary (RNA) structure* is a graph (without loops and multiple edges) on the vertex set  $[n]$  such that (i)  $\{i, i+1\}$  is an edge for all  $1 \leq i \leq n-1$ ; (ii) for all  $i$ , there is at most one  $j$  such that  $\{i, j\}$  is an edge and  $|j-i| \neq 1$ , and (iii) if  $\{i, j\}$  and  $\{k, l\}$  are edges with  $i < k < j$ , then  $i < l < j$  [21]. For the sake of simplicity, in the graphical representation of a secondary structure we shall delete all the edges required by condition (i). The obtained graph is a noncrossing partition satisfying requirement (ii) from the definition of a secondary structure.

Secondary structures are in a simple bijection with Motzkin paths without peaks [19]. Indeed, given a secondary structure, we traverse it from left to right and for each isolated vertex we draw a level step, for each vertex where an edge starts we draw an up step, and for each vertex where an edge ends we draw a down step. For an example, see Fig. 9.

Now we apply the bijection  $\Phi^{-1}$  to these Motzkin paths. We can consider two cases.

(i) The level steps of the Motzkin paths, viewed as 2-Motzkin paths, are considered to be straight. In this case we obtain planted trees with nodes of degree at most two and such that the left child of a branch node is not a leaf. Alternatively, removing the

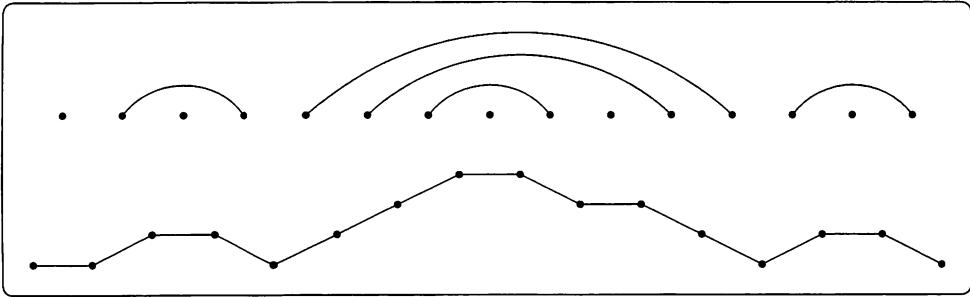
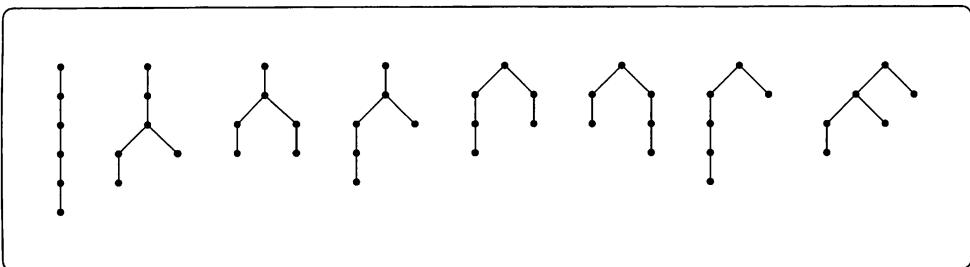


Figure 9:

planting stalk, we have trees with vertices of degree at most two and such that the left child of a vertex of degree two is not a leaf. The eight such trees with five edges are given in Fig. 10. We remark that this result can be obtained also via a bijection due to

Figure 10:  $\{0, 1, 2\}$ -trees in which the left child of a vertex of degree two is not a leaf

Dershowitz and Zaks [3].

(ii) The level steps of the Motzkin paths, viewed as 2-Motzkin paths, are considered to be wavy. In this case we obtain bushes such that the left child of a node of degree 2 is not a leaf. The eight such trees with six edges are given in Fig. 11.

If we apply the Dershowitz-Zaks bijection [3] to these bushes, then we obtain non-crossing partitions satisfying the following two conditions: (a) there are no singletons, except possibly the block  $\{1\}$  and (b) there are no blocks of two consecutive integers, except possibly  $\{1, 2\}$ . The eight such noncrossing partitions on six points are given in Fig. 12.

## 5.6 Fine paths.

By a *Fine path* we mean a Dyck path without peaks of height 1. They are counted by the *Fine numbers*, having generating function  $F(z) = \frac{1-\sqrt{1-4z}}{z(3-\sqrt{1-4z})}$  ( $F(z) = 1 + z^2 + 2z^3 + 6z^4 + 18z^5 + 57z^6 + 186z^7 + \dots$ ). A survey of the Fine numbers can be found in [7]. Applying the bijection  $\Phi^{-1}$  to the Fine paths, we obtain planted full binary trees. Removing the planting stalk, we obtain full binary trees. However, the absence of peaks of height one in the Fine paths implies that the full binary tree has no leaf as the left child of a node on the rightmost path. The six such trees with eight edges are given in

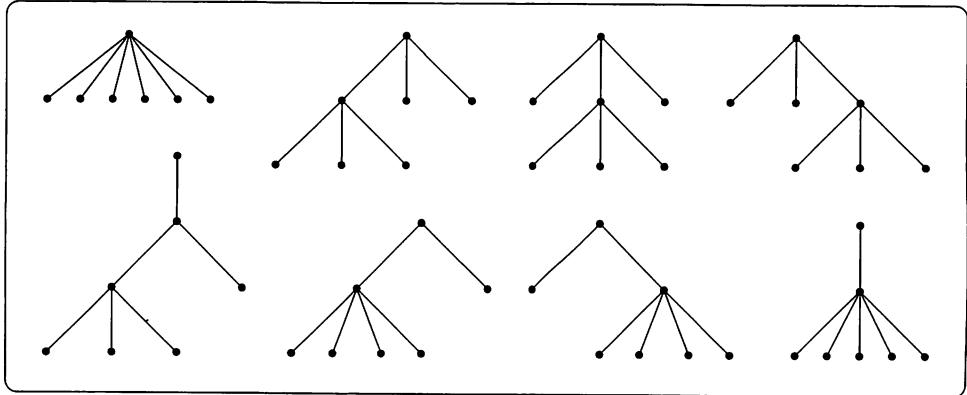


Figure 11: Bushes in which the left child of a node of degree two is not a leaf

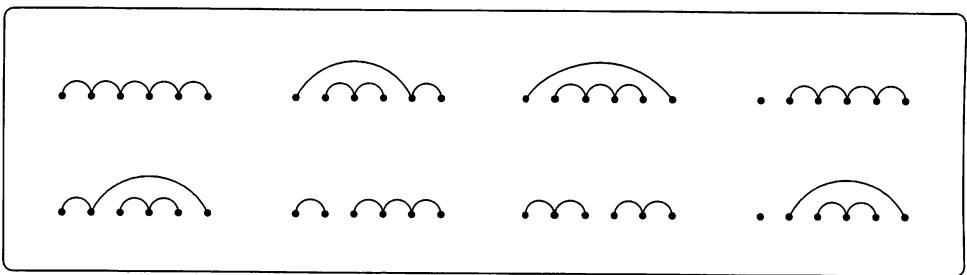


Figure 12: No singletons, except possibly {1}; no  $\{i, i + 1\}$ , except possibly {1, 2}.

Fig. 13.

Making use of a well-known bijection between Dyck paths and ordered trees, it follows at once that the Fine numbers count also the ordered trees with no leaves at level 1. In order to see which 2-Motzkin paths correspond to these trees under the bijection  $\Phi$ , we prefer to look at the 2-Motzkin paths obtained when we do have a leaf at level 1. If a leaf at level 1 is the endpoint of the leftmost edge emanating from the root, then the corresponding 2-Motzkin path is either empty or it starts with a wavy level step. If a leaf at level 1 is the endpoint of the rightmost edge emanating from the root, then the corresponding 2-Motzkin path ends with a wavy level step. Finally, if a leaf at level one is the endpoint of an edge strictly between the leftmost edge and the rightmost edge emanating from the root, then the corresponding 2-Motzkin path contains two consecutive wavy level steps at level zero. Consequently, 2-Motzkin paths that do not start or end with a wavy level step and do not have two consecutive wavy level steps at level zero are counted by the Fine numbers. The six such 2-Motzkin paths with three edges are given in Fig. 14.

It is known [7] that ordered trees having root of even degree are counted by the Fine numbers. Applying to these the bijection  $\Phi$ , we obtain immediately that 2-Motzkin paths with an odd number of wavy level steps at level zero are counted by the Fine numbers. The six such 2-Motzkin paths with three edges are given in Fig. 15.

It is also known [7] that 2-Motzkin paths with no level steps at level zero are counted

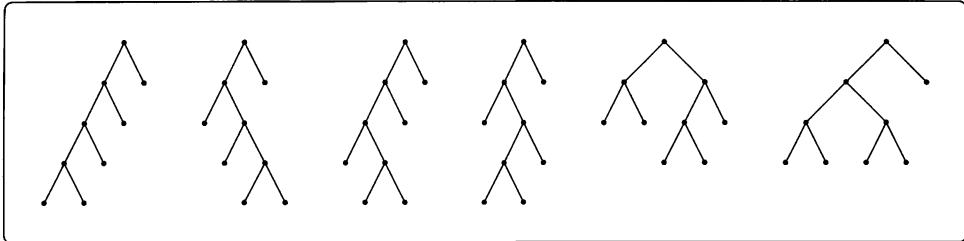


Figure 13: No leaf is the left child of a vertex on the rightmost path

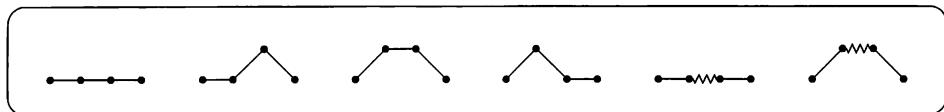


Figure 14: 2-Motzkin paths do not start or end with a wavy level step and do not have two consecutive wavy level steps at level zero

by the Fine numbers. Let us apply the bijection  $\Phi^{-1}$  to these paths. The absence of straight level steps at level zero implies that the corresponding tree has no right lonely edge, while the absence of wavy steps at level zero implies that the corresponding tree is planted. After removing the planting stalks, the obtained trees are characterized by the absence of lonely edges on the rightmost path. Consequently, trees with no vertices (root or node) of degree one on the rightmost path are counted by the Fine numbers. The six such trees with four edges are given in Fig. 16.

## 6 A bijection between Davenport-Schinzel sequences and bushes

In this paper by a *Davenport-Schinzel sequence* (more briefly, a DS sequence) of rank  $n$  we shall mean a finite sequence selected from the set  $[n] \stackrel{\text{def}}{=} \{1, 2, \dots, n\}$  and satisfying the following conditions: (a) each integer  $i \in [n]$  occurs in the sequence; (b) for each pair  $i, j \in [n], i < j$ , the first appearance of  $i$  in the sequence precedes that of  $j$ ; (c) no two adjacent symbols in the sequence are identical; (d) for each pair  $i, j \in [n]$ , the sequence contains no subsequence of the form  $ijij$ . The number of symbols in a DS sequence is called the *length* of the sequence.

Given a bush, using the sequence of positive integers, we label the nodes and the leaves in preorder, except that each node and its youngest child (i.e. the rightmost child) have the same label. It is immediate that the sequence of labels is a DS sequence. Indeed, (i) there are no immediate repetitions in the sequence since a bush has no nodes of outdegree 1 and (ii) the preorder rule precludes subsequences of the form  $ijij$ . The inverse mapping can be easily defined. An example illustrating the bijection is given in Fig. 17.

Obviously, the length of the DS sequence is equal to the number of edges. The rank of the DS sequence is equal to the number of leaves since each label occurs at exactly



Figure 15: 2-Motzkin paths with an odd number of wavy level steps at level zero

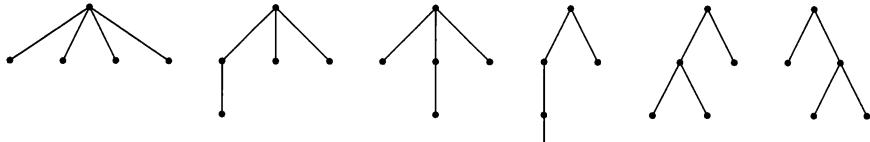


Figure 16: Trees with no vertices of degree one on the rightmost path

one leaf. From here it follows that DS sequences, grouped by length, are counted by the Motzkin numbers and, if grouped by rank, then they are counted by the Schröder numbers [11], [13], [14], [18]. From relation (1) we can easily find that in a bush

$$\# \text{ edges} = 2 \times \# \text{ leaves} - \# \text{ redundant edges} - 1.$$

From here it follows that if the number of leaves is prescribed, then the number of edges is maximum if and only if the number of redundant edges is equal to zero, i.e. if and only if the bush is a planted full binary tree. We obtain, applying the above bijection, that the number of DS sequences with a prescribed rank and having maximal length is given by a Catalan number [18].

Now, if we take the composition of this bijection with the bijection of Sec. 3 or, more precisely, with its restriction to bushes (see Subsection 5.2), then we obtain a bijection between Motzkin paths and DS sequences. This can be described directly: given a Motzkin path, using the sequence of positive integers, label the step endpoints in sequence from left to right, except that points that can be connected by a horizontal line lying strictly under the path have the same label. It is immediate that the sequence of labels, read from left to right, is a DS sequence. The inverse mapping can be easily defined. An example illustrating the bijection is given in Fig. 18.

Obviously, the length of the obtained DS sequence is equal to the the length of the Motzkin path increased by one unit and one can show that the rank of the DS sequence is equal to  $1 + (\text{length of Motzkin path} + \text{number of level steps})/2$ .

*Remark.* This last bijection from Motzkin paths to DS sequences can be trivially modified to a bijection from Schröder paths to DS sequences, in which the rank of the DS sequence is equal to the length of the Schröder path increased by one unit.

## Acknowledgment

The authors are grateful to Guy Giffard for preparing the figures in Xfig.

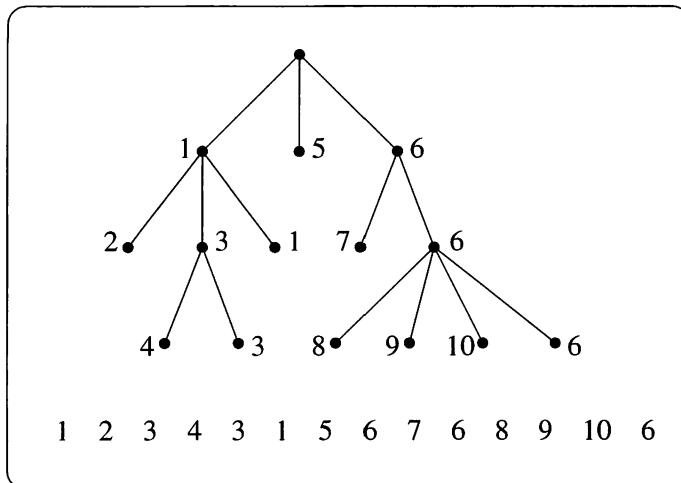


Figure 17:

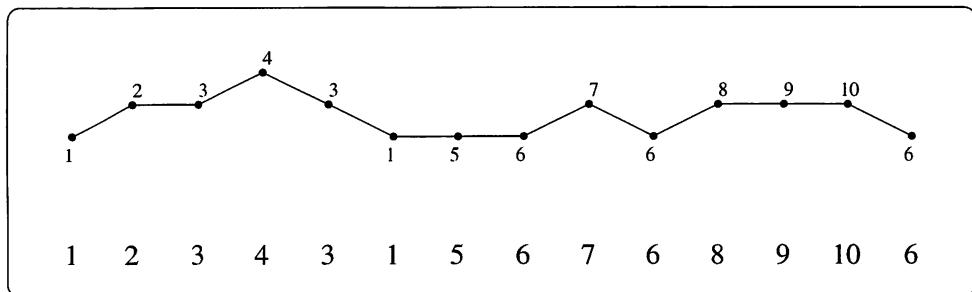


Figure 18:

## References

- [1] S. Bencheikoun and P. Moszkowski, A new bijection between ordered trees and legal bracketings, European J. Combin., 17, 605-611, 1996.
- [2] N. Dershowitz and S. Zaks, Enumerations of ordered trees, Discrete Math., 31, 9-28, 1980.
- [3] N. Dershowitz and S. Zaks, Ordered trees and non-crossing partitions, Discrete Math., 62, 215-218, 1986.
- [4] E. Deutsch, A bijection on Dyck paths and its consequences, Discrete Math., 179, 253-256, 1998. Corrigendum for omitted Fig. 1: 187, 297, 1998.
- [5] E. Deutsch, An involution on Dyck paths and its consequences, Discrete Math., 204, 163-166, 1999.
- [6] E. Deutsch, Dyck path enumeration, Discrete Math., 204, 167-202, 1999.
- [7] E. Deutsch and L. Shapiro, A survey of the Fine numbers, Discrete Math., submitted.

- [8] R. Donaghey, Restricted plane tree representations of four Motzkin-Catalan equations, *J. Combin. Theory Ser. B*, 22, 114-121, 1977.
- [9] R. Donaghey and L. W. Shapiro, Motzkin numbers, *J. Combin. Theory Ser. A*, 23, 291-301, 1977.
- [10] J.-M. Fedou, Grammaires et  $q$ -énumérations de polyominos, Thèse, Univ. Bordeaux I, 1989.
- [11] D. Gardy and D. Gouyou-Beauchamps, Enumerating Davenport-Schinzel sequences, *RAIRO Inform. Théor. Appl.*, 26, No. 5, 387-402, 1992.
- [12] D. A. Klarner, Correspondences between plane trees and binary sequences, *J. Combin. Theory*, 9, 401-411, 1970.
- [13] M. Klazar, On numbers of Davenport-Schinzel sequences, *Discrete Math.*, 185, 77-87, 1998.
- [14] M. Klazar, On *abab*-free and *abba*-free set partitions, *European J. Combin.*, 17, 53-68, 1996.
- [15] G. Kreweras, Sur les éventails de segments, *Cahiers du B. U. R. O.*, No. 15, 3-41, 1970.
- [16] J. C. Lalanne, Une involution sur les chemins de Dyck, *European J. Combin.*, 13, 477-487, 1992.
- [17] J. C. Lalanne, Sur une involution sur les chemins de Dyck, *Theoret. Comput. Sci.*, 117, 203-215, 1993.
- [18] R. C. Mullin and R. G. Stanton, A map-theoretic approach to Davenport-Schinzel sequences, *Pacific J. Math.*, 40, No. 1, 167-172, 1972.
- [19] A. Nkwanta, Lattice paths and RNA secondary structures, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 34, 137-147, 1997.
- [20] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, Reading, 1996.
- [21] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, Cambridge, 1999.
- [22] R. A. Sulanke, Catalan path statistics having the Narayana distribution, *Discrete Math.*, 180, 369-389, 1998.
- [23] R. A. Sulanke, Constraint-sensitive Catalan path statistics having the Narayana distribution, *Discrete Math.*, 204, 397-414, 1999.

# Interprétation bijective d'une récurrence des nombres de Motzkin

Serge Dulucq et Jean-Guy Penaud \*

*LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France*

## Résumé

Cette note a pour objectif de donner une preuve combinatoire d'une formule de récurrence à trois termes vérifiée par les nombres de Motzkin. Elle suit le même cheminement que la preuve bijective de Rémy [3] pour la récurrence des nombres de Catalan, et que la preuve beaucoup plus récente donnée par Foata et Zeilberger [2] pour la récurrence des nombres de Schröder [4].

## Abstract

The purpose of this note is to give a combinatorial proof of the three-term linear recurrence for Motzkin numbers. The present work is inspired by Rémy's combinatorial proof of the linear recurrence for Catalan numbers [3] and the more recent proof given by Foata and Zeilberger [2] for Schröder numbers[4].

## 1 Dyck, Catalan, Motzkin et Schröder

Rémy[3] a le premier donné une interprétation combinatoire de la formule de récurrence pour les nombres de Catalan,

$$\begin{aligned} 2(2n+1)C_n &= (n+2)C_{n+1}, \quad n \geq 1 \\ C_0 &= 1, \end{aligned} \tag{1}$$

nombres qui comptent les mots ou chemins de Dyck. C'est en choisissant, parmi la longue liste des objets énumérés par ces nombres, la classe des *arbres binaires complets*, encore appelés par D. Knuth, les arbres binaires *étendus*, (c'est-à-dire des arbres binaires où tout sommet a exactement 0 ou 2 fils), qu'il a pu interpréter bijectivement cette formule.

Lorsqu'on insère dans les chemins de Dyck des paliers horizontaux, on obtient, en énumérant ces chemins selon la longueur, soit les nombres de Motzkin [1] soit les nombres de Schröder [4], selon qu'on attribue au palier horizontal élémentaire la longueur 1 ou 2. Les nombres de Motzkin vérifient la récurrence à 3 termes suivante,

$$\begin{aligned} (n+2)m_n &= (2n+1)m_{n-1} + 3(n-1)m_{n-2}, \quad n \geq 2 \\ m_0 = m_1 &= 1, \end{aligned} \tag{2}$$

et les nombres de Schröder (simples), la suivante,

$$\begin{aligned} 3(2n-1)s_n &= (n+1)s_{n+1} + (n-2)s_{n-1}, \quad n \geq 2, \\ s_1 = s_2 &= 1, \end{aligned} \tag{3}$$

---

\*Email: dulucq@labri.u-bordeaux.fr, penaud@labri.u-bordeaux.fr.

Récemment, D. Foata et D. Zeilberger[2] ont établi bijectivement la formule de récurrence des nombres de Schröder [4], en introduisant une nouvelle classe d'arbres binaires, dont les sommets sont judicieusement bicoloriés.

Dans le même esprit nous donnons dans cet article une interprétation combinatoire de la récurrence à 3 termes des nombres de Motzkin. Cette interprétation procède en deux étapes,

1. interprétation des deux membres de l'égalité comme les cardinaux de deux ensembles d'objets combinatoires,
2. définition d'une opération sur ces objets prouvant l'égalité.

## La bijection de Rémy

Nous rappelons cette bijection car, outre son élégance, elle constitue un outil important pour l'interprétation que nous présentons.

Le membre gauche de l'égalité (1) énumère les couples constitués d'un arbre binaire complet à  $n$  sommets internes, soit  $2n + 1$  sommets en tout, pointé sur un sommet (interne ou feuille), et d'une direction choisie entre droite et gauche.

Le membre droit énumère les arbres binaires complets à  $n + 1$  sommets internes (soit  $n + 2$  feuilles) pointés sur une feuille.

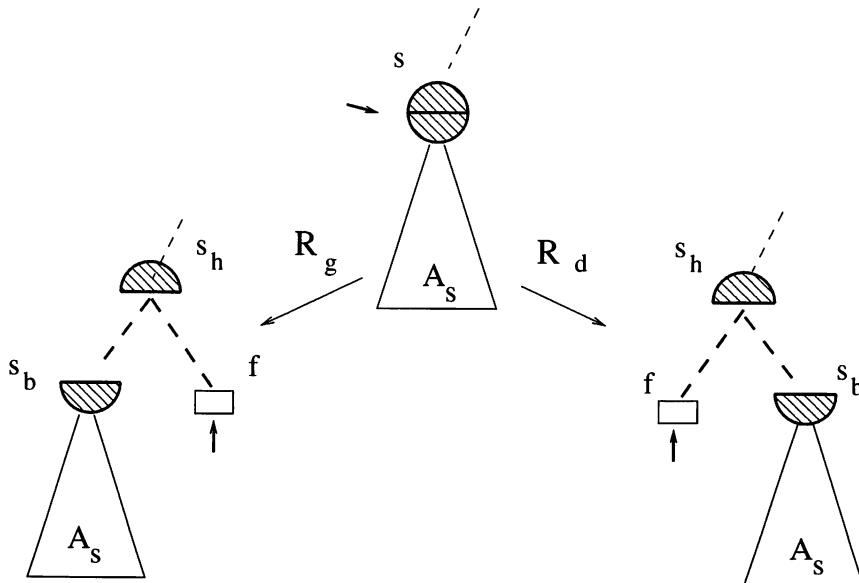


FIG. 1 – La bijection de Rémy.

L'opération bijective définie par Rémy permet de construire un objet du membre droit à partir d'un objet du membre gauche ; elle consiste à séparer en deux parties le sommet pointé  $s$ . La partie haute  $s_h$  reste associée à la partie supérieure de l'arbre, la partie basse devenant un sous arbre de racine  $s_b$ , et si le côté choisi pour la transformation est le côté gauche, alors  $s_b$  devient fils gauche de  $s_h$ , et une feuille  $f$  est ajoutée comme fils droit de  $s_h$  (opération que nous appellerons *Rémy-gauche*), tandis que si le côté droit est choisi on fait l'opération identique en inversant gauche et droite, opéra-

tion que nous appellerons *Rémy-droite*. Dans les deux cas, la feuille ajoutée est la feuille pointée dans l'arbre binaire complet ainsi obtenu.

Cette opération est décrite dans la figure 1. C'est clairement une bijection.

## 2 Arbres binaires penchés

Il est classique d'associer aux nombres de Motzkin  $m_n$ , les arbres 1-2, c'est-à-dire des arbres dessinés plantés dont les sommets ont 0, 1, ou 2 fils, à  $n$  arcs, ou  $n+1$  sommets (on ne représente pas le *tronc*, mais le qualificatif *planté* permet de rompre l'ordre circulaire et de préciser que la racine a un *premier* fils, contrairement aux arbres planaires). Le mot de Motzkin associé s'obtient par un parcours préfixe, en codant un point double par la lettre  $x$ , un point simple par la lettre  $a$  et une feuille (sauf la dernière) par la lettre  $y$ .

Ces arbres sont de façon évidente en correspondance avec la famille dite des arbres binaires *penchés* c'est-à-dire des arbres binaires dont tous les fils uniques sont fils gauches, à  $n+1$  sommets, et également avec la famille des arbres binaires complétés associés, c'est-à-dire à  $n+1$  sommets internes et  $n+2$  feuilles, donc  $2n+3$  sommets en tout, que nous noterons en abrégé *abpc*. Il s'agit clairement d'un codage bijectif.

La figure 2 montre un arbre 1-2 ayant 10 sommets ( $n=9$ ), le mot de Motzkin qui le code, ainsi que l'arbre binaire penché et son complété (avec les 11 feuilles ajoutées mises en évidence par des petits rectangles) qui lui correspondent.

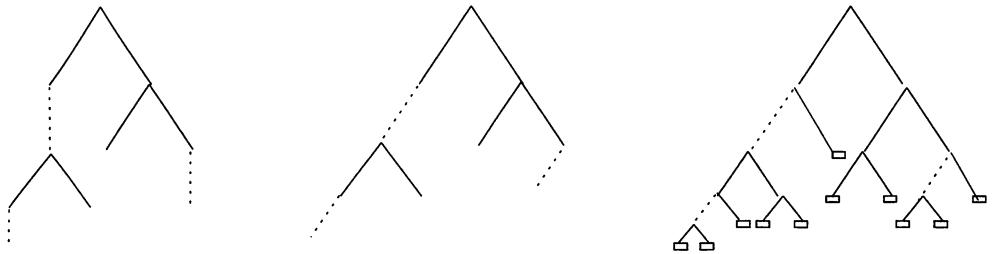


FIG. 2 – Un arbre 1-2, son code, et les deux arbres binaires associés.

Le membre gauche de l'identité (2) compte le nombre d'*abpc* de taille  $n$  pointés sur une feuille, ensemble que nous appelerons  $M_n^f$ , le premier terme du membre droit compte les *abpc* de taille  $n-1$  pointés sur un sommet quelconque (interne ou feuille), ensemble noté  $M_{n-1}^s$ , et le second terme est égal à 3 fois le nombre des *abpc* de taille  $n-2$  pointés sur un sommet interne, ensemble noté  $M_{n-2}^i$ .

Pour prouver de façon bijective la formule (2), il suffit de trouver une bijection entre les ensembles comptés par les deux membres, c'est-à-dire, en suivant la méthode de Rémy, une construction bijective permettant de construire un objet de  $M_n^f$  de façon non ambiguë, soit à partir d'un objet de  $M_{n-1}^s$ , et cela d'une façon, soit à partir d'un objet de  $M_{n-2}^i$ , et cela de trois façons.

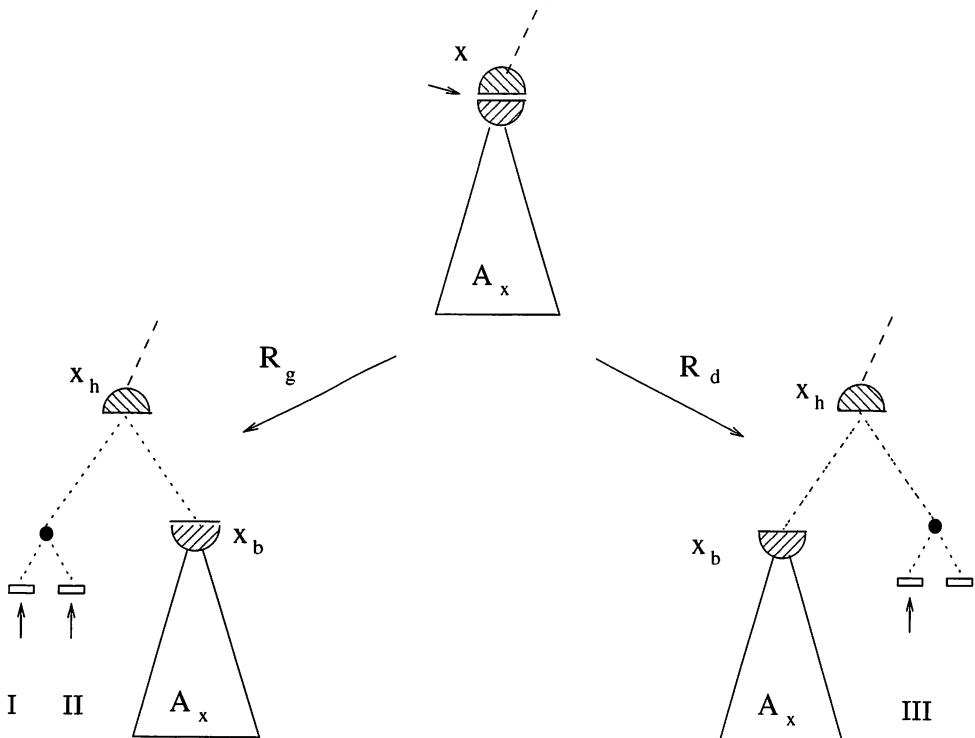


FIG. 3 – Les 3 cas de Super-Rémy.

### Les sept motifs de $M_n^f$

Dans les *abpc*, il ne peut y avoir, après complétion, de feuille isolée gauche, puisque par définition, les seuls fils uniques de l'arbre binaire penché sous jacent sont des fils à gauche. Parmi les *abpc* pointés sur une feuille, c'est-à-dire éléments de  $M_n^f$ , on distingue ainsi 7 sous-ensembles disjoints, selon que la feuille pointée est isolée, (cas noté *fid*, car c'est alors une feuille isolée droite), ou bien une feuille jumelle droite, et là nous distinguons 3 sous-cas selon que l'oncle (c'est-à-dire le sommet frère du père) est un fils droit feuille ou sommet interne, cas notés *fjdOdf*, et *fjdOdi* ou un fils gauche, nécessairement interne, cas noté *fjdOgi*, ou bien pour finir la feuille pointée est une feuille jumelle gauche, avec de façon identique 3 sous-cas *fjgOdf*, *fjgOdi* et *fjgOgi*.

Pour l'ensemble  $M_{n-1}^s$ , on distingue 4 sous-ensembles selon que le sommet pointé est un sommet interne, cas *si*, une feuille jumelle gauche ou droite, cas *fjg* et *fjd*, ou une feuille isolée droite, cas *fid*. Il n'y a qu'un seul cas pour l'ensemble  $M_{n-2}^i$ .

### Construction de $M_n^f$ à partir de $M_{n-1}^s$

On applique systématiquement la demi-construction de Rémy, appelée ci-dessus Rémy-gauche, sur un élément de  $M_{n-1}^s$ , c'est-à-dire un arbre pointé soit sur un sommet interne, soit une feuille de l'un des trois types *fjg*, *fid* et *fjd*, et l'on obtient un arbre de type respectivement *fid*, *fjdOdf*, *fjdOgi*; pour le 4<sup>ème</sup> cas, l'application directe de Rémy-gauche engendrerait le cas *fjdOgf*, qui est interdit dans cette famille d'arbres; on procède alors à une symétrie verticale donnant le motif *fjgOdf*, impossible à obtenir

directement par la construction.

### Construction de $M_n^f$ à partir de $M_{n-2}^i$

Pour augmenter de 2 le nombre de sommets internes, on utilise la construction de Rémy complète, c'est-à-dire bilatérale, mais on la modifie en ajoutant non pas une feuille mais un sous-arbre composé de 3 sommets, un sommet interne et ses deux fils qui constituent deux feuilles jumelles. En pointant sur l'arbre obtenu la feuille jumelle gauche ou droite ajoutée, on obtiendrait 4 cas différents. Cependant le cas où les deux feuilles jumelles ajoutées sont à droite, et où on pointe la plus à droite de ces deux feuilles redonne le même arbre que celui obtenu à partir du motif *fid* dans l'ensemble  $M_{n-1}^s$ . En supprimant cette possibilité redondante, on obtient exactement 3 arbres pointés distincts de  $M_n^f$  pour chaque arbre de  $M_{n-2}^i$ . La figure 3 montre les 3 arbres pointés que l'on peut obtenir par cette construction, appelée un peu cavalièrement Super-Rémy.

La preuve s'achève en vérifiant que les 7 arbres pointés obtenus en appliquant l'ensemble de ces deux constructions sont bien distincts. La figure 4 suivante récapitule les sept cas de cette double construction.

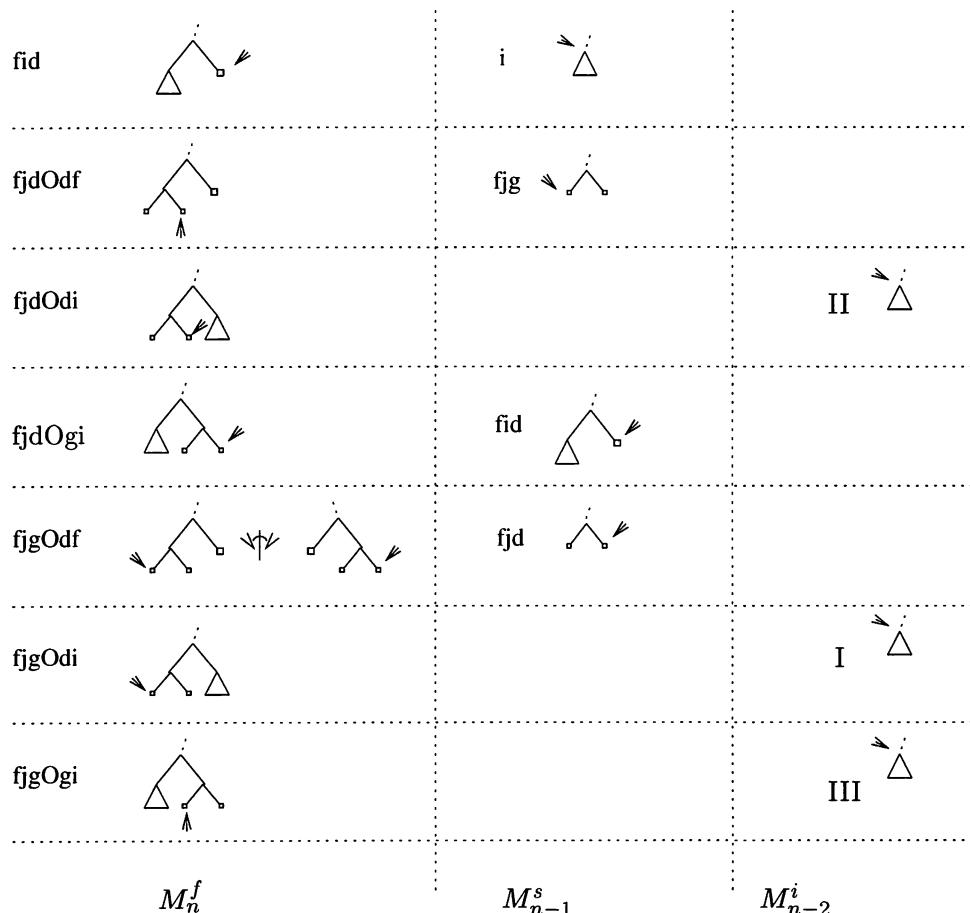


FIG. 4 – Les 7 cas de construction

## Références

- [1] Donaghey (R.) et Shapiro (L. W.). – Motzkin numbers. *Journal of Combinatorial Theory, A*, vol. 23, 1977, pp. 291–301.
- [2] Foata (D.) et Zeilberger (D.). – A classic proof of a recurrence for a very classical sequence. *J. Combin. Theory Ser. A*, vol. 80, n<sup>o</sup> 2, 1997, pp. 380–384.
- [3] Rémy (J.L.). – Un procédé itératif de dénombrement d'arbres binaires et son application à leur génération aléatoire. *R. A. I. R. O. Informatique Théorique*, vol. 19, n<sup>o</sup> 2, 1985, pp. 179–195.
- [4] Schröder (E.). – Vier combinatorische probleme. *Z.Math. Phys*, vol. 15, 1870, pp. 361–370.

# ÉNUMÉRATION DE TABLEAUX STANDARDS DE HAUTEUR BORNÉE

Francis Gascon

LACIM  
Université du Québec à Montréal

## Résumé

On sait que les séries génératrices des tableaux standards de hauteur bornée par  $h$  satisfont des équations différentielles avec des coefficients polynomiaux, voir [5] et [7]. On a montré que l'ordre de ces équations différentielles est  $\leq h/2 + 1$ , ce qui correspond aux P-réurrences conjecturées par Bergeron, Favreau et Krob pour les nombres de tableaux standards de hauteur bornée, voir [2] et [3]. On démontre ici, en utilisant des propriétés des fonctions de Bessel, que le degré et la forme de ces équations différentielles correspondent également à ces P-réurrences.

## Abstract

It is well known that the generating functions for Young tableaux of height bounded by  $h$  satisfy linear differential equations with polynomial coefficients, see [5] and [7]. We have proved, see [2], that the order of these differential equations is  $\leq h/2 + 1$ ; this corresponds to P-recurrences conjectured by Bergeron, Favreux and Krob for the number of bounded height tableaux see [3]. We show here, using properties of Bessel's functions, that the degree and the shape of the differential equations correspond also to these P-recurrences.

## 1. DÉFINITIONS ET RÉSULTATS

### 1.1 SÉRIES DE BESSEL

*Les séries de Bessel combinatoires* sont définies comme étant les solutions  $y = I_n(x)$  des équations différentielles suivantes :

$$x^2 y'' + x y' - (n^2 + 4x^2) y = 0$$

où  $n \geq 0$  est un entier. On a

$$I_n(x) = \sum_{k \geq 0} \frac{x^{2k+n}}{k! (k+n)!}. \quad (1.1)$$

Ces séries satisfont de nombreuses identités voir [1]. En particulier on a

$$I'_0(x) = 2 I_1(x), \quad (1.2)$$

$$I'_1(x) = 2 I_0(x) - \frac{1}{x} I_1(x), \quad (1.3)$$

$$x I_n(x) = (1-n) I_{n-1}(x) + x I_{n-2}(x), \quad (1.4)$$

$$I_i(x) I_j(x) = \sum_{l \geq 0} \binom{2l+i+j}{l} \binom{2l+i+j}{l+j} \frac{x^{2l+i+j}}{(2l+i+j)!}. \quad (1.5)$$

On remarque qu'en utilisant l'identité (1.4) on peut écrire les séries  $I_n$  en termes de  $I_0$  et  $I_1$ , par exemple

$$\begin{aligned} I_5(x) &= -6 \frac{(x^2 + 4) I_0(x)}{x^3} + \frac{(x^4 + 18x^2 + 24) I_1(x)}{x^4}, \\ I_6(x) &= \frac{(x^4 + 36x^2 + 120) I_0(x)}{x^4} - 3 \frac{(3x^4 + 32x^2 + 40) I_1(x)}{x^5}. \end{aligned}$$

En général on a

**Lemme 1.1.1.** Soit  $n \geq 0$  un entier, alors

$$I_n(x) = \frac{P_n(x)I_0(x)}{x^{n-2}} + \frac{Q_n(x)I_1(x)}{x^{n-1}},$$

où  $P_n$  et  $Q_n$  sont des polynômes à coefficients entiers satisfaisant la récurrence

$$a_n = x^2 a_{n-2} - (n-1) a_{n-1}$$

avec les conditions initiales  $P_0 = x^{-2}$ ,  $P_1 = 0$ ,  $Q_0 = 0$  et  $Q_1 = 1$ . ■

## 1.2 SÉRIES DIFFÉRENTIELLEMENT FINIES ET P-RÉCURSIVITÉ

On dit d'une série formelle  $f(x)$  quelle est *différentiellement finie* ou *D-finie* si elle satisfait une équation différentielle homogène avec coefficients polynomiaux. C'est-à-dire,

$$p_l(x)f^{(l)}(x) + p_{l-1}(x)f^{(l-1)}(x) + \cdots + p_0(x)f(x) = 0,$$

où  $f^{(i)}(x)$  est la dérivée d'ordre  $i$  de  $f$ .

De façon équivalente, la série  $f$  est D-finie si  $f$  et ses dérivées successives forment un espace vectoriel de dimension finie sur  $\mathbb{C}[x]$  (le corps des fractions rationnelles en  $x$ ).

Une équation différentielle est *d'ordre l* si  $l$  est l'ordre le plus élevé de dérivation de l'équation et qu'elle est de *degré n* si  $n$  est le degré le plus élevé de tous les polynômes apparaissant dans cette équation. Concernant l'ordre des équations différentielles on a

**Lemme 1.2.2.** Soit  $D_1$  et  $D_2$  des opérateurs différentiels linéaires d'ordre  $n$  et  $k$  respectivement. Soit  $y_1(x)$  et  $y_2(x)$  deux séries formelles telles que  $D_1(y_1(x)) = 0$  et  $D_2(y_2(x)) = 0$ , alors il existe des équations différentielles  $D^*$  et  $D^+$  d'ordre  $\leq n \cdot k$  et  $\leq n + k$  respectivement qui seront satisfaites par  $y_1(x)y_2(x)$  et  $y_1(x) + y_2(x)$  respectivement. ■

Soit  $u(n)$ , une fonction définie sur les entiers positifs ;  $u(n)$  est dite *P-récurse* si il existe des polynômes  $q_i(n)$  pour  $0 \leq i \leq m$  tels que

$$\sum_{i=0}^m q_i(n)u(n+i) = 0.$$

On remarque que ces récurrences peuvent être utilisées afin de calculer efficacement les nombres  $u(n+d)$ , voir [7]. Le théorème suivant fait le lien entre les concepts de séries D-finies et la P-récurse.

**Théorème 1.2.3.** Soit  $f = \sum_n f_n x^n$  une série formelle, alors on a que  $f$  est différentiellement finie si et seulement si  $f_n$  est  $P$ -récursive. ■

### 1.3 SÉRIES GÉNÉRATRICES

Un partage  $\mu$  de  $n$  est une suite d'entiers positifs

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_h,$$

telle que  $\sum_i \mu_i = n$ . On note  $l(\mu) = h$  le nombre de parts de  $\mu$ , on dit également que  $l(\mu)$  est la hauteur de  $\mu$ . Le diagramme de Ferrers d'un partage  $\mu$  est l'ensemble des points  $(i, j) \in \mathbb{Z}^2$  tel que  $1 \leq i \leq h$  et  $1 \leq j \leq \mu_i$ .

Un tableau  $t$  de forme  $\mu$  (un partage de  $n$ ) est une fonction qui associe pour chaque case du diagramme de Ferrers de  $\mu$  un élément de  $\{1, 2, \dots, n\}$ . On dit d'un tableau  $t$  qu'il est standard si les conditions suivantes sont vérifiées :

$$t(i, j) < t(i+1, j) \text{ et } t(i, j) < t(i, j+1).$$

Un tableau standard de forme  $\mu$  est de hauteur au plus  $h$  si  $l(\mu) \leq h$ . Par exemple, les tableaux standards sur 5 cases de hauteur au plus 2 sont

$$\boxed{1 \ 2 \ 3 \ 4 \ 5},$$

$$\begin{array}{c} \boxed{5} \\ \boxed{1 \ 2 \ 3 \ 4} \end{array}, \quad \begin{array}{c} \boxed{3} \\ \boxed{1 \ 2 \ 4 \ 5} \end{array}, \quad \begin{array}{c} \boxed{4} \\ \boxed{1 \ 2 \ 3 \ 5} \end{array}, \quad \begin{array}{c} \boxed{2} \\ \boxed{1 \ 3 \ 4 \ 5} \end{array},$$
  

$$\begin{array}{c} \boxed{4 \ 5} \\ \boxed{1 \ 2 \ 3} \end{array}, \quad \begin{array}{c} \boxed{2 \ 5} \\ \boxed{1 \ 3 \ 4} \end{array}, \quad \begin{array}{c} \boxed{3 \ 5} \\ \boxed{1 \ 2 \ 4} \end{array}, \quad \begin{array}{c} \boxed{2 \ 4} \\ \boxed{1 \ 3 \ 5} \end{array}, \quad \begin{array}{c} \boxed{3 \ 4} \\ \boxed{1 \ 2 \ 5} \end{array}.$$

Soit  $\tau_h(n)$  le nombre de tableaux standards de hauteur au plus  $h$  et  $\tau_h^{(2)}(n)$  le nombre de couples de tableaux standards de hauteur au plus  $h$ . Mentionnons, à propos de  $\tau_h^{(2)}$ , que la correspondance de Schensted [11] implique que  $\tau_h^{(2)}$  égal au nombre de permutations de  $S_n$  (le groupe symétrique sur  $n$  éléments) ayant des sous-suites décroissantes de longueur au plus  $h$ . À partir d'un résultat de Gordon [6], Gessel [5] a obtenu les formules suivantes :

$$y_h(x) = \sum_{n \geq 0} \tau_h(n) \frac{x^n}{n!} = \begin{cases} \det(I_{|i-j|}(x) + I_{i+j-1}(x))_{1 \leq i,j \leq h/2} & \text{si } h \text{ est pair} \\ e^x \det(I_{|i-j|}(x) - I_{i+j}(x))_{1 \leq i,j \leq (h+1)/2} & \text{si } h \text{ est impair,} \end{cases}$$

$$Y_h(x) = \sum_{n \geq 0} \tau_h^{(2)}(n) \frac{x^{2n}}{n!^2} = \det(I_{|i-j|}(x))_{1 \leq i,j \leq h},$$

où  $I_n(x) = \sum_{k \geq 0} \frac{x^{2k+n}}{k!(k+n)!}$ , est la série de Bessel combinatoire.

Concernant la classification de ces séries, il est clair, puisque des sommes et des produits de séries différentiellement finies sont différentiellement finies, qu'on a les résultats suivants :

**Proposition 1.3.4.** Les séries  $y_h(x)$  et  $Y_h(x)$  sont différentiellement finies. ■

**Corollaire 1.3.5.** Les fonctions définies par les nombres  $\tau_h(n)$  et  $\tau_h^{(2)}(n)$  sont P-récurrentes. ■

De plus, en utilisant les identités (1.4) et (1.5), il est possible d'extraire le coefficient de  $x^n$  de ces séries pour  $h \leq 5$ . On trouve, après simplification, des formules explicites, en particulier

$$\begin{aligned}\tau_4(n) &= C_{\lfloor(n+1)/2\rfloor} C_{\lceil(n+1)/2\rceil}, \\ \tau_5(n) &= 6 \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} C_i \frac{(2i+2)!}{(i+2)!(i+3)!},\end{aligned}$$

où  $C_i = \frac{1}{i+1} \binom{2i}{i}$  est le  $i$ -ième nombre de Catalan.

Pour ces deux identités, Gouyou-Beauchamps a donné des preuves bijectives, voir [8] et [9]. Pour  $h > 5$ , il est également possible de simplifier  $y_h(x)$  afin d'obtenir des formules, mais cela est moins automatique, voir [4]. Par exemple

$$\frac{\tau_6(2n)}{(2n)!} = 12 \sum_{k=0}^n \frac{g(n, k)(n-k+1)C_{k+1}}{(n-k+2)!^2(k+4)!k!},$$

où  $g(n, k) = -6k^2 - 11k + 4kn + 6n + 2$ . □

De la même façon pour les couples de tableaux, voir [10] et [12], on a

$$\begin{aligned}\tau_2^{(2)}(n) &= \frac{1}{n+1} \binom{2n}{n}, \\ \tau_3^{(2)}(n) &= 2 \sum_{k \geq 0} \binom{2k}{k} \binom{n}{k}^2 \frac{3k^2 + 2k + 1 - 2kn}{(k+1)^2(k+2)(n-k+1)}, \\ \frac{\tau_4^{(2)}(n)}{n!} &= \sum_{k \geq 0} \binom{n}{k} \frac{(n-k)!}{(2n-k)!} (-1)^k \tau_4(k) \tau_4(n-k), \\ &= \sum_{k \geq 0} \binom{n}{k} \frac{(n-k)!}{(2n-k)!} (-1)^k C_{\lfloor s/2 \rfloor} C_{\lceil s/2 \rceil} C_{\lfloor t/2 \rfloor} C_{\lceil t/2 \rceil},\end{aligned}$$

où  $s = k+1$  et  $t = n-k+1$ . □

#### 1.4 P-RÉCURRENCES LIÉES AUX NOMBRES $\tau_h(n)$ ET $\tau_h^{(2)}(n)$

Puisque les nombres  $\tau_h(n)$  sont P-récurrents, ils doivent satisfaire une récurrence de la forme

$$\sum_{k=0}^m p_k(n) \tau_h(n-k) = 0.$$

Bergeron, Favreau et Krob [3] ont énoncé des conjectures concernant le nombre de termes de cette récurrence ainsi que la forme des polynômes  $p_k(n)$  à savoir

$$m = \lfloor h/2 \rfloor + 1, \tag{1.6}$$

$$\deg(p_k(n)) \leq h/2, \tag{1.7}$$

$$p_k(n) = q_k(n) \prod_{i=1}^{k-1} (n-i), \quad 1 \leq k \leq m \quad (1.8)$$

$$p_0(n) = \prod_{k=1}^{\lfloor h/2 \rfloor} (n + k(h-k)), \quad (1.9)$$

où les  $q_k(n)$  sont des polynômes de degré au plus  $\lfloor h/2 \rfloor - k + 1$ .

Pour  $h = 2m + 1$ , il y a deux conjectures supplémentaires

$$\deg(q_k(n)) = m - k + 1, \quad (1.10)$$

$$- p_1(n) = n p_0(n) - (n-1) p_0(n-1). \quad (1.11)$$

On peut pour  $h \leq 20$  les calculer explicitement, voir [2] et [3]. Par exemple

$$(n+1)\tau_2(n) - 2\tau_2(n-1) - 4(n-1)\tau_2(n-2) = 0,$$

$$(n+2)\tau_3(n) - (2n+1)\tau_3(n-1) - 3(n-1)\tau_3(n-2) = 0,$$

$$(n+3)(n+4)\tau_4(n) - 4(2n+3)\tau_4(n-1) - 16(n-1)n\tau_4(n-2) = 0,$$

$$(n+4)(n+6)\tau_5(n) - (3n^2 + 17n + 15)\tau_5(n-1) - (n-1)$$

$$(13n+9)\tau_5(n-2) + 15(n-1)(n-2)\tau_5(n-3) = 0.$$

□

Pour les couples de tableaux, il y a des conjectures similaires voir[3]. Dans ce qui suit, on appellera ces conjectures *les conjectures BFK*. Ces récurrences sont aisément vérifiées pour  $h \leq 12$ , voir [2].

Afin de démontrer les conjectures BFK pour un  $h$  quelconque, on doit d'abord simplifier les séries  $y_h(x)$  et  $Y_h(x)$ . On utilise pour ce faire une idée de Gessel [5] et le lemme (1.1.1). Par exemple, pour  $h = 6$ ,

$$\begin{aligned} y_6(x) &= \det \begin{pmatrix} I_0(x) + I_1(x) & I_1(x) + I_2(x) & I_2(x) + I_3(x) \\ I_1(x) + I_2(x) & I_0(x) + I_3(x) & I_1(x) + I_4(x) \\ I_2(x) + I_3(x) & I_1(x) + I_4(x) & I_0(x) + I_5(x) \end{pmatrix} \\ &= 2I_2(x)I_3(x)I_1(x) + I_1(x)I_3(x)I_5(x) + 2I_2(x)I_3(x)I_4(x) \\ &\quad - I_3(x)^2I_0(x) - 2I_0(x)I_1(x)^2 + I_1(x)I_0(x)^2 + I_3(x)I_0(x)^2 \\ &\quad - I_0(x)I_4(x)^2 + I_0(x)^3 - J_1(x)^3 - 2I_2(x)^2I_0(x) \\ &\quad + I_1(x)I_3(x)I_0(x) - 2J_1(x)I_2(x)I_0(x) - \\ &\quad 2I_1(x)I_2(x)I_5(x) + 2I_1(x)I_2(x)I_4(x) + 2I_1(x)I_3(x)I_4(x) \\ &\quad + I_1(x)I_0(x)I_5(x) - I_2(x)^2I_3(x) - I_3(x)^3 \\ &\quad - 2I_2(x)I_3(x)I_0(x) + I_0(x)I_3(x)I_5(x) - 2I_0(x)I_1(x)I_4(x) - \\ &\quad I_2(x)^2I_5(x) + 2I_2(x)^2I_1(x) \\ &\quad + 2I_2(x)^2I_4(x) + I_0(x)^2I_5(x) - 2I_1(x)^2I_4(x) - I_1(x)I_4(x)^2 \\ &\quad - I_1(x)^2I_5(x) + 2I_2(x)J_1(x)^2 + 2I_3(x)I_1(x)^2 - 2I_2(x)I_3(x)^2, \end{aligned}$$

après simplification, on obtient

$$\begin{aligned} y_6(x) &= -4 \frac{(4x-3)I_0(x)^3}{x^4} - 4 \frac{(4x^2-3x+6)I_1(x)I_0(x)^2}{x^5} \\ &\quad + 4 \frac{(4x^3-x^2+3)I_1(x)^2I_0(x)}{x^6} + 4 \frac{(4x^3-x^2+5x+1)I_1(x)^3}{x^6}. \end{aligned}$$

En général, on a

**Lemme 1.4.6.** Soit  $h = 2g$ , alors

$$\begin{aligned} y_h(x) &= \det(I_{|i-j|(x)} + I_{i+j-1}(x))_{1 \leq i,j \leq g} \\ &= \sum_{k \geq 0}^g P_k(x) \frac{I_0^k(x) I_1^{g-k}(x)}{x^{m_k}}, \end{aligned} \quad (1.12)$$

où

1. Le degré de  $P_k(x)$  pour  $k \neq 0$  est donné par

$$n_k = \begin{cases} \frac{1}{2}g(g-1) - (k-1) & \text{si } g \text{ est impair} \\ \frac{1}{2}g(g-1) - (k-1) - \frac{1}{2}(1 + (-1)^{k+1}) & \text{sinon.} \end{cases}$$

Pour  $k = 0$ , on a

$$n_0 = \frac{1}{2}g(g-1).$$

2.  $P_k(x)$  est un polynôme à coefficients entiers tel que  $P_k(0) \neq 0$ .

3.  $m_k$  est donnée par

$$m_k = \begin{cases} g(g-1) - (k-1) & \text{si } k \neq 0 \\ g(g-1) & \text{sinon.} \end{cases}$$

Pour le cas impair ainsi que pour les couples de tableaux, on a des résultats semblables, voir [4]. ■

## 2. ORDRE ET DEGRÉ DES ÉQUATIONS DIFFÉRENTIELLES

### 2.1 ORDRE ET DEGRÉ DE L'ÉQUATION DIFFÉRENTIELLE SATISFAITE PAR $I_0^k I_1^n$

Dans un premier temps, on remarque que le problème consistant à construire des récurrences qui seront satisfaites par les coefficients des séries  $I_0^k I_1^n$  peut être résolu en utilisant les bornes théoriques pour l'ordre (lemme 1.2.2). Par exemple, à partir des équations différentielles satisfaites par  $I_0(x)$  et  $I_1(x)$ , on construit

$$x^4 y^{(4)} + 6x^3 y''' + (-16x^4 + 5x^2) y'' + (-64x^3 - x) y' + (-32x^2 + 1) y = 0,$$

qui est satisfait par  $I_0 I_1$ . Mais on peut faire mieux. On a

**Lemme 2.1.1.** La série  $I_0^k I_1^n$  satisfait une équation différentielle d'ordre  $\leq k+n+1$ . ■

Pour ce qui est du degré, on a la conjecture suivante :

**Conjecture.** Les séries  $I_0^k I_1^n$  avec  $k \neq 0$  et  $n \neq 0$  satisfont des équations différentielles de degré  $\leq k+n+1$ . □

En particulier, pour  $I_0 I_1$  on trouve

$$x^3 y''' + 4x^2 y'' + (-16x^3 + x) y' + (-32x^2 - 1) y = 0.$$

On peut également calculer des récurrences pour les coefficients des séries  $I_0^k I_1^n$ . Pour ce faire, on définit les nombres  $\gamma_k(2j+n)$  :

$$\sum_{j \geq 0} \gamma_k(2j+n) x^{2j+n} := I_0^k(x) I_1^n(x).$$

Par exemple, les coefficients  $\gamma_1(2j+1)$  de la série  $I_0 I_1$  satisfont la P-référence suivante :

$$(-16n - 32)\gamma_1(n) + (n^3 + 7n^2 + 15n + 9)\gamma_1(n+2) = 0,$$

où  $\gamma_1(0) = 0$  et  $n = 2j+1$ .

## 2.2 ORDRE ET DEGRÉ DE L'ÉQUATION DIFFÉRENTIELLE SATISFAITE PAR $y_h(x)$

On sait qu'il existe une équation différentielle satisfaite par  $y_h(x)$  d'ordre au plus  $(\lfloor h/2 \rfloor)! 2^h$ , mais comme précédemment on peut faire mieux.

**Proposition 2.2.2.** Soit  $V_h$  l'espace vectoriel, sur  $\mathbb{C}(x)$ , engendré par les dérivées de tout ordre de  $y_h(x)$ , alors  $\dim V_h \leq \lfloor h/2 \rfloor + 1$ . ■

**Corollaire 2.2.3.** Pour tout nombre entier  $h > 0$ , il existe une équation différentielle d'ordre plus petit ou égal à  $\lfloor h/2 \rfloor + 1$  satisfaite par la série  $y_h(x)$ . ■

Pour  $Y_h(x)$ , on a un résultat similaire, voir [2].

On sait maintenant que  $y_h(x)$  doit satisfaire une équation de la forme

$$Q_0(x)y + Q_1(x)y' + Q_2(x)y'' + \dots + Q_l y^{(l)} = 0,$$

où  $l = \lfloor h/2 \rfloor + 1$  et  $Q_i$  est un polynôme à coefficients entiers.

Soit  $\theta_h^{(K)}$  un opérateur différentiel défini de la façon suivante :

$$\theta_h^{(K)} = \sum_{i=0}^l \left( \sum_{j=0}^K q_{i,j} x^j \right) D^i$$

où  $l = \lfloor h/2 \rfloor + 1$ .

Concernant cet opérateur, on a le résultat suivant :

**Lemme 2.2.4.** Pour  $h$  un entier pair, lorsqu'on applique  $\theta_h^{(K)}$  à  $y_h(x)$  on obtient une expression de la forme

$$\theta_h^{(K)}(y_h(x)) = R(h, K) = \sum_{k=0}^{h/2} \sum_{i=-h/2}^{m_0/2+K+1} c_{k,i}^{(K,h)} \frac{I_0(x)^k I_1(x)^{h/2-k}}{x^{m_k-i+1}},$$

où  $c_{k,i}^{(K,h)}$  sont des combinaisons linéaires en les variables  $q_{ij}$ . ■

On montrera que pour tout  $K \geq h/2$ , le système  $c_{k,i}^{(K,h)} = 0$  admet des solutions non-triviales.

Afin d'obtenir des matrices associées aux coefficients des systèmes  $c_{k,i}^{(K,h)} = 0$  les plus simples possibles (blocs diagonaux), on ordonne les variables  $q_{i,j}$  selon l'ordre croissant de la différence  $j - i$  et selon l'ordre croissant du premier indice pour un  $j - i$  fixé. On ordonne également les lignes de la façon suivante : la  $j$ -ième ligne correspond aux coefficients de l'équation

$$\left[ \frac{I_0^a I_1^{h/2-a}}{x^m} \right] (R(h, K)) = 0,$$

où  $a = j + \nu - 1 - (h/2 + 1) \left\lfloor \frac{j+\nu-1}{(h/2+1)} \right\rfloor$  et  $m = ((h/2 + 2) \left\lfloor \frac{j+\nu-1}{(h/2+1)} \right\rfloor + h^2/4 - (j + \nu - 1) - 2) K$

avec  $\nu = ((1 + (K - h/2)) - h/4 (h/2 - 1)) (h/2 + 1)$ .

Par exemple, pour  $h = 4$  et  $K = 3$ , si on considère seulement les lignes telles que  $4 \leq j \leq 18$ , on a la matrice suivante :

$$C_4^{(3)} = \begin{pmatrix} 1 & 2 & -2 & 0 & 0 & 8 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 0 & 16 & 0 & 0 & 0 & 64 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 2 & -2 & -4 & 0 & 0 & 8 & -88 & 0 & 0 & 32 \\ 0 & 2 & -6 & 0 & 4 & -28 & 0 & 0 & 16 & -160 & 0 & 0 & 64 \\ 0 & -2 & 6 & 0 & 0 & 8 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -4 & 20 & 2 & -2 & -4 & 68 & 0 & 8 & -88 \\ 0 & 0 & 0 & 2 & -6 & 24 & 0 & 4 & -28 & 192 & 0 & 16 & -160 \\ 0 & 0 & 0 & -2 & 6 & -24 & 0 & 0 & 8 & -72 & 0 & 0 & 32 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 20 & -120 & -2 & -4 & 68 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -6 & 24 & -120 & 4 & -28 & 192 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 6 & -24 & 120 & 0 & 8 & -72 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 20 & -120 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 24 & -120 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & -24 & 120 \end{pmatrix}$$

Après l'étude de ces matrices, on conclut que

**Lemme 2.2.5.** Il existe un entier  $K$  tel que le système  $c_{k,j}^{(K,h)} = 0$  admet des solutions non-triviales.

**Lemme 2.2.6.** Le système  $c_{k,j}^{(h/2)} = 0$  admet également des solutions non-triviales.

**Corollaire 2.2.7.** Pour un entier  $g \geq 1$ , la série génératrice

$$y_{2g}(x) = \sum_{n \geq 0} \tau_{2g}(n) \frac{x^n}{n!},$$

satisfait une équation différentielle d'ordre  $g + 1$  et de degré  $g$  de la forme

$$\sum_{i=0}^l x^{i-1} Q_i(x) y^{(i)} = 0, \quad (2.1)$$

où  $l = \lfloor h/2 \rfloor + 1$  et  $Q_i$  est le polynôme à coefficients entiers tel que

$$\deg(Q_i) = \begin{cases} \lfloor h/2 \rfloor - i + 1 & \text{si } \lfloor h/2 \rfloor - i + 1 \text{ est pair} \\ \lfloor h/2 \rfloor - i & \text{sinon.} \end{cases}$$

■

**Corollaire 2.2.8.** Pour  $g \geq 1$ , les nombres  $\tau_{2g}(n)$  satisfont des P-réurrences de la forme

$$\sum_{k=0}^m p_k(n) \tau_{2g}(n-k) = 0.$$

où  $m = \lfloor h/2 \rfloor + 1$ ,  $\deg(p_k(n)) \leq h/2$  et telles que

$$p_0(n) = \prod_{k=1}^{\lfloor h/2 \rfloor} (n+k(h-k))$$

$$p_k(n) = q_k(n) \prod_{i=1}^{k-1} (n-i),$$

où les  $q_k(n)$  sont des polynômes de degré au plus  $\lfloor h/2 \rfloor - k + 1$ . ■

Ce qui démontre les conjectures BFK dans le cas pair. Dans le cas impair ainsi que pour les couples de tableaux, on a des résultats semblables, voir [4].

## RÉFÉRENCES

- [1] M. Abramowitz et I.A. Stegun, *Handbook of Mathematical Functions*, Dover publications, New York, (1972) 355-436.
- [2] F. Bergeron et F. Gascon, *Counting Young Tableaux of Bounded Height*, Jour. of Integer Sequences, à paraître.
- [3] F. Bergeron, L. Favreau et D. Krob, *Conjectures on the Enumeration of Tableaux of bounded height*, Discrete Math, 139 (1995), 257-285.
- [4] F. Gascon, *Tableaux de Hauteur Bornée et Séries de Bessel*, thèse Ph. D., en préparation.
- [5] I.M. Gessel, *Symmetric Functions and P-recursiveness*, J. Comb. Th. Series A, 53, (1990), 257-285.
- [6] B. Gordon, *Notes on Plane Partitions*, J. Comb. Th. 11 (1971), 157-168.
- [7] R.P. Stanley, *Differentiably Finite Power Series*, European J. Comb. (1980), 175-188.
- [8] D. Gouyou-Beauchamps, *Codages par des mots et des chemins : problèmes combinatoires et algorithmes*, thèse Ph. D., Université de Bordeaux I, (1985).
- [9] D. Gouyou-Beauchamps, *Standard Young Tableau of Height 4 and 5*, European J. Comb. 10 (1989), 62-82.
- [10] A. Rogers, *Ascending Sequences in Permutations*, Discrete Math. 22 (1978), 35-40.
- [11] C. Schensted, *Longest Increasing and Decreasing Subsequences*, Canad. J. Math. 13 (1961), 179-191.
- [12] H.S. Wilf *Ascending Subsequences of Permutations and the Shapes of Tableaux* J. Comb. Th. Series A, (1992), 155-157.

# Symmetries in trees and parking functions

Louis H. Kalikow  
 College of the Holy Cross  
 Department of Mathematics and Computer Science  
 Worcester, MA 01610  
[kalikow@mathcs.holycross.edu](mailto:kalikow@mathcs.holycross.edu)

## Abstract

We investigate a particular symmetry in labeled trees discovered first by Gessel, which can be stated as follows: In the set of rooted labeled trees on  $n + 1$  vertices rooted at the smallest vertex, the number of trees with  $i$  descents and  $j + 1$  leaves equals the number of trees with  $j$  descents and  $i + 1$  leaves. We present two new ways to prove the symmetry resulting from decompositions of trees, which lead to three different bijections from trees to trees in which leaves and descents are swapped. We also interpret the symmetry in terms of parking functions: the number of parking functions on  $[n]$  with  $i$  descents and  $j$  unfavorable spaces (defined in this paper) equals the number of parking functions on  $[n]$  with  $j$  descents and  $i$  unfavorable spaces.

## Résumé

Nous étudions une symétrie particulière dans les arbres étiquetés, découverte par Gessel, qu'on peut énoncer comme suit: Dans l'ensemble des arbres étiquetés pointés avec  $n + 1$  sommets, pointés au sommet minimum, le nombre d'arbres avec  $i$  descentes et  $j + 1$  sommets pendants égale le nombre d'arbres avec  $j$  descentes et  $i + 1$  sommets pendants. Nous présentons deux nouvelles démonstrations de la symétrie, qui résultent des décompositions des arbres; à partir des décompositions, nous obtenons trois bijections des arbres sur les arbres qui échangent les sommets pendants et les descentes. De plus, nous interprétons la symétrie en termes des “fonctions de stationnement” (parking functions): le nombre des fonctions de stationnement avec  $i$  descentes et  $j$  positions défavorables (définies dans cette article) égale le nombre de fonctions de stationnement avec  $i$  positions défavorable et  $j$  descentes.

## 1 Introduction

A *labeled tree* is a tree whose vertices are labeled with elements of some set. We shall consider trees labeled by nonnegative integers. In particular, if  $i$  is an integer and  $S$  a finite set of integers for which  $i \notin S$ , then  $\mathcal{T}_{i,S}$  will denote the set of rooted labeled trees on  $|S| + 1$  vertices with root labeled  $i$  and other vertices labeled by the elements of  $S$  (with each label used exactly once). In a rooted labeled tree, the *parent* of  $v$  is first vertex after  $v$  on the unique path from  $v$  to the root;  $v$  is a *child* of the vertex  $w$  if  $w$  is the parent of  $v$ . Two vertices  $w$  and  $v$  are *siblings* if they have the same parent.

We will consider two statistics. A *leaf* of a tree is a vertex with no children. A *descent* of a labeled tree is a vertex whose label is greater than at least one of its children's labels (see [4], [6]). We will be considering two sets of trees,  $\mathcal{T}_{0,[n]}$  and  $\mathcal{T}_{n,[n-1]}$  (where we use  $[n] := \{1, 2, \dots, n\}$ ).

We will first focus on the fact, discovered first by Gessel [4], that the number of trees in  $\mathcal{T}_{0,[n]}$  with  $i$  descents and  $j + 1$  leaves equals the number of trees in  $\mathcal{T}_{0,[n]}$  with  $j$

descents and  $i+1$  leaves. Gessel's proof used marked forests, instead of directly counting the trees. For our proof, we use decompositions of trees. The decompositions lead to a generalization of a well-known bijection from permutations of  $[n]$  to increasing trees in  $\mathcal{T}_{0,[n]}$  (i.e., trees with no descents). They also lead to a generalization of an identity found by Knuth [10] involving *intransitive* trees, which are labeled trees in which all the vertices adjacent to  $i$  (i.e., the parent and the children of  $i$ ) are either less than  $i$  or greater than  $i$  (see [16, p. 90]).

It turns out the first decomposition for trees is related to a symmetry in parking functions. A *parking function on  $[n]$*  is a function  $p : [n] \rightarrow [n]$  such that for all  $j \in [n]$ ,  $|\{i \in [n] \mid p(i) \leq j\}| \geq j$ . Parking functions were first considered by Konheim and Weiss in 1966 [11]. Their name comes from the following colorful (and equivalent) description: Consider  $n$  cars numbered 1 to  $n$  driving down a one-way street containing  $n$  spaces numbered 1 to  $n$ . Each car  $i$  has a preferred space  $p(i)$  it wishes to park in. In increasing order of  $i$ , the cars try to park. When car  $i$  goes to park, it goes straight to space  $p(i)$ ; if that space is taken, the car proceeds forward looking for the next empty space. If no space is empty, it drives away. The parking functions are the preference functions  $p : [n] \rightarrow [n]$  such that all cars park using the above algorithm. For more on parking functions, see [1, 2, 7, 12] (the last deals with *suites majeures*, which are equivalent to parking functions).

For a parking function  $p$  on  $[n]$ , the permutation of parked cars resulting from running the parking algorithm is called the *output permutation* of the parking function. We then define a *descent* of a parking function to be a descent of the output permutation. Next, for a parking function  $p \in P_n$ , let  $p([n])$  denote the range of the function. We call an element of  $[n] - p([n])$  an *unfavorable space*. In other words, an unfavorable space is a space in which no car prefers to park. For example, for the parking function  $p(1) = 2$ ,  $p(2) = 4$ ,  $p(3) = 1$ ,  $p(4) = 3$ , the output permutation is 2413, and  $p$  has 2 unfavorable spaces and 1 descent (we do not consider the last element to be a descent). We will show that number of parking functions on  $[n]$  with  $i$  descents and  $j$  unfavorable spaces equals the number of parking functions on  $[n]$  with  $j$  descents and  $i$  unfavorable spaces.

The following important definitions follow Gessel [3]. Given any object with  $n$  different labels taken from an arbitrary totally ordered finite set  $S = \{x_1 < x_2 < \dots < x_n\}$ , we can define the *content* of the object to be the set  $S$ . We define the *reduction* of the object *with respect to a set of  $n$  consecutive integers*  $\{a, a+1, \dots, a+(n-1)\}$  to be the object obtained by replacing the label  $x_i$  with the label  $a+(i-1)$ . We note that a tree with labels in a totally ordered set is determined by its content and its reduction with respect to a given set of consecutive integers. For example, Figure 1 shows a tree  $T$  with content  $\{2, 4, 7, 8\}$  and its reduction with respect to the set  $[4] = \{1, 2, 3, 4\}$ . In this work, we will be chiefly concerned with reductions with respect to  $[n]$  and  $[n] \cup \{0\}$ .



$T$       reduction of  $T$  with respect to  $[4]$

Figure 1: A tree on 4 vertices and its reduction with respect to the set  $[4]$

Unless otherwise specified, the reduction of a rooted labeled tree in  $\mathcal{T}_{0,[n]}$  will be taken with respect to  $[n] \cup \{0\}$ . In addition, we can generalize parking functions to functions  $p : A \rightarrow [n]$  where  $A$  is a totally ordered set of  $n$  elements. We can then take the reduction of such a function to obtain a parking function on  $[n]$ . The reduction of a parking function on a set  $A$  with  $n$  elements will always be taken with respect to  $[n]$ .

## 2 Descents and leaves in trees: First decomposition

We now define two sequences of polynomials. First, let  $u_n(\alpha, \beta)$  be the polynomial in which the coefficient of  $\alpha^i \beta^j$  is the number of trees in  $\mathcal{T}_{0,[n]}$  with  $i$  descents and  $j+1$  leaves. Next, let  $t_n(\alpha, \beta)$  be the polynomial in which the coefficient of  $\alpha^i \beta^j$  is the number of trees in  $\mathcal{T}_{n,[n-1]}$  with  $i$  descents and  $j$  leaves. It is not hard to see that by these definitions,  $\alpha \beta u_n = t_{n+1}$ .

We first prove the following theorem.

**Theorem 1** For all  $n \geq 0$ ,  $u_n(\alpha, \beta)$  is symmetric in  $\alpha$  and  $\beta$ .

This result was first shown by Gessel [4], who obtained a functional equation for the generating function  $U(x; \alpha, \beta) = \sum_{n=1}^{\infty} u_n(\alpha, \beta) \frac{x^n}{n!}$ ,

$$1 + U = (1 + \alpha U)(1 + \beta U)e^{x(1-\alpha-\beta-\alpha\beta U)}. \quad (1)$$

Gessel obtained this equation by considering marked forests, instead of directly counting the trees. In this section and the next, we present two new proofs of Theorem 1, which use various decompositions of labeled trees to obtain recurrences for the  $u_n$ .

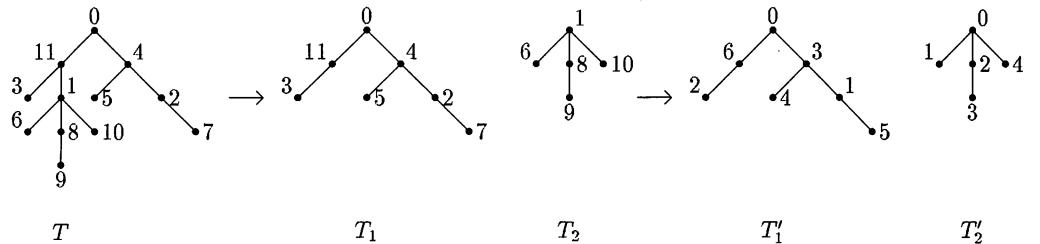


Figure 2: First decomposition of trees.

To show  $u_n$  is symmetric, we obtain a recurrence for  $u_n$  in terms of  $u_i$ ,  $1 < i < n-1$ . We use a decomposition that involves forming two trees from a tree in  $\mathcal{T}_{0,[n]}$  by deleting the edge  $e$  connecting 1 to its parent. The decomposition works as follows. Given a tree  $T$ , let  $T - e$  be the graph obtained by removing the edge  $e$  from  $T$ . Then let  $T_1$  be the component of  $T - e$  containing the root of  $T$  (labeled 0), and let  $T_2$  be the component of  $T - e$  containing the vertex labeled 1. We can then take the reduction of the two trees  $T_1$  and  $T_2$ . Figure 2 shows a tree  $T$ , the trees  $T_1$  and  $T_2$ , and the decomposition of  $T$  into reduced trees,  $T'_1$  and  $T'_2$ .

We can see that two reduced trees  $T'_1$ ,  $T'_2$ , the content of  $T_1$ , and the parent of 1 in  $T$  specify  $T$ . In Figure 2, the quadruple  $(T'_1, T'_2, \{0, 2, 3, 4, 5, 7, 11\}, 11)$  specifies  $T$ . Note that vertex 11, the parent of vertex 1 in  $T$ , comes from a vertex in  $T'_1$  which is a descent. We describe this by saying that the parent of 1 in  $T$  is a descent in  $T'_1$ .

To prove Theorem 1, we use the following lemma. In the statement of the lemma, the arguments of the polynomials  $u_n(\alpha, \beta)$  are suppressed.

**Lemma 1** If  $n \geq 1$ , then

$$\begin{aligned} u_n = u_{n-1} + \sum_{i=1}^{n-1} \binom{n-1}{i} u_{n-1-i} & \left( \beta \frac{\partial}{\partial \alpha} \alpha u_i + \alpha \frac{\partial}{\partial \beta} \beta u_i \right. \\ & \left. + \alpha \beta ((i+1)u_i - \frac{\partial}{\partial \alpha} \alpha u_i - \frac{\partial}{\partial \beta} \beta u_i) \right). \end{aligned} \quad (2)$$

Note that this recurrence is symmetric in  $\alpha$  and  $\beta$ , as are the initial cases are  $u_0(\alpha, \beta) = u_1(\alpha, \beta) = 1$ . So if (2) holds, then the  $u_n$  must all be symmetric polynomials. Hence, once Lemma 2 is proven, Theorem 1 follows immediately.

**Proof.** Let  $T'_1$  be a tree in  $\mathcal{T}_{0,[i]}$  and  $T'_2$  a tree in  $\mathcal{T}_{0,[n-i-1]}$ . By considering four cases of how we can obtain a tree  $T \in \mathcal{T}_{0,[n]}$  which decomposes into  $T_1$  and  $T_2$ , we can obtain a recurrence for  $u_n$ . In each case, to build  $T$  from smaller trees, we first choose an  $i$ -element subset  $A$  of  $\{2, 3, 4, \dots, n\}$  and form  $T_1$  with reduction  $T'_1$  and content  $A \cup \{0\}$ . Then we form  $T_2$  with content  $[n] - A$  and reduction  $T'_2$ ; note that  $T_2$  will be rooted at 1. To form a tree  $T \in \mathcal{T}_{0,[n]}$ , we can choose any vertex  $v$  of  $T_1$  and make the root of  $T_2$  a child  $v$ . The number of descents and leaves of  $T$  depend upon the number of descents and leaves of  $T_1$  and  $T_2$  as well as the vertex of  $T_1$  chosen to be the parent of the root of  $T_2$ .

In each case below, for  $j = 1$  or  $2$ ,  $a_j$  denotes the number of descents of  $T'_j$  (as well as of  $T_j$ ) and  $b_j$  denotes the number of leaves of  $T'_j$  (as well as of  $T_j$ ). Since any tree  $T'_1 \in \mathcal{T}_{0,[i]}$  has  $i$  non-root vertices, we must choose  $i$  labels from  $[n] - \{1\}$  to be the content of  $T_1$ ; clearly, this choice can be made in  $\binom{n-1}{i}$  ways. We also assume  $n \geq 1$ .

*Case i:*  $i = 0$ . In this case,  $T_1$  is a tree consisting only of a root node labeled 0.

When we attach the root of  $T_2$  (labeled 1) to 0, we get a tree on  $n+1$  vertices with  $a_2$  descents and  $b_2$  leaves. These trees are simply counted by  $u_{n-1}(\alpha, \beta)$ .

*Case ii:*  $i > 0$  and we attach the root of  $T_2$  to a descent of  $T_1$ , or to the root of  $T_1$ .

Here, the number of descents of the new tree will simply be  $a_1 + a_2$ , and the number of leaves will be  $b_1 + b_2$ . Since there are  $b_1$  descents and one root in  $T'_1$ , the contribution of this case to  $u_n$  is

$$\binom{n-1}{i} u_{n-1-i}(\alpha, \beta) \beta \left( \frac{\partial}{\partial \alpha} \alpha u_i(\alpha, \beta) \right).$$

We need the extra factor of  $\beta$  since the coefficient of  $\beta^j$  in  $u_i(\alpha, \beta)$  counts trees with  $j+1$  leaves.

*Case iii:*  $i > 0$  and we attach the root of  $T_2$  to a leaf of  $T_1$ . In this case, the number of descents of the new tree will be  $a_1 + a_2 + 1$ , since the parent of 1 in  $T$  is a leaf in  $T_1$  and hence not a descent in  $T_1$ . The number of leaves of the new tree will be  $b_1 + b_2 - 1$ . Recalling that the coefficient of  $\beta^j$  in  $u_i(\alpha, \beta)$  counts trees with  $j+1$  leaves, the contribution to  $u_n(\alpha, \beta)$  is

$$\binom{n-1}{i} u_{n-1-i}(\alpha, \beta) \alpha \left( \frac{\partial}{\partial \beta} \beta u_i(\alpha, \beta) \right).$$

*Case iv:*  $i > 0$  and we attach the root of  $T_2$  to a vertex of  $T_1$  which is neither a leaf nor a descent nor the root. The new tree in this case will have  $a_1 + a_2 + 1$  descents and  $b_1 + b_2$  leaves, so that the contribution to  $u_n(\alpha, \beta)$  is

$$\binom{n-1}{i} \alpha \beta u_{n-1-i}(\alpha, \beta) \left( (i+1)u_i(\alpha, \beta) - \frac{\partial}{\partial \alpha} \alpha u_i(\alpha, \beta) - \frac{\partial}{\partial \beta} \beta u_i(\alpha, \beta) \right).$$

We add up the contribution for each  $i$  from 0 to  $n-1$  to get the recurrence stated in the lemma.  $\square$

Note that in  $U(x; \alpha, \beta) = \sum_{n=1}^{\infty} u_n(\alpha, \beta) \frac{x^n}{n!}$ , we do not include the constant term  $u_0 = 1$ . If we wish, we can rewrite (2) without  $u_0$  by simply replacing it with 1. By multiplying both sides of (2) by  $x^{n-1}/(n-1)!$  and summing both sides from 1 to  $\infty$ , we can obtain the differential equation (where all arguments have been suppressed)

$$\begin{aligned} \left( \frac{1}{U+1} \right) \frac{\partial}{\partial x} U &= \beta \frac{\partial}{\partial \alpha} (\alpha U) + \alpha \frac{\partial}{\partial \beta} (\beta U) + \alpha \beta \frac{\partial}{\partial x} (xU) \\ &\quad - \alpha \beta \frac{\partial}{\partial \alpha} (\alpha U) - \alpha \beta \frac{\partial}{\partial \beta} (\beta U) + 1. \end{aligned} \tag{3}$$

We will later use this differential equation to obtain Gessel's functional equation for  $U$ .

From the recurrence, it is easy to describe an involution  $\Pi_n$  on  $\mathcal{T}_{0,[n]}$  which sends a tree with  $i$  descents and  $j+1$  leaves to a tree with  $j$  descents and  $i+1$  leaves. We define the involution recursively.

It is trivial to describe  $\Pi_0$  and  $\Pi_1$ . We now assume  $\Pi_i$  has been defined for  $i < n$ . For simplicity, if the content of a tree  $T$  is some totally ordered set and  $T$  is rooted at its smallest vertex, we let  $\Pi_n(T)$  be the tree whose reduction is  $\Pi_n(T')$  and whose content is the content of  $T$ . By an *increasing* vertex, we shall mean a non-root vertex which is neither a descent nor a leaf (that is, an increasing vertex is a non-leaf vertex all of whose children are greater than it).

Consider a tree  $T \in \mathcal{T}_{0,[n]}$  which decomposes into  $T_1$  and  $T_2$  (which are not reduced) as in Figure 2. We denote the number of vertices of  $T_1$  by  $i$ . We note whether the parent  $p$  of 1 in  $T$  becomes a descent, leaf, an increasing vertex, or the root in  $T_1$ . If  $p$  is the root of  $T_1$ , let  $v$  the least leaf in  $\Pi_i(T_1)$ . If  $p$  is the least leaf of  $T_1$ , let  $v$  be the root of  $\Pi_i(T_1)$ . In all other cases, suppose that the parent  $p$  is the  $k$ th greatest vertex of its type—descent, leaf, or increasing vertex—in  $T_1$ . Then let  $v$  be the vertex in  $\Pi_i(T_1)$  which is, respectively, the  $k$ th greatest leaf, descent, or increasing vertex (that is, if  $p$  is a leaf,  $v$  is a descent; if  $p$  is a descent,  $v$  is a leaf; and if  $p$  is an increasing vertex, so is  $v$ ). We then form  $\Pi_n(T)$  by adding an edge from  $v$  in  $\Pi_i(T_1)$  to 1 in  $\Pi_{n-i-1}(T_2)$ .

It is not hard to show by induction that  $\Pi_n$  is an involution for any nonnegative integer  $n$ . To obtain Theorem 1 from  $\Pi_n$ , we need to consider how to obtain the number of descents and leaves of  $\Pi_n(T)$  given  $T$  from each of the four cases, considered in the proof of Lemma 1, of how  $T$  is formed from  $T_1$  and  $T_2$ .

It turns out that  $\Pi_n$  is an extension of a well-known mapping  $\Pi_n^I$  which takes permutations to increasing trees (see, for example, [15, p. 25]). We can consider a permutation  $\pi = \pi(1)\pi(2)\dots\pi(n)$  of  $[n]$  to be a rooted tree with a single leaf in  $\mathcal{T}_{0,[n]}$ ,

such that the parent of  $\pi(1)$  is 0 and the parent of  $\pi(i)$  is  $\pi(i-1)$ . To map a permutation  $\pi$  to an increasing tree  $\Pi_n^I(\pi) \in \mathcal{T}_{0,[n]}$ , we use the following procedure:

- $\Pi^I$  1. Set  $i := 1$ . Define  $\pi(0) = 0$ . Set the tree  $T$  to a lone root labeled 0.
- $\Pi^I$  2. Let  $j$  be the largest index such that  $j < i$  and  $\pi(j) < \pi(i)$ . Set  $p := \pi(j)$ .
- $\Pi^I$  3. Make  $\pi(i)$  a child of  $p$  in  $T$ .
- $\Pi^I$  4. If  $i = n$ , stop;  $T$  is the  $\Pi_n^I(\pi)$ . If not, increase  $i$  by 1 and return to step  $\Pi^I$  2.

Figure 3 shows how the image of the permutation  $\pi = 34816275$  under  $\Pi_8^I$  is formed at three stages.

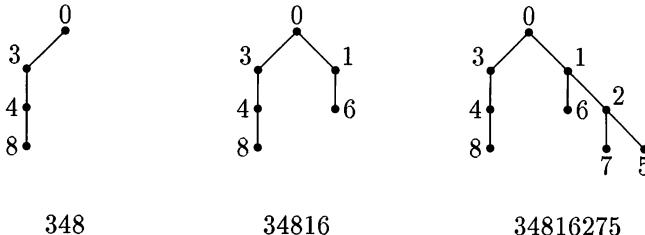


Figure 3: The bijection between permutations and increasing trees

It is not hard to prove by induction that  $\Pi_n$  restricted to rooted trees with one leaf is  $\Pi_n^I$ . First, note that for a given permutation  $\pi$ , any element appearing to the right of 1 will be a descendent of 1 in  $\Pi_n^I$ . Hence, to form  $\Pi_n^I(\pi)$ , we can write  $\pi$  as  $\pi_1\pi_2$ . In the example of Figure 3, we can take  $\pi_1 = 348$  and  $\pi_2 = 6275$ . We can then find the increasing trees  $\Pi_i^I(\pi_1)$  and  $\Pi_{n-1-i}^I(\pi_2)$ . We subsequently obtain  $\Pi_n^I(\pi)$  by changing the label of the root of  $\Pi_n^I(\pi_2)$  from 0 to 1 and making 1 a child of the root of  $\Pi_i^I(\pi_1)$ .

However, if  $T$  is the tree with one leaf corresponding to  $\pi$ , note that if we decompose  $T$  into two trees  $T_1$  and  $T_2$  as in Figure 2, the parent of the vertex 1 in  $T$  becomes the sole leaf of  $T_1$ . Therefore, when we combine  $\Pi_i(T_1)$  and  $\Pi_{n-1-i}(T_2)$  to form  $\Pi_n(T)$ , the vertex 1 must become a child of the root since in  $T$  it was a child of the only leaf of  $T_1$ . But by induction,  $\Pi_i(T_1) = \Pi_i^I(T_1)$  and  $\Pi_{n-1-i}(T_2) = \Pi_{n-1-i}^I(T_2)$ . Thus, the procedure described for finding  $\Pi_n^I(\pi)$  exactly mimics the procedure for finding  $\Pi_n(T)$ .

### 3 Descents and leaves in trees: Additional decompositions

The symmetry of the  $u_n$  can be proved using two other recurrences, in which switching  $\alpha$  and  $\beta$  in one yields the other. Since the  $u_n$  satisfy both, this will show that the  $u_n$  are symmetric. Since the proofs of these recurrences are very similar to the proof of (2), they are only briefly sketched; details are in [8].

In the first additional decomposition, as before, we create two rooted trees from one rooted tree in  $\mathcal{T}_{0,[n]}$ . The only difference from the decomposition in Section 2 is that now we delete the edge connecting  $n$  to its parent. We then relabel  $n$  to be 0 and take the reduction of each resulting tree. Figure 4 shows the decomposition of a tree using this procedure.

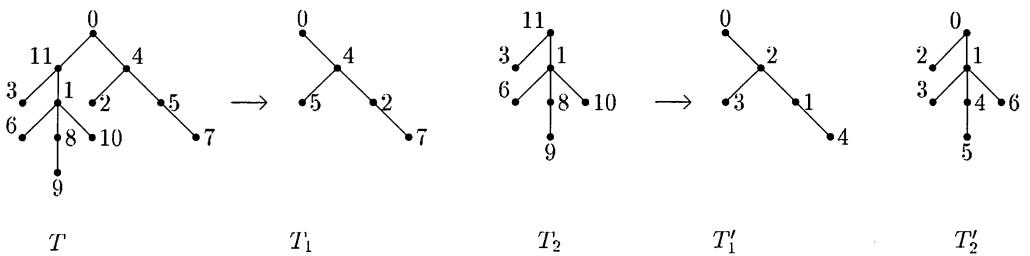


Figure 4: Second decomposition of trees.

As we similarly had for the first decomposition, the two reduced trees  $T'_1$  and  $T'_2$ , the content of  $T_1$ , and the parent of  $n$  in  $T$  determine  $T$ . Using this second decomposition, we obtain the following lemma (in which the arguments of  $u_n$  are suppressed).

**Lemma 2** *If  $n \geq 1$ , then*

$$\begin{aligned} u_n = & \frac{\partial}{\partial \beta} \beta u_{n-1} + \beta(nu_{n-1} - \frac{\partial}{\partial \beta} \beta u_{n-1}) \\ & + \sum_{i=0}^{n-2} \binom{n-1}{i} \alpha u_{n-1-i} \left( \frac{\partial}{\partial \beta} \beta u_i + \beta(i+1)u_i - \beta \frac{\partial}{\partial \beta} \beta u_i \right) \end{aligned} \quad (4)$$

**Idea of proof.** We perform an analysis similar to that used in the proof of Lemma 1. We will need four cases to obtain the recurrence. As in the proof of Lemma 1, in each case, we start with  $T'_1 \in \mathcal{T}_{0,[i]}$ , where  $0 \leq i \leq n-1$ , and  $T'_2 \in \mathcal{T}_{0,[n-i-1]}$ , and form a tree  $T \in \mathcal{T}_{0,[n]}$  by running the decomposition of Figure 4 backwards. The four cases are (i)  $0 \leq i \leq n-2$  and node  $n$  is made a child of a leaf of  $T_1$ ; (ii)  $0 \leq i \leq n-2$  and node  $n$  is made a child of a non-leaf of  $T_1$ ; (iii)  $i = n-1$  and node  $n$  is made a child of a leaf of  $T_1$ ; and (iv)  $i = n-1$  and node  $n$  is made a child of a non-leaf of  $T_1$ .  $\square$

From (4), we can derive the differential equation

$$\left( \frac{1}{\alpha U + 1} \right) \frac{\partial}{\partial x} U = \frac{\partial}{\partial \beta} (\beta U) + \beta \frac{\partial}{\partial x} (xU) - \beta \frac{\partial}{\partial \beta} (\beta U) + 1. \quad (5)$$

Since (4) is not symmetric, it does not prove Theorem 1. However, we can derive another recurrence from a different decomposition of trees which turns out to be (4) with  $\alpha$  and  $\beta$  swapped.

In this third decomposition, for  $T \in \mathcal{T}_{0,[n]}$ , we remove all subtrees of  $T$  rooted at siblings of 1 (if there are any), and we make the roots of these subtrees the children of a root node labeled 0. This tree will be called  $T_2$ . We then remove 1 from what is left of  $T$  and make the children of 1 into children of the parent of 1 in  $T$ . This tree is  $T_1$ . Finally, we take the reductions of  $T_1$  and  $T_2$ . Figure 5 demonstrates this procedure.

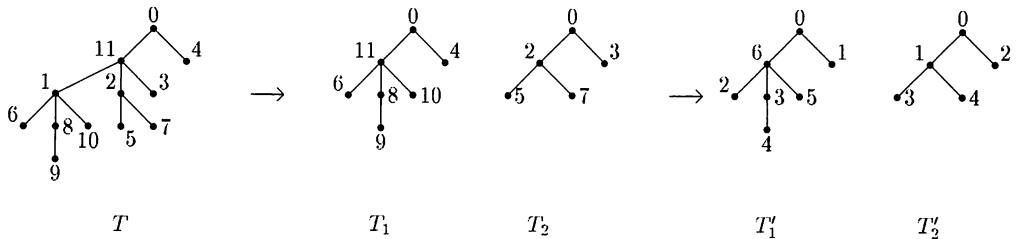


Figure 5: Third decomposition of trees.

From this decomposition, we obtain the following.

**Lemma 3** *If  $n \geq 1$ ,*

$$\begin{aligned} u_n = & \frac{\partial}{\partial \alpha} \alpha u_{n-1} + \alpha(n u_{n-1} - \frac{\partial}{\partial \alpha} \alpha u_{n-1}) \\ & + \sum_{i=0}^{n-2} \binom{n-1}{i} \beta u_{n-1-i} \left( \frac{\partial}{\partial \alpha} \alpha u_i + \alpha(i+1) u_i - \alpha \frac{\partial}{\partial \alpha} \alpha u_i \right). \end{aligned} \quad (6)$$

**Idea of proof.** As in the derivation of the previous recurrences, we first choose  $T'_1 \in \mathcal{T}_{0,[i]}$  and  $T'_2 \in \mathcal{T}_{0,[n-i-1]}$ , and then run the decomposition of Figure 5 backwards. There are five cases to consider, depending on the vertex  $v$  in  $T_1$  which becomes the parent of 1 in  $T$ : (i)  $1 \leq i \leq n-2$  and  $v$  is a descent or the root of  $T_1$ ; (ii)  $1 \leq i \leq n-2$  and  $v$  is neither a descent nor the root of  $T_1$ ; (iii)  $i = n-1$  and  $v$  is a descent or the root of  $T_1$ ; (iv)  $i = n-1$  and  $v$  is neither a descent nor the root of  $T_1$ ; and (v)  $i = 0$ .  $\square$

Note that (6) is simply (4) with  $\alpha$  and  $\beta$  swapped. Equations (4) and (6) thus provide another proof of Theorem 1.

As with the decompositions described in Section 2, we can use the decompositions in this section to describe bijections which transform trees with  $i$  descents and  $j+1$  leaves to trees with  $j$  descents and  $i+1$  leaves. The bijections are mutually inverse, but neither is an involution. They can be defined recursively in a manner like the bijection in Section 2. They are described explicitly in [8].

We now derive (1), Gessel's functional equation. In (6), we can replace  $u_0$  with 1. Then, if we multiply both sides by  $x^{n-1}/(n-1)!$  and sum both sides on  $n$  from 2 to  $\infty$ , we can obtain the differential equation

$$\left( \frac{1}{\beta U + 1} \right) \frac{\partial}{\partial x} U = \frac{\partial}{\partial \alpha} \alpha U + \alpha \frac{\partial}{\partial x} x U - \alpha \frac{\partial}{\partial \alpha} \alpha U + 1. \quad (7)$$

We can use (5) and (7) to eliminate the derivative with respect to  $\alpha$  and  $\beta$  from (3), obtaining

$$\left( \frac{1}{U+1} \right) \frac{\partial}{\partial x} U = \left( \frac{\alpha}{\alpha U + 1} \right) \frac{\partial}{\partial x} U + \left( \frac{\beta}{\beta U + 1} \right) \frac{\partial}{\partial x} U - \alpha \beta \frac{\partial}{\partial x} x U - \alpha - \beta + 1.$$

If we integrate both sides of the last equation, using the condition  $U = 0$  when  $x = 0$ , we obtain (1).

## 4 A generalization related to intransitive trees

Let  $v_n(\alpha, \beta)$  count trees in  $\mathcal{T}_{n,[n-1]}$  for which every vertex is either a descent or a leaf. In other words,  $v_n$  is the sum of the terms of total degree  $n$  in  $t_n$ ; so in particular,  $v_n$  is homogeneous. Using some known bijections [9,p. 333; 13; 14], it is not hard to show that the coefficient of  $\alpha^i\beta^{n-i}$  in  $v_n$  also counts intransitive trees on  $[n]$  (rooted at some vertex) in which  $i$  vertices are larger than their respective parent and children and  $n - i$  vertices are smaller than their respective parent and children. Knuth [10] found that if  $V = \sum_{n=2}^{\infty} \frac{v_n(\alpha, \beta)}{n!}$ , we have

$$\beta + \left( \alpha \frac{\partial}{\partial \alpha} + \beta \frac{\partial}{\partial \beta} \right) V = \beta \exp \left( \alpha + \alpha \frac{\partial}{\partial \alpha} V \right). \quad (8)$$

This equation can be derived from (5). To do this, recall  $t_n(\alpha, \beta) = \alpha\beta u_{n-1}(\alpha, \beta)$ , and let  $T(x; \alpha, \beta) = \sum_{n=2}^{\infty} t_n \frac{x^n}{n!}$ . Since  $t_n$  counts trees in  $\mathcal{T}_{n,[n-1]}$  by descents and leaves, we have

$$z^n t_n \left( \frac{\alpha}{z}, \frac{\beta}{z} \right) \Big|_{z=0} = v_n(\alpha, \beta), \quad T \left( z; \frac{\alpha}{z}, \frac{\beta}{z} \right) \Big|_{z=0} = V. \quad (9)$$

Moreover, by the definition of the  $t_n$ , we have

$$\alpha\beta \int_0^x U(y; \alpha, \beta) dy = T(x; \alpha, \beta). \quad (10)$$

Integrating both sides of (5) and using the fact that  $U = 0$  and  $T = 0$  when  $x = 0$ , we get

$$1 + \frac{1}{\beta} \frac{\partial}{\partial x} T = \exp \left( (1 - \beta) \frac{\partial}{\partial \beta} T + x \left( \frac{\partial}{\partial x} T + \alpha \right) \right).$$

We then multiply both sides of the previous equation by  $\beta$  to obtain

$$\beta + \frac{\partial}{\partial x} T = \beta \exp \left( \frac{\partial}{\partial \beta} T + \left( x \frac{\partial}{\partial x} T - \beta \frac{\partial}{\partial \beta} T \right) + x\alpha \right). \quad (11)$$

This equation generalizes (8), and we can easily obtain (8) from (11). We first multiply both sides of (11) by  $x$ . We then replace  $\alpha$  with  $\frac{\alpha}{z}$ ,  $\beta$  with  $\frac{\beta}{z}$ , and  $x$  with  $z$ . Finally, we set  $z = 0$ . Note that since  $v_n$  is homogenous of degree  $n$ , we see that  $\alpha \frac{\partial}{\partial \alpha} v_n + \beta \frac{\partial}{\partial \beta} v_n = nv_n$ , so that  $nV - \beta \frac{\partial}{\partial \beta} V = \alpha \frac{\partial}{\partial \alpha} V$ . From this and because of (9), we obtain (8).

We can also explain (11) combinatorially. The coefficient of  $\frac{x^n}{n!}$  in  $\frac{\partial}{\partial x} T$  counts rooted trees in  $\mathcal{T}_{\infty,[n]}$ . On the right side, the term  $\frac{\partial}{\partial \beta} T$  counts trees in  $\mathcal{T}_{n,[n-1]}$  which are additionally rooted at a leaf; the term  $x \frac{\partial}{\partial x} T - \beta \frac{\partial}{\partial \beta} T$  counts trees in  $\mathcal{T}_{n,[n-1]}$

additionally rooted at a non-leaf (perhaps the root); and  $\alpha x$  simply counts a lone vertex, which we can consider to be rooted tree additionally rooted at its root. To form an object counted by the left side, we take a set of trees  $T_1, T_2, T_3, \dots, T_{j-1}$ , whose reductions are counted by the exponent on the right and the set of whose contents is a partition of  $[n]$ . We add to these trees a tree  $T_j$  consisting of a lone root labeled  $-\infty$ ; this last tree corresponds to the extra factor of  $\beta$  on the right side of (11). Note that the greatest vertex in each tree is the root; we assume that if  $1 \leq i \leq j-1$ , the root of  $T_{j+1}$  is greater than the root of  $T_i$ . Next, for  $1 \leq i \leq j-1$ , we relabel the root of  $T_{i+1}$  with the root of  $T_i$ , and we relabel the root of  $T_1$  with  $\infty$ . Finally, for each  $i$  from 1 to  $j-1$ , we attach the root of  $T_{i+1}$  to the additional root of  $T_i$ . We now have an object counted by the left side of (11). Figure 6 gives an example of this procedure. In the figure, the additional roots of the  $T_i$  are marked with a circle, and the vertices which the additional roots give rise to in  $T$  are also marked with a circle. Note that because of the relabeling procedure, the circled vertices of  $T$  are labeled with a different set of elements from the circled vertices of the  $T_i$ .

To run this bijection the other way, given a tree  $T$ , identify the largest non-root vertex  $v_1$  in the tree (the vertex 10 in  $T$  in Figure 6). Eliminate the edge connecting  $v_1$  to its parent, relabel the vertex which was the root of  $T$  by  $v_1$ , and relabel the vertex originally labeled  $v_1$  by  $\infty$ . Take component of the remaining graph containing the former root of  $T$ , now labeled  $v_1$ , and give it an additional root at the vertex which was the parent of  $v_1$  in  $T$ ; this component will be  $T_1$ . Let  $C_1$  be the other component. We now perform the same procedure on  $C_1$  as we did on  $T$  originally to get  $T_2$  and another component  $C_2$ . We continue recursively until, for some  $j$ ,  $C_j$  is a lone vertex. We relabel this vertex  $-\infty$ , and we are done.

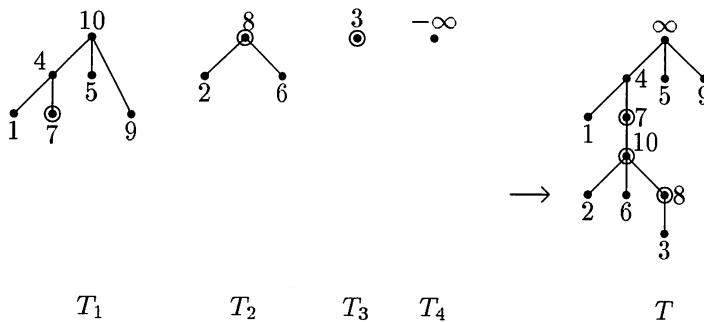


Figure 6: Interpretation of (11)

## 5 A symmetry in parking functions

Recurrence (4) has a simple interpretation in terms of parking functions. We prove the following theorem.

**Theorem 2** *The coefficient of  $\alpha^i \beta^j$  in  $u_n(\alpha, \beta)$  is the number of parking functions on  $[n]$  with  $i$  descents and  $j$  unfavorable spaces.*

**Proof.** For the moment, we denote by  $w_n(\alpha, \beta)$  the polynomials counting parking functions on  $[n]$  by descents (by  $\alpha$ ) and unfavorable spaces (by  $\beta$ ). We use a kind of composition of parking functions to show the  $w_n$  satisfy recurrence (4). Since it is easy to see that  $w_0 = 1$ ,  $w_1 = 1$ , this will show  $w_n = u_n$  for all positive integers  $n$ .

To derive the recurrence for the  $w_n$ , we form a parking function  $p$  on  $n$  from a parking function  $p'_1 \in P_i$  and a parking function  $p'_2 \in P_{n-1-i}$  ( $0 \leq i \leq n-1$ ) as follows. Let  $A = \{a_1, a_2, \dots, a_i\}$  be an  $i$ -element subset of  $[n-1]$ . Then let  $p_1$  be the parking function with content  $A$  and reduction  $p'_1$ , and let  $p_2$  be the parking function with content  $[n-1] - A = \{b_1, b_2, \dots, b_{n-1-i}\}$  and reduction  $p'_2$ . We define a parking function  $p \in P_n$  by letting  $p(n)$  be any element of  $[i+1]$  and setting

$$\begin{aligned} p(a_j) &= p_1(a_j), & j \in [i]; \\ p(b_j) &= p_2(b_j) + i + 1, & j \in [n-1-i]. \end{aligned}$$

Figure 7 demonstrates how this works. In the figure, if car  $a$  likes space  $j$  (i.e.,  $p(a) = j$ ) in a parking function,  $a$  is written in space  $j$  of the diagram. Additionally, for a parking function  $p$ , the notation  $p = (a_1 \ a_2 \ \dots \ a_n)$  means  $p(i) = a_i$ . In the middle step, we see that  $A = \{1, 3, 5\}$ . In the last step, car 8 is assigned space 2 (i.e.,  $p(8) = 2$ ), and the two sequences of spaces are joined by putting a new empty space between them.

To recover the original two parking functions, it is simplest to consider the parking function as numbers written in boxes. Then, for a parking function on  $n$  cars, let  $i+1$  be the space car  $n$  parks in (often not equal to  $p(n)$ ). If we simply erase car  $n$  from the diagram and remove space  $i+1$ , we will end up with two parking functions (one possibly the “empty” function). This follows because no car except car  $n$  will attempt to park in the space car  $n$  parks in; see [7, sec. 5] for more on this property of parking functions. If we take the reduction of the parking functions, we end up the original two.

$$\begin{array}{|c|c|} \hline 13 & 2 & \square \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 4 & 13 & 2 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{|c|c|} \hline 15 & 3 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 7 & 26 & 4 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{|c|c|c|c|c|} \hline 15 & 38 & \square & 7 & 26 & 4 \\ \hline \end{array}$$

$$p'_1 = (1 \ 2 \ 1) \quad p'_2 = (2 \ 4 \ 2 \ 1) \quad \quad \quad p_1 \quad \quad \quad p_2 \quad \quad \quad p = (1 \ 6 \ 2 \ 8 \ 1 \ 6 \ 5 \ 2)$$

Figure 7: Composition of parking functions

It is not too difficult to see that the composition process must yield a parking function. By construction, the cars in  $A$  will park in the first  $i$  spaces; and the cars in  $[n-1] - A$  will park in the last  $n-1-i$  spaces. When car  $n$  goes to park, it will start looking in one of the first  $i+1$  spaces, but the only one of those spaces still empty will be  $i+1$ , so car  $n$  will park there.

We again consider four cases in order to obtain  $w_n$  in terms of  $w_i$  with  $i < n$ . The cases depend on how we choose  $p(n)$ .

*Case i:*  $p_1 \in P_i$  with  $0 \leq i \leq n-2$  and either  $p(n)$  is an unfavorable space of  $p_1$  or  $p(n) = i+1$ . The parking function  $p$  will have one more descent than the sum of the descents of  $p_1$  and  $p_2$ , since by  $p$ , car  $n$  will park in space  $i+1 \neq n$ . As well, the number of unfavorable spaces of  $p$  equals the sum of the number of unfavorable spaces of  $p_1$  and  $p_2$ . So the contribution to  $w_n$  for this case is

$$\binom{n-1}{i} \alpha w_{n-1-i} \frac{\partial}{\partial \beta} \beta w_i.$$

*Case ii:*  $p_1 \in P_i$  with  $0 \leq i \leq n-2$ ,  $p(n) \neq i+1$ , and  $p(n)$  is not an unfavorable space of  $p_1$ . Again, the parking function  $p$  will have one more descent than the sum of the descents of  $p_1$  and  $p_2$ , since by  $p$ , car  $n$  will park in space

$i+1 \neq n$ . Moreover,  $p$  has one more unfavorable space than the sum of the unfavorable spaces of  $p_1$  and  $p_2$ , since space  $i+1$  is an unfavorable space of  $p$ . So the contribution to  $w_n$  for this case is

$$\binom{n-1}{i} \alpha \beta w_{n-1-i} \left( (i+1)w_i - \frac{\partial}{\partial \beta} \beta w_i \right).$$

*Case iii:*  $p_1 \in P_{n-1}$  and  $p(n)$  is an unfavorable space of  $p_1$  or  $p(n) = n$ . The parking function  $p$  will have the same number of descents as  $p_1$ , since car  $n$  parks in space  $n$  under  $p$ . Also,  $p$  has the same number of unfavorable spaces as  $p_1$ . The contribution to  $w_n$  for this case is

$$\frac{\partial}{\partial \beta} \beta w_{n-1}.$$

*Case iv:*  $p_1 \in P_{n-1}$ ,  $p(n) \neq n$ , and  $p(n)$  is not an unfavorable space of  $p_1$ . The parking function  $p$  will have the same number of descents as  $p_1$ , and  $p$  has one more unfavorable space than  $p_1$ . The contribution to  $w_n$  for this case is

$$\beta(nw_{n-1} - \frac{\partial}{\partial \beta} \beta w_{n-1}).$$

If we add up the four cases, we get exactly recurrence (4) with  $w$  written for  $u$ . Since  $w_0 = u_0 = w_1 = u_1 = 1$ , we conclude that  $w_n = u_n$  for all positive integers  $n$ . Since the  $u_n$  are symmetric, this shows that the number of parking functions with  $i$  descents and  $j$  unfavorable spaces equals the number of parking functions with  $j$  descents and  $i$  unfavorable spaces.  $\square$

## 6 Further comments

The symmetry in trees can be generalized to a symmetry in binary trees first found by Gessel [5]. In a labeled binary tree, we can define left descents, right descents, left ascents, and right ascents. There are several obvious symmetries, but Gessel showed the non-trivial fact that the number of labeled binary trees on  $[n]$  with  $i$  left descents and  $j$  right ascents equals the number of binary trees on  $[n]$  with  $j$  left descents and  $i$  right ascents. Gessel showed this by establishing a particular function equation, but the result can also be proven using decompositions similar to those in sections 2 and 3 [8].

Although Theorem 2 demonstrates a symmetry between unfavorable spaces and descents in parking functions, a nice bijection on parking functions that swaps those statistics seems elusive, since the recursion Theorem 2 is based on is not symmetric. It seems nevertheless such a bijection should exist.

## 7 Acknowledgments

This work is taken from the author's Ph.D. thesis, written under the supervision of Ira Gessel. The author thanks Ira Gessel for suggesting study of this symmetry and for his help and encouragement.

## References

- [1] D. Foata and J. Riordan, Mappings of acyclic and parking functions, *Aequationes Math.* **10** (1974), 10–22.
- [2] Jean Françon, Acyclic and parking functions, *J. Combinatorial Theory Ser. A* **18** (1975), 27–35.
- [3] I. M. Gessel, A  $q$ -analog of the exponential formula, *Discrete Mathematics* **40** (1982), 69–80.
- [4] I. M. Gessel, Counting forests by descents and leaves, *The Foata Festschrift, El. J. Combin.*, **3** (1996), no. 2, #R8, 5 pp.
- [5] I. M. Gessel, unpublished manuscript (1996).
- [6] I. M. Gessel and K.-Y. Wang, A bijective approach to the permutational power of a priority queue, unpublished (1994).
- [7] J. D. Gilbey and L. H. Kalikow, Parking functions, valet functions and priority queues, *Discrete Mathematics*, **197-198** (1999), nos.1–3, 351–373.
- [8] L. H. Kalikow, *Enumeration of parking functions, allowable permutation pairs, and labeled trees*, Ph.D. thesis, Brandeis University (1999).
- [9] D. E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, MA (1973).
- [10] D. E. Knuth, Solution to problem 10572, unpublished (1997). The original problem was proposed by Richard P. Stanley, Problem 10572 in the *American Mathematical Monthly*, **104** (1997), 168.
- [11] A. G. Konheim, and B. Weiss, An occupancy discipline and applications, *SIAM J. Applied Math.*, **14** (1966), 1266–1274.
- [12] G. Kreweras, Une famille de polynomes ayant plusieurs propriétés énumératives, *Periodica Mathematica Hungarica* **11** (1980), 309–320.
- [13] A. G. Kuznetzov, I. M. Pak, A. E. Postnikov, Increasing trees and alternating permutations, *Russian Math. Surveys* **49** (1994), no. 6, 79–110.
- [14] A. Postnikov, Private communication (to I. Gessel, 1995).
- [15] Richard P. Stanley, *Enumerative Combinatorics, Volume 1*, Cambridge University Press, Cambridge (1997). (First published by Wadsworth & Brooks/Cole, Monterey, CA, 1986.)
- [16] Richard P. Stanley, *Enumerative Combinatorics, Volume 2*, Cambridge University Press, Cambridge (1999).

# Nouvelles statistiques de partitions pour les $q$ -nombres de Stirling de seconde espèce

Gerald Ksavrelof et Jiang Zeng

Institut Girard Desargues

Université Claude Bernard (Lyon 1)

69622 Villeurbanne Cedex, France

[ksavrelo@desargues.univ-lyon1.fr](mailto:ksavrelo@desargues.univ-lyon1.fr), [zeng@desargues.univ-lyon1.fr](mailto:zeng@desargues.univ-lyon1.fr)

## Abstract

We study some new statistics on the partitions of a finite set whose generating functions are the  $q$ -Stirling numbers of second kind. In particular we prove a conjecture of Steingrímsson [7, Conj. 12].

## Résumé

Nous étudions plusieurs nouvelles statistiques sur les partitions d'un ensemble fini dont les fonctions génératrices sont les  $q$ -nombres de Stirling de seconde espèce. En particulier on démontre une conjecture de Steingrímsson [7, Conj. 12].

## 1 Introduction

Les  $q$ -nombres de Stirling de seconde espèce, notés  $S_q(n, k)$ , peuvent être définis par la relation de récurrence:

$$S_q(n, k) = \begin{cases} q^{k-1} S_q(n-1, k-1) + [k]_q S_q(n-1, k), & \text{si } 1 \leq k \leq n \\ \delta_{n, k} & \text{si } n = 0 \text{ ou } k = 0 \end{cases} \quad (1)$$

où  $[k]_q = 1 + q + \cdots + q^{k-1}$  pour tout entier  $k \geq 1$ .

Beaucoup d'auteurs [5, 4, 9, 1, 6, 7] ont étudié des interprétations combinatoires de ces  $q$ -nombres. Comme  $S_q(n, k)$  compte le nombre de partitions en  $k$  blocs de  $[n] = \{1, 2, \dots, n\}$  lorsque  $q = 1$ , il est donc naturel de chercher des statistiques sur les partitions pour interpréter ces  $q$ -nombres. Soit  $\mathcal{P}_n^k$  l'ensemble des *partitions* en  $k$  *blocs* de  $[n]$ . Pour toute partition  $\pi = B_1 - \cdots - B_k \in \mathcal{P}_n^k$  on convient que les blocs sont classés par ordre croissant de leurs plus petits éléments et on note  $b_i$  le cardinal de  $B_i$ .

On pourrait distinguer deux types de statistiques sur  $\mathcal{P}_n^k$ : celles dont la vérification de la récurrence (1) est facile et celles dont la vérification de la récurrence (1) est difficile [9]. Par exemple, si l'on pose  $\text{Mil}(\pi) = \sum_{i=1}^k (i-1)b_i$ , alors il est facile (voir [5]) de vérifier que  $\sum_{\pi \in \mathcal{P}_n^k} q^{\text{Mil}(\pi)}$  satisfait (1). D'autre part, il existe des statistiques ayant

pour fonction génératrice  $S_q(n, k)$ , mais la récurrence (1) est beaucoup plus difficile à vérifier (voir [9]). Dans cet article nous étudions quelques nouvelles statistiques sur  $\mathcal{P}_n^k$  ayant pour fonction génératrice  $S_q(n, k)$ , mais la vérification de la récurrence (1) n'est pas immédiate. Pour ce faire, on a besoin d'introduire quelques définitions.

Etant donnée une partition  $\pi$  de  $[n]$  nous répartissons les entiers de  $[n]$  en quatre types de la manière suivante:

- un *ouvrant* est le plus petit élément d'un bloc ;
- un *fermant* est le plus grand élément d'un bloc ;
- un *passant* est un élément ni ouvrant ni fermant d'un bloc non réduit à un seul élément ;
- un *singleton* est l'élément d'un bloc qui n'a qu'un seul élément.

On note respectivement  $\mathcal{O}(\pi)$ ,  $\mathcal{F}(\pi)$ ,  $\mathcal{P}(\pi)$  et  $\mathcal{S}(\pi)$  l'ensemble des ouvrants, fermants, passants et singltons de  $\pi$ . Il est évident que  $\mathcal{S}(\pi) = \mathcal{O}(\pi) \cap \mathcal{F}(\pi)$ . Par exemple, si  $\pi = 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6$ , alors  $\mathcal{O}(\pi) = \{1, 2, 3, 5\}$ ,  $\mathcal{F}(\pi) = \{8, 2, 9, 6\}$ ,  $\mathcal{P}(\pi) = \{4, 7\}$  et  $\mathcal{S}(\pi) = \{2\}$ .

**Definition 1 .** Soit  $\pi = B_1 - \cdots - B_k \in \mathcal{P}_n^k$  et soit  $\tilde{B}_i$  l'indice du bloc comprenant l'entier  $i \in [n]$ . On pose alors:

$$\begin{aligned}\text{ros}_i(\pi) &= \#\{j \in \mathcal{O}(\pi) \mid i > j, \tilde{B}_j > \tilde{B}_i\}. \\ \text{rob}_i(\pi) &= \#\{j \in \mathcal{O}(\pi) \mid i < j, \tilde{B}_j > \tilde{B}_i\}. \\ \text{rcs}_i(\pi) &= \#\{j \in \mathcal{F}(\pi) \mid i > j, \tilde{B}_j > \tilde{B}_i\}. \\ \text{rcb}_i(\pi) &= \#\{j \in \mathcal{F}(\pi) \mid i < j, \tilde{B}_j > \tilde{B}_i\}. \\ \text{los}_i(\pi) &= \#\{j \in \mathcal{O}(\pi) \mid i > j, \tilde{B}_j < \tilde{B}_i\}. \\ \text{lob}_i(\pi) &= \#\{j \in \mathcal{O}(\pi) \mid i < j, \tilde{B}_j < \tilde{B}_i\}. \\ \text{lcs}_i(\pi) &= \#\{j \in \mathcal{F}(\pi) \mid i > j, \tilde{B}_j < \tilde{B}_i\}. \\ \text{lcb}_i(\pi) &= \#\{j \in \mathcal{F}(\pi) \mid i < j, \tilde{B}_j < \tilde{B}_i\}.\end{aligned}$$

Ainsi on peut définir  $\text{ros}(\pi) = \sum_i \text{ros}_i(\pi)$  et de même pour les autres statistiques.

**Remarque.**  $\text{ros}$  est l'abréviation en anglais pour “right, opener, smaller”, et de même  $\text{lcb}$  est celle pour “left, closer, bigger”, etc (voir [7]). Les huit statistiques ci-dessus sont essentiellement dues à Wachs et White [9].

La définition suivante est due à Steingrímsson [7], qui a remarqué qu'il existe un lien très étroit entre la statistique mak et celle définie par Foata-Zeilberger [3] sur les permutations, et qui porte le même nom.

**Definition 2 .** Pour tout  $\pi \in \mathcal{P}_n^k$ , on pose

$$\begin{aligned}\text{mak}(\pi) &= \text{ros}(\pi) + \text{lcs}(\pi), & \text{lmak}'(\pi) &= n(k-1) - [\text{lcb}(\pi) + \text{rob}(\pi)]. \\ \text{mak}'(\pi) &= \text{lob}(\pi) + \text{rcb}(\pi), & \text{lmak}(\pi) &= n(k-1) - [\text{los}(\pi) + \text{rcs}(\pi)].\end{aligned}$$

Donnons un exemple de calculs des statistiques précédemment définies.

**Exemple.** Soit  $\pi = 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6$ , alors on a:

$$\begin{array}{ll} \pi = & 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6 \\ \text{ros}_i : & \begin{matrix} 0 & 2 & 3 & 0 & 0 & 1 & 1 & 0 & 0 \end{matrix} \\ \text{lcs}_i : & \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \end{matrix} \\ \text{lob}_i : & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \\ \text{rcb}_i : & \begin{matrix} 3 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \end{matrix} \end{array} \quad \begin{array}{ll} \pi = & 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6 \\ \text{lcb}_i : & \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 2 & 2 \end{matrix} \\ \text{rob}_i : & \begin{matrix} 3 & 1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \end{matrix} \\ \text{los}_i : & \begin{matrix} 0 & 0 & 0 & 1 & 2 & 2 & 2 & 3 & 3 \end{matrix} \\ \text{rcs}_i : & \begin{matrix} 0 & 1 & 2 & 0 & 0 & 1 & 1 & 0 & 0 \end{matrix} \end{array}$$

On en déduit donc d'après la définition:

$$\begin{aligned} \text{mak}(\pi) &= 7 + 6 = 13, & \text{lmak}'(\pi) &= 27 - (7 + 7) = 13, \\ \text{mak}'(\pi) &= 0 + 9 = 9, & \text{lmak}(\pi) &= 27 - (13 + 5) = 9. \end{aligned}$$

Cet article a été motivé par la conjecture suivante de Steingrímsson [7, Conj. 12].

**Conjecture 1 (Steingrímsson)** *Les quatre statistiques mak, mak', lmak et lmak' ont pour fonction génératrice sur  $\mathcal{P}_n^k$  les q-nombres de Stirling  $S_q(n, k)$ , c'est-à-dire,*

$$\sum_{\pi \in \mathcal{P}_n^k} q^{\text{mak}(\pi)} = \sum_{\pi \in \mathcal{P}_n^k} q^{\text{lmak}'(\pi)} = \sum_{\pi \in \mathcal{P}_n^k} q^{\text{mak}'(\pi)} = \sum_{\pi \in \mathcal{P}_n^k} q^{\text{lmak}(\pi)} = S_q(n, k).$$

En fait, la statistique mak' a été introduite par Wachs et White [9], qui l'appellent rb. On pourrait donc déduire une preuve de la conjecture ci-dessus d'un résultat de Wachs et White [8]. Ici on propose une approche différente, qui nous a permis d'obtenir un résultat plus général comme suit.

Pour tout ensemble fini  $B$  d'entiers on désigne respectivement par  $p(B)$  et  $g(B)$  le plus petit et le plus grand élément de  $B$ .

**Definition 3 .** Soit  $\pi = B_1 - \cdots - B_k$  une partition de  $\mathcal{P}_n^k$  et  $b$  un élément du bloc  $B_j$ , on définit  $\text{rinv}(b, B_i) = \#\{a \in B_i \mid b > a\}$  pour tout  $i > j$ , et  $\text{linv}(b, B_i) = \#\{a \in B_i \mid b < a\}$  pour tout  $i < j$ , ainsi que  $\text{nrinv}(b, \pi) = \#\{a \in B_i \mid i > j, b < a\}$ . On peut alors poser:

$$\begin{aligned} \text{rinv}(b, \pi) &= \sum_{i>j} \text{rinv}(b, B_i), & \text{linv}(b, \pi) &= \sum_{i<j} \text{linv}(b, B_i). \\ \text{rinv } \mathcal{F}(\pi) &= \sum_{b \in \mathcal{F}(\pi)} \text{rinv}(b, \pi), & \text{linv } \mathcal{O}(\pi) &= \sum_{b \in \mathcal{O}(\pi)} \text{linv}(b, \pi). \end{aligned}$$

Pour tout  $\pi = B_1 - \cdots - B_k \in \mathcal{P}_n^k$  et  $0 \leq l \leq k-1$ , on généralise la statistique mak comme suit:

$$\text{mak}_l(\pi) = \text{mak}(\pi) - \text{nrinv}(g(B_{l+1}), \pi) + k-1-l, \quad \forall \pi \in \mathcal{P}_n^k.$$

**Théorème 1** Pour  $0 \leq l \leq k-1$ , on a

$$\sum_{\pi \in \mathcal{P}_n^k} q^{\text{mak}_l(\pi)} = S_q(n, k).$$

La démonstration de la conjecture et théorème ci-dessus se décompose en plusieurs étapes. Dans la section 2 on établit une involution sur  $\mathcal{P}_n^k$ , qui échange mak et mak'. Dans la section 3 on identifie les quatre statistiques ci-dessus deux par deux en introduisant des nouvelles notations. Enfin, dans la section 4, on montre que la fonction génératrice de mak satisfait la récurrence (1), ce qui achève la démonstration. Nous terminons l'article avec quelques remarques sur les problèmes ouverts.

## 2 Bijection entre mak et mak'

Nous avons besoin de quelques définitions supplémentaires. Pour tout ensemble fini d'entiers  $B$  et entier  $i$ , on note  $B(\leq i)$  la restriction de  $B$  sur  $[i]$ , qui est soit *complet*, si l'ensemble  $B$  est inclus dans  $[i]$ , soit *incomplet*, si une partie non vide de  $B$  est dans

[i] et l'autre partie non vide dans  $[n] \setminus [i]$ , soit *vide* si  $B \cap [i] = \emptyset$ . Etant donnée une partition  $\pi = B_1 - B_2 - \cdots - B_k \in \mathcal{P}_n^k$  et un entier  $i$  tel que  $1 \leq i \leq n$ , on définit la *trace*  $T_i$  de  $\pi$  sur  $[i]$  comme suit:

$$T_i = B_1(\leq i) - B_2(\leq i) - \cdots - B_k(\leq i),$$

Il est clair qu'une partition est entièrement déterminée par ses traces  $T_1, T_2, \dots, T_n$ . Posons  $T_0 = \emptyset$ , pour  $i = 1, 2, \dots, n$ , on définit  $l_i(\pi)$  comme le nombre de blocs incomplets dans  $T_{i-1}$  et  $\gamma_i(\pi) - 1$  le nombre de blocs incomplets situés à gauche du bloc contenant  $i$  dans  $T_i$ . Par exemple, si  $\pi = 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6$ , alors la trace sur [5] est

$$T_5 = 1\ 4. - 2 - 3. - 5.,$$

où on ajoute un point dans chaque bloc incomplet. On remarque que  $l_5(\pi) = 2$  et  $\gamma_5(\pi) = 3$ .

Nous donnons d'abord une involution  $\varphi : \mathcal{P}_n^k \rightarrow \mathcal{P}_n^k$  définie par l'algorithme suivant:

1. Etant donnée une partition  $\pi$  de  $[n]$ , on partage l'ensemble  $[n]$  en quatre parties  $\mathcal{S}(\pi)$ ,  $\mathcal{O}_s(\pi) = \mathcal{O}(\pi) \setminus \mathcal{S}(\pi)$ ,  $\mathcal{F}_s(\pi) = \mathcal{F}(\pi) \setminus \mathcal{S}(\pi)$  et  $\mathcal{P}(\pi)$ , et on calcule les  $\gamma_i$  pour tout  $i \in [n]$ . Notons  $f$  (resp.  $p$ ) la suite croissante des éléments de  $\mathcal{F}_s(\pi)$  (resp.  $\mathcal{P}(\pi)$ ). On forme alors les deux matrices:

$$\begin{pmatrix} f \\ \gamma \end{pmatrix} = \begin{pmatrix} f_1 & f_2 & \cdots & f_r \\ \gamma_{f_1} & \gamma_{f_2} & \cdots & \gamma_{f_r} \end{pmatrix}, \quad \begin{pmatrix} p \\ \gamma \end{pmatrix} = \begin{pmatrix} p_1 & p_2 & \cdots & p_s \\ \gamma_{p_1} & \gamma_{p_2} & \cdots & \gamma_{p_s} \end{pmatrix}.$$

2. Posons  $\mathcal{S}' = \{n+1-i \mid i \in \mathcal{S}(\pi)\}$ ,  $\mathcal{O}'_s = \{n+1-j \mid j \in \mathcal{F}_s(\pi)\}$ ,  $\mathcal{F}'_s = \{n+1-i \mid i \in \mathcal{O}_s(\pi)\}$  et  $\mathcal{P}' = \{n+1-i \mid i \in \mathcal{P}(\pi)\}$ . Soient  $f'$  et  $p'$  les suites croissantes des éléments de  $\mathcal{F}'_s$  et  $\mathcal{P}'$ , formons les deux matrices:

$$\begin{pmatrix} f' \\ \gamma' \end{pmatrix} = \begin{pmatrix} f'_1 & f'_2 & \cdots & f'_r \\ \gamma_{f'_1} & \gamma_{f'_2} & \cdots & \gamma_{f'_r} \end{pmatrix}, \quad \begin{pmatrix} p' \\ \gamma' \end{pmatrix} = \begin{pmatrix} p'_1 & p'_2 & \cdots & p'_s \\ \gamma_{p'_s} & \gamma_{p'_{s-1}} & \cdots & \gamma_{p'_1} \end{pmatrix}.$$

3. Construisons les  $k$  blocs (complets ou incomplets) avec les éléments de  $\mathcal{O}'$ :

$$\pi_0 = B_1 - B_2 - \cdots - B_k.$$

Supposons que  $x_1, x_2, \dots, x_{n-k}$  est le réarrangement croissant des éléments de  $\mathcal{F}'_s \cup \mathcal{P}'$ . Pour  $j = 1, \dots, n-k$  on insère  $x_j$  dans  $\pi_{j-1}$  en tenant compte de  $\gamma'_{x_j}$  et on obtient  $\pi_j$ .

4. Définissons  $\varphi(\pi) = \pi' = \pi_{n-k}$ , alors  $\mathcal{S}(\pi') = \mathcal{S}'$ ,  $\mathcal{O}_s(\pi') = \mathcal{O}'_s$ ,  $\mathcal{F}_s(\pi') = \mathcal{F}'_s$ ,  $\mathcal{P}(\pi') = \mathcal{P}'$ .

On vérifie que  $\varphi$  est une involution sur  $\mathcal{P}_n^k$  telle que  $\mathcal{F}_s(\varphi(\pi)) = \{n+1-i \mid i \in \mathcal{O}_s(\pi)\}$ .

**Remarque.** On pourrait décrire l'involution ci-dessus dans le modèle de *chemins de Motzkin valués* (voir [2]), dans lequel l'involution  $\varphi$  peut être visualisée comme une “symétrie miroir” par rapport à l'axe (Oy).

**Exemple.** Reprenons la partition  $\pi = 1\ 4\ 8 - 2 - 3\ 7\ 9 - 5\ 6$  de l'exemple précédent, alors  $\mathcal{O}_s = \{1, 3, 5\}$ ,  $\mathcal{F}_s = \{8, 9, 6\}$ ,  $\mathcal{P} = \{4, 7\}$  et  $\mathcal{S} = \{2\}$ . On en déduit donc  $\mathcal{F}'_s = \{9, 7, 5\}$ ,  $\mathcal{O}'_s = \{2, 1, 4\}$ ,  $\mathcal{P}' = \{6, 3\}$  et  $\mathcal{S}' = \{8\}$ . Ainsi on a

$$\begin{pmatrix} f \\ \gamma \end{pmatrix} = \begin{pmatrix} 6 & 8 & 9 \\ 3 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} p \\ \gamma \end{pmatrix} = \begin{pmatrix} 4 & 7 \\ 1 & 2 \end{pmatrix},$$

et

$$\begin{pmatrix} f' \\ \gamma' \end{pmatrix} = \begin{pmatrix} 5 & 7 & 9 \\ 3 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} p' \\ \gamma' \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 2 & 1 \end{pmatrix}.$$

On obtient d'abord les  $k = 4$  blocs (complets ou incomplets) formés des éléments de  $\mathcal{O}'$ :

$$\pi_0 = 1. - 2. - 4. - 8.$$

On insère successivement les éléments  $i$  de  $\mathcal{F}'_s \cup \mathcal{P}'$  en tenant compte de  $\gamma_i$  et on obtient

$$\begin{aligned} \pi_1 &= 1. - 23. - 4. - 8 \\ \pi_2 &= 1. - 23. - 45 - 8 \\ \pi_3 &= 16. - 23. - 45 - 8 \\ \pi_4 &= 167 - 23. - 45 - 8 \\ \pi_5 &= 167 - 239 - 458. \end{aligned}$$

D'où  $\pi' = \pi_5 = 167 - 239 - 45 - 8$ . On vérifie que  $(\pi')' = \pi$ .

**Lemme 1** Soit  $\pi \in \mathcal{P}_n^k$  une partition fixée, on pose  $\mathcal{O} = \mathcal{O}(\pi)$ ,  $\mathcal{F}_s = \mathcal{F}(\pi) \setminus \mathcal{S}(\pi)$ ,  $\mathcal{P} = \mathcal{P}(\pi)$ ,  $l_i = l_i(\pi)$  et  $\gamma_i = \gamma_i(\pi)$ . Alors, les statistiques mak et mak' peuvent s'écrire:

$$\begin{aligned} \text{mak}(\pi) &= \sum_{i \in \mathcal{F}_s \cup \mathcal{P}} (l_i - \gamma_i + \#\{a \in \mathcal{F} | a < i\}) + \sum_{i \in \mathcal{O}} \#\{a \in \mathcal{F} | a < i\}. \\ \text{mak}'(\pi) &= \sum_{i \in \mathcal{F}_s \cup \mathcal{P}} (k - \gamma_i - \#\{a \in \mathcal{F} | a < i\}) + \sum_{i \in \mathcal{O}} (k - l_i - \#\{a \in \mathcal{F} | a < i\}). \end{aligned}$$

**Preuve :** Pout tout  $i \in [n]$ , si  $i \in \mathcal{F}_s \cup \mathcal{P}$ , dans la trace  $T_i$  de  $\pi$ , on constate que  $\text{ros}_i(\pi)$  compte le nombre de blocs (complets ou incomplets) à droite du bloc contenant  $i$ , et  $\text{lcs}_i(\pi)$  le nombre de blocs complet à gauche de bloc contenant  $i$ , ainsi  $(\text{ros}_i + \text{lcs}_i)(\pi) = l_i - \gamma_i + \#\{a \in \mathcal{F} | a < i\}$ ; si  $i \in \mathcal{O}$ , alors  $\text{ros}_i(\pi) = 0$  et  $\text{lcs}_i(\pi) = \#\{a \in \mathcal{F}(\pi) | a < i\}$ . D'où la première égalité. De même, comme  $\text{rcb}_i(\pi)$  comptent non seulement les blocs incomplets à droite du bloc contenant  $i$  dans  $T_i$ , c'est à dire  $l_i - \gamma_i$  si  $i \in \mathcal{F}_s \cup \mathcal{P}$  et 0 sinon, mais aussi les blocs qui ne sont pas encore créés, soit  $k - l_i - \#\{a \in \mathcal{F} | a < i\}$ ; en sommant sur  $i$  on obtient la seconde égalité.  $\square$

**Théorème 2** Pour tout  $\pi \in \mathcal{P}_n^k$ , on a  $\text{mak}(\pi) = \text{mak}'(\phi(\pi))$ .

**Preuve :** D'après la construction de  $\phi(\pi)$ , on a

$$\sum_{i \in \mathcal{F}_s} \gamma_i = \sum_{i \in \mathcal{F}'_s} \gamma'_i, \quad \sum_{i \in \mathcal{P}} \gamma_i = \sum_{i \in \mathcal{P}'} \gamma'_i.$$

D'autre part, pour tout  $i \in [n]$  on a

$$k = \begin{cases} l_i + \#\{a \in \mathcal{O} | a > i\} + \#\{a \in \mathcal{F} | a < i\}, & \text{si } i \in \mathcal{P} \cup \mathcal{S}, \\ 1 + l_i + \#\{a \in \mathcal{O} | a > i\} + \#\{a \in \mathcal{F} | a < i\}, & \text{si } i \in \mathcal{O}_s \cup \mathcal{F}_s. \end{cases} \quad (2)$$

On en déduit donc

$$\sum_{i \in \mathcal{P} \cup \mathcal{S}} (l_i - \gamma_i + \#\{a \in \mathcal{F} | a < i\}) = \sum_{i \in \mathcal{P} \cup \mathcal{S}} (k - \gamma_i - \#\{a \in \mathcal{O} | a > i\}).$$

Il en résulte du lemme 1 que l'identité  $\text{mak}(\pi) = \text{mak}'(\phi(\pi))$  équivaut à:

$$\begin{aligned} & \sum_{i \in \mathcal{F}_s} (l_i + \#\{a \in \mathcal{F} \mid a < i\}) + \sum_{i \in \mathcal{O}_s} \#\{a \in \mathcal{F} \mid a < i\} \\ &= \sum_{i \in \mathcal{O}_s} (k - \#\{a \in \mathcal{O} \mid a > i\}) + \sum_{i \in \mathcal{F}_s} (k - l_i - \#\{a \in \mathcal{O} \mid a > i\}), \end{aligned}$$

qui s'écrit encore

$$\begin{aligned} & \sum_{i \in \mathcal{F}_s} (2l_i - k + \#\{a \in \mathcal{O} \mid a > i\} + \#\{a \in \mathcal{F} \mid a < i\}) \\ &= \sum_{i \in \mathcal{O}_s} (k - \#\{a \in \mathcal{O} \mid a > i\} - \#\{a \in \mathcal{F} \mid a < i\}). \end{aligned}$$

Compte tenu de (2) l'équation ci-dessus équivaut à:

$$\sum_{i \in \mathcal{F}_s} l_i = \sum_{i \in \mathcal{O}_s} (l_i + 1). \quad (3)$$

On remarque d'abord qu'il y a autant de fermants que d'ouvrants dans  $\pi$ , et  $\forall i \in [n]$ ,  $l_i \geq 0$  et  $l_{i+1} = l_i + 1$  (resp.  $l_{i+1} = l_i - 1$ ) si  $i \in \mathcal{O}_s$  (resp. si  $i \in \mathcal{F}_s$ ). On va construire une bijection de  $\mathcal{O}_s$  dans  $\mathcal{F}_s$  telle que si  $a \mapsto a'$ , alors  $l_a + 1 = l_{a'}$ . Ce qui démontre clairement l'égalité (3). En effet, soit  $\mathcal{O}_s = \{a_1, \dots, a_r\}$  tel que  $a_1 < a_2 < \dots < a_r$ . Comme  $l_{a_1} = 0$  et  $l_g = 1$ , où  $g$  est le plus grand fermant de  $\pi$ , on définit  $a'_1$  comme le plus petit fermant tel que  $l_{a_1} + 1 = l_{a'_1}$ . Supposons ainsi définis les  $i - 1$  fermants  $a'_1, \dots, a'_{i-1}$  associés avec les  $i$  premiers ouvrants  $a_1, \dots, a_{i-1}$  respectivement. A  $a_i$  on associe le plus petit fermant, soit  $a'_i$ , dans  $\mathcal{F}_s \setminus \{a'_1, \dots, a'_{i-1}\}$  tel que  $l_{a_i} + 1 = l_{a'_i}$ . D'où la bijection.  $\square$

### 3 Quelques lemmes

**Lemme 2** Pour toute partition  $\pi \in \mathcal{P}_n^k$ , on a

$$\text{mak}(\pi) = \text{lmak}'(\pi) = \text{Mil}(\pi) - \text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi).$$

$$\text{mak}'(\pi) = \text{lmak}(\pi) = n(k-1) - \text{Mil}(\pi) - \text{linv } \mathcal{F}(\pi).$$

**Preuve :** On ne démontre que la première partie du lemme car la seconde peut être vérifiée de façon analogue. Soit  $\pi = B_1 - \dots - B_k \in \mathcal{P}_n^k$ , en utilisant les notations de la définition 2, nous pouvons réécrire les statistiques  $\text{lcs}(\pi)$  et  $\text{ros}(\pi)$  de la façon suivante:

$$\begin{aligned} \text{lcs}(\pi) &= \sum_{i=1}^{k-1} \sum_{j>i} (b_j - \text{rinv}(g(B_i), B_j)) . \\ \text{ros}(\pi) &= \sum_{i=2}^k \sum_{j<i} \text{linv}(p(B_i), B_j). \end{aligned}$$

Ainsi, la statistique  $\text{mak}(\pi)$  peut s'écrire :

$$\text{mak}(\pi) = \sum_{i=1}^{k-1} \sum_{j>i} (b_j - \text{rinv}(g(B_i), B_j)) + \sum_{i=2}^k \sum_{j<i} \text{linv}(p(B_i), B_j).$$

$$\begin{aligned}
&= \sum_{j=2}^k (j-1)b_j - \sum_{b \in \mathcal{F}(\pi)} \text{rinv}(b, \pi) + \sum_{b \in \mathcal{O}(\pi)} \text{linv}(b, \pi). \\
&= \text{Mil}(\pi) - \text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi).
\end{aligned}$$

D'autre part, on a:

$$\begin{aligned}
\text{lcb}(\pi) &= \sum_{i=1}^{k-1} \sum_{j>i} \text{rinv}(g(B_i), B_j) = \text{rinv } \mathcal{F}(\pi). \\
\text{rob}(\pi) &= \sum_{i=2}^k \sum_{j<i} [b_j - \text{linv}(p(B_i), B_j)] = \sum_{j=1}^k (k-j)b_j - \text{linv } \mathcal{O}(\pi).
\end{aligned}$$

Et donc la statistique  $\text{lmak}'$  peut s'écrire :

$$\begin{aligned}
\text{lmak}'(\pi) &= [n(k-1) - \sum_{i=1}^k (k-i)b_i] - \text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi) \\
&= [\sum_{i=1}^k (k-1)b_i - \sum_{i=1}^k (k-i)b_i] - \text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi) \\
&= \text{Mil}(\pi) - \text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi).
\end{aligned}$$

Ce qu'il fallait démontrer.  $\square$

Etant donné un sous-ensemble  $\mathcal{O}$  de  $[n]$ , notons  $\mathcal{P}_n^k(\mathcal{O})$  l'ensemble des partitions de  $\mathcal{P}_n^k$  ayant pour l'ensemble des ouvrants  $\mathcal{O}$ . Pour tout  $i \in [k]$  et  $\pi \in \mathcal{P}_n^{k+1}$  on pose

$$\text{stat}_i(\pi) = k - \text{rinv } \mathcal{F}(\pi) - \text{nrinv}(g(B_i), \pi). \quad (4)$$

**Lemme 3** Pour tout  $i \in [k]$  et tout sous-ensemble de  $[n]$  à  $k+1$  éléments  $\mathcal{O}$  fixé, il existe une bijection  $\varphi$  sur  $\mathcal{P}_n^{k+1}(\mathcal{O})$  telle que :

$$\text{stat}_i(\pi) = \text{stat}_{i+1}(\varphi(\pi)) - 1. \quad (5)$$

**Preuve :** Soit  $\pi = B_1 - \dots - B_{k+1} \in \mathcal{P}_n^{k+1}(\mathcal{O})$ . On définit  $\pi' = \phi(\pi) = B'_1 - \dots - B'_{k+1}$  comme suit:  $B'_j = B_j$  pour tout  $j \neq i, i+1$ ,

$$B'_i = \begin{cases} B_i \setminus \{a \in B_i \mid a > g(B_{i+1})\} \cup \{g(B_{i+1})\}, & \text{si } b_{i+1} > 1; \\ B_i \setminus \{a \in B_i \mid a > g(B_{i+1})\}, & \text{si } b_{i+1} = 1; \end{cases}$$

et

$$B'_{i+1} = \begin{cases} B_{i+1} \setminus \{g(B_{i+1})\} \cup \{a \in B_i \mid a > g(B_{i+1})\}, & \text{si } b_{i+1} > 1; \\ B_{i+1} \cup \{a \in B_i \mid a > g(B_{i+1})\}, & \text{si } b_{i+1} = 1; \end{cases}$$

Il est clair que l'on peut construire de manière analogue l'application inverse  $\phi^{-1}$ . Donc  $\phi$  est une bijection. Il reste à vérifier l'équation (5). On distingue trois cas suivants:

1. Si  $b_i = b_{i+1} = 1$ , alors  $\pi' = \pi$ .
2. Si  $b_i > 1$  et  $b_{i+1} = 1$ , alors il est évident que  $\text{rinv } \mathcal{F}(\pi') = \text{rinv } \mathcal{F}(\pi) - 1$ , et  $\text{nrinv}(g(B'_{i+1}), \pi') = \text{nrinv}(g(B_i), \pi)$ .
3. Si  $b_{i+1} > 1$ , on a  $\text{nrinv}(g(B'_{i+1}), \pi') = b_{i+2} + \dots + b_{k+1} - \text{rinv}(g(B'_{i+1}), \pi')$ , et  $\text{rinv } \mathcal{F}(\pi') = \text{rinv } \mathcal{F}(\pi) - \text{rinv}(g(B_i), \pi) + b_{i+1} - 1 + \text{rinv}(g(B'_{i+1}), \pi')$ .

Il est clair que l'égalité (5) a lieu dans les deux premiers cas; Pour le dernier cas, l'égalité (5) découle du fait que  $\text{rinv}(g(B_i), \pi) + \text{nrv}(g(B_i), \pi) = b_{i+1} + \dots + b_{k+1}$ .  $\square$

**Exemple.** Prenons  $i = 3$ ,  $k = 3$ ,  $n = 9$ ,  $\mathcal{O} = \{1, 2, 3, 5\}$  et fixons  $B_1 = \{1, 4, 8\}$  et  $B_2 = \{2\}$ . Alors il y a 8 partitions dans  $P_n^{k+1}(\mathcal{O})$ . Posons  $\pi_0 = 148 - 2 - 3 - 5679$  et  $\pi_j = \varphi(\pi_{j-1})$  pour  $j \geq 1$ , alors l'application de  $\varphi$  donne successivement :

partition	$stat_{i+1} - 1$	$stat_i$
$\pi_0 = 148 - 2 - 3 - 5679$	-3	-6
$\pi_1 = 148 - 2 - 39 - 567$	-6	-5
$\pi_2 = 148 - 2 - 37 - 569$	-5	-5
$\pi_3 = 148 - 2 - 379 - 56$	-5	-4
$\pi_4 = 148 - 2 - 36 - 579$	-4	-5
$\pi_5 = 148 - 2 - 369 - 57$	-5	-4
$\pi_6 = 148 - 2 - 367 - 59$	-4	-4
$\pi_7 = 148 - 2 - 3679 - 5$	-4	-3

On constate que  $\pi_{j+8} = \pi_j$  pour  $j \geq 0$ .

**Lemme 4** Pour tout  $i \in \{1, \dots, k\}$  et tout sous-ensemble de  $[n]$  à  $k+1$  éléments  $\mathcal{O}$  fixé, on a

$$q^i \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{-\text{rinv } \mathcal{F}(\pi)} = \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{\text{stat}_{i+1}(\pi)}. \quad (6)$$

**Preuve :** On montre ce résultat par récurrence décroissante sur  $i$ :

- Pour  $i = k$  le résultat est vrai car  $\text{nrv}(g(B_{i+1})) = 0$ .
- Supposons le résultat vrai à l'ordre  $i$ , alors

$$q^{i-1} \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{-\text{rinv } \mathcal{F}(\pi)} = \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{\text{stat}_{i+1}(\pi)-1}.$$

Or le lemme 3 implique

$$\sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{\text{stat}_{i+1}(\pi)-1} = \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{\text{stat}_i(\pi)}. \quad (7)$$

Ce qui nous permet de conclure.  $\square$

**Lemme 5** Pour tout sous-ensemble de  $[n]$  à  $k+1$  éléments  $\mathcal{O}$  fixé, la statistique Mil + linv est constante sur  $P_n^{k+1}(\mathcal{O})$ . Autrement dit, pour toutes partitions  $\pi, \pi' \in P_n^{k+1}(\mathcal{O})$ , on a

$$\text{Mil}(\pi) + \text{linv } \mathcal{O}(\pi) = \text{Mil}(\pi') + \text{linv } \mathcal{O}(\pi').$$

**Preuve :** Notons  $\pi_0 = B_1 - \dots - B_{k+1}$  la partition de  $P_n^{k+1}(\mathcal{O})$ , telle que tout non-ouvrant  $a$  soit le plus à droite possible, c'est-à-dire,  $a \in B_j$  tel que  $p(B_j) < a < p(B_{j+1})$ . Alors toutes les autres partitions  $\pi$  de  $P_n^{k+1}(\mathcal{O})$  s'obtiennent à partir de  $\pi_0$  par déplacements successifs des non-ouvrants vers la gauche. Or lorsque l'on déplace une lettre vers la gauche (de  $i$  blocs),  $\text{Mil}(\pi_0)$  diminue de  $i$ , et  $\text{linv } \mathcal{O}(\pi_0)$  augmente de  $i$ . Ainsi on montre que  $\text{Mil} + \text{linv}$  est constant sur l'ensemble  $P_n^{k+1}(\mathcal{O})$ .  $\square$

**Lemme 6** *On a*

$$[k]_q \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)} = \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)} \left[ \sum_{i=0}^k q^{k - \text{nrinv}(g(B_{i+1}), \pi)} \right].$$

**Preuve :** Du lemme 4 et l'expression (6) on déduit :

$$q^i \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{-\text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi) + \text{Mil}(\pi)} = \sum_{\pi \in P_{n-1}^{k+1}(\mathcal{O})} q^{\text{stat}_{i+1}(\pi) + \text{linv } \mathcal{O}(\pi) + \text{Mil}(\pi)}.$$

Compte tenu de (4) et le lemme 1, on obtient en sommant sur tous les ouvrants  $\mathcal{O}$  possibles :

$$q^i \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)} = \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi) + k - \text{nrinv}(g(B_{i+1}), \pi)}. \quad (8)$$

D'où le résultat en sommant sur tous les  $i$  possibles.  $\square$

## 4 La fonction génératrice de mak<sub>l</sub>

On est maintenant en mesure de démontrer que la fonction génératrice de mak sur  $\mathcal{P}_n^k$  vérifie la relation de récurrence (1). Il est clair que c'est vrai pour  $n = 1$ . Etant donnée une partition  $\pi \in \mathcal{P}_n^{k+1}$ , on note  $\pi'$  la partition obtenue en supprimant  $n$ . On distingue alors deux cas selon que  $n$  est un singleton ou non.

1.  $n$  est un singleton. Alors  $\pi' \in P_{n-1}^k$  et on vérifie sans peine que

$$\begin{aligned} \text{Mil}(\pi) &= \text{Mil}(\pi') + k. \\ \text{linv } \mathcal{O}(\pi) &= \text{linv } \mathcal{O}(\pi'). \\ \text{rinv } \mathcal{F}(\pi) &= \text{rinv } \mathcal{F}(\pi'). \end{aligned}$$

Ainsi  $\text{mak}(\pi) = \text{mak}(\pi') + k$ . D'où on déduit la fonction génératrice correspondante:

$$q^k \sum_{\pi' \in \mathcal{P}_{n-1}^k} q^{\text{mak}(\pi')}.$$

2.  $n$  n'est pas un singleton. Alors  $\pi' \in P_{n-1}^{k+1}$ . Supposons que  $n$  soit dans le  $i$ ème bloc de  $\pi$ . Alors

$$\begin{aligned} \text{Mil}(\pi) &= \text{Mil}(\pi') + i - 1. \\ \text{linv } \mathcal{O}(\pi) &= \text{linv } \mathcal{O}(\pi') + k + 1 - i. \\ \text{rinv } \mathcal{F}(\pi) &= \text{rinv } \mathcal{F}(\pi') - \text{rinv } (g(B_i), \pi) + b_{i+1} + \cdots + b_{k+1}. \end{aligned}$$

Ainsi, en vertu du lemme 1, on a

$$\text{mak}(\pi) = \text{mak}(\pi') + k - \text{nrinv}(g(B_i), \pi).$$

On en déduit donc, d'après le lemme 5, la fonction génératrice correspondante:

$$\sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)} \left[ \sum_{i=0}^k q^{k - \text{nrinv}(g(B_{i+1}), \pi)} \right] = [k]_q \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)}.$$

En récapitulant les deux cas précédants, on a:

$$\sum_{\pi \in P_n^{k+1}} q^{\text{mak}(\pi)} = q^k \sum_{\pi' \in P_{n-1}^k} q^{\text{mak}(\pi')} + [k]_q \sum_{\pi \in P_{n-1}^{k+1}} q^{\text{mak}(\pi)}.$$

Ce qui est exactement la relation de récurrence (1) pour les  $q$ -nombres de Stirling  $S_q(n, k)$ . Donc la fonction génératrice de mak sur  $\mathcal{P}_n^k$  est  $S_q(n, k)$ .

En récapitulant tout ceux qui précédent, nous avons démontré la conjecture 1. Enfin, en reitant  $j$  ( $j \leq i$ ) fois l'équation (7) on a

$$\sum_{\pi \in \mathcal{P}_n^k(\mathcal{O})} q^{\text{stat}_{i+1}(\pi)} = q^j \sum_{\pi \in \mathcal{P}_n^k(\mathcal{O})} q^{\text{stat}_{i+1-j}(\pi)}.$$

En posant  $l = i - j$ , on en déduit en vertu des lemmes 3 et 4 :

$$\sum_{\pi \in \mathcal{P}_n^k(\mathcal{O})} q^{-\text{rinv } \mathcal{F}(\pi) + \text{linv } \mathcal{O}(\pi) + \text{Mil}(\pi)} = q^{-l} \sum_{\pi \in \mathcal{P}_n^k(\mathcal{O})} q^{\text{stat}_{i+1}(\pi) + \text{linv } \mathcal{O}(\pi) + \text{Mil}(\pi)}.$$

En sommant sur tous les ouvrants  $\mathcal{O}$  possibles on obtient en appliquant (4) et le lemme 2:

$$\sum_{\pi \in \mathcal{P}_n^k} q^{\text{mak}(\pi)} = q^{-l} \sum_{\pi \in \mathcal{P}_n^k} q^{k-1+\text{mak}(\pi)-\text{rinv}(g(B_{l+1}), \pi)}.$$

D'où le théorème 1 en vertu du résultat sur mak plus haut.

## 5 Remarques

La conjecture 1 n'est qu'un cas particulier d'une conjecture plus générale de [7]. Le lecteur intéressé peut se reporter à [7] pour plus de détails. D'autre part, il serait intéressant de savoir s'il existe des extensions analogues  $\text{mak}'_l$  pour  $\text{mak}'$ .

## Références

- [1] De Médicis (A.) and Leroux (P.), *A unified combinatorial approach for  $q$ - (and  $p, q$ -) Stirling numbers*, J. Statistical Planning and Inference, **34** (1993), 89-105.
- [2] Flajolet (Ph.), *Combinatorial aspects of continued fractions*, Disc. Math. **41**(1982), 125-161.
- [3] Foata (D.) and Zeilberger (D.), *Denert's permutation statistic is indeed Euler-Mahonian*, Studies in Appl. Math. 83(1990), 31-59.
- [4] Garsia (A.) et Remmel (J.B.),  *$Q$ -counting rook configurations and a formula of Frobenius*, J. Combin. Theory Ser. A **41** (1986), 246-275.
- [5] Milne (S.), *A  $q$ -analog of restricted growth functions, Dobinsky's equality and Charlier polynomials*, Trans. Amer. Math. Soc., 245(1978), 89-118.
- [6] Sagan (B.), *A maj statistics for set partitions*, European J. Combin. 12 (1991), 69-79.
- [7] Steingrímsson (E.), *Statistics on ordered partitions of sets*, preprint, 1999.
- [8] Steingrímsson (E.), *Personal communication*, juin, 2000.
- [9] Wachs (M.) and White (D.),  *$p, q$ -stirling numbers and set partition statistics*, J. Combin. Theory ser. A 56 (1991), 27-46.

# Identité du groupe du Tas de Sable sur des grilles rectangulaires

Yvan Le Borgne

*Labri, Université Bordeaux 1*

Dominique Rossin

*LIX, École polytechnique*

## Abstract

In 1991, Dhar [6],[7] proves that the recurrent configurations of the sandpile automaton form an abelian group for the addition operator  $\oplus$ . In this article we study the identity element of this group for the sandpile automaton on rectangular grids of size  $p \times q$ . We prove that for  $q \geq p(2 + 3\sqrt{2})/2$ , this identity is made of 3 parts ( $x < \frac{p(2+3\sqrt{2})}{4}, \frac{p(2+3\sqrt{2})}{4} < x < p - \frac{p(2+3\sqrt{2})}{4}, x > p - \frac{p(2+3\sqrt{2})}{4}$ .) Extremal parts are symmetric whereas the central one has 2 grains of sand on every vertex. We give a new method to compute the identity element of the group. This method is twice as fast experimentally as the other known methods.

## Résumé

En 1991, Dhar [6],[7] montre que l'on peut munir les états récurrents de l'automate cellulaire du Tas de Sable d'une structure de groupe abélien fini. Le but de cet article est d'étudier la structure de l'identité de ce groupe dans le cas où l'automate est défini sur une grille rectangulaire de taille  $p \times q$ . Dans cet article, nous montrons que pour  $q \geq p(2 + 3\sqrt{2})/2$  alors il existe une bande 2-uniforme de taille  $p \times (q - 2\frac{p(2+3\sqrt{2})}{4})$  au centre de la configuration. Enfin, nous donnons une nouvelle méthode pour calculer l'identité du groupe. Cette méthode se révèle expérimentalement 2 fois plus rapide que les autres méthodes connues.

## 1 L'automate cellulaire du Tas de Sable

Dans cette partie, nous définissons de manière formelle le modèle du Tas de Sable et nous énonçons les principaux résultats algébriques connus.

Cet automate introduit par Bak, Tang et Wiesenfeld [2] se comporte de manière semblable à différents phénomènes physiques. [1],[3],[8].

L'automate cellulaire est défini dans cet article sur une grille finie ou infinie mais la généralisation à un graphe quelconque de ces définitions est triviale.

### 1.1 Description

Soit une grille à maille carrée dont les cellules sont repérées par un couple  $(x, y) \in [-n..n]^2$ . Chaque cellule  $(i, j)$  est reliée à quatre cellules  $(i + 1, j)$ ,  $(i - 1, j)$ ,  $(i, j + 1)$ ,  $(i, j - 1)$ .

Une configuration  $u = (u_{-n,-n}, u_{-n,-n+1}, \dots, u_{n,n}) \in \mathbb{Z}^{(2n+1)^2}$  est l'assignation de la valeur  $u_{i,j}$  au sommet  $(i, j)$ . Une configuration est dite *positive* si  $u_{i,j} \geq 0$  pour tout  $(i, j) \in \{1 \dots n\}^2$ .

Dans le cas du modèle du Tas de Sable, les sommets du graphe représentent les cellules de l'automate et le nombre  $u_{i,j}$  représente la hauteur de la pile de sable sur le sommet  $(i, j)$ .

Dans la suite de l'article on supposera qu'un sommet ou un ensemble de sommets est distingué : Dans le cas de la grille, on crée un sommet supplémentaire  $\rho$  qui sera le

puits. Les cellules du bord ( $i, j = \pm n$ ) sont reliées par une arête au puits tandis que les 4 coins de la grille sont reliés par deux arêtes au sommet  $\rho$ .

On dira que deux configurations sont égales si le nombre de grains sur chaque sommet est identique sauf éventuellement pour le puits.

Ainsi les configurations pouvant être additionnées ou soustraites, l'espace des configurations est isomorphe à  $\mathbb{Z}^{(2n+1)^2}$ .

La règle de transition de l'automate est la suivante : Si un sommet  $(i, j)$  possède 4 grains de sable au moins, alors ce sommet s'éboule, perdant 4 grains de sable et donne 1 grain à chacun de ses quatre voisins. Le puits, par contre, peut recevoir des grains mais ne peut jamais en donner. Cette règle équivaut à l'addition de la configuration  $(\Delta_{i,j})$  à la configuration  $u$ , où  $(\Delta_{i,j})_{i,j} = -4$  et  $(\Delta_{i,j})_{k,l} = 1$  où  $(k, l)$  est un voisin de  $(i, j)$ . Nous noterons  $u \rightarrow v$  s'il existe  $(i, j)$  tel que  $v = u + \Delta_{i,j}$ .

La clôture transitive de  $\rightarrow$  est noté  $\xrightarrow{*}$ . Ainsi on dira que  $u \xrightarrow{*} v$  si on peut passer de  $u$  à  $v$  par une suite d'éboulements.

La clôture symétrique de  $\xrightarrow{*}$  sera notée  $\equiv$ . Ainsi  $u \equiv v$  si et seulement s'il existe une suite de configurations  $w_1 = u, \dots, w_k = v$  telle que pour tout  $i$  de  $\{1, \dots, k-1\}$  on ait soit  $w_i \rightarrow w_{i+1}$  soit  $w_{i+1} \rightarrow w_i$ .

Grâce à la connexité du graphe et l'interdiction d'ébouler le puits, on peut montrer que si  $u$  est une configuration positive alors il existe une série d'éboulements partant de  $u$  et conduisant à une configuration *stable* où tous les sommets ont un nombre de grains strictement inférieur à 4. De plus la configuration stable obtenue ne dépend pas de l'ordre choisi pour réaliser les éboulements.

Nous noterons par  $\hat{u}$  la configuration stable obtenue après éboulement de la configuration  $u$ .

## 1.2 Configurations récurrentes

Le concept de configuration récurrente est important dans ce modèle, car ce sont ces configurations qui interviennent lors de l'étude des différents phénomènes critiques.

Une configuration est dite *récurrente* si elle apparaît un nombre infini de fois lors de l'évolution d'un système. Cette évolution nous est donnée ici par une chaîne de Markov. Soit  $u$  une configuration positive stable. On répète les étapes suivantes :

1. On choisit un sommet  $(i, j)$  du graphe aléatoirement
2. On ajoute 1 sur le sommet  $(i, j)$  dans la configuration  $u$
3. On éboule la nouvelle configuration pour obtenir une nouvelle configuration stable  $u'$

On obtient ainsi une suite de configurations stables. Dans ces configurations, certaines apparaissent un nombre fini de fois, on les appellera configurations *transientes*, d'autres un nombre infini de fois, elles seront dites *récurrentes*.

Cette relation peut se traduire d'une manière plus intrinsèque aux configurations.

**Définition 1.** Une configuration  $u$  est dite récurrente si elle est positive, stable et s'il existe une configuration positive  $v \neq 0$  telle que  $u + v \xrightarrow{*} u$ .

On définit l'opérateur  $\oplus$  sur les configurations positives de la manière suivante :

**Définition 2.** Soient  $u$  et  $v$  deux configurations positives. On notera  $u \oplus v$  la configuration  $w$  telle que  $w = \widehat{u+v}$ .

**Proposition 1.** ([7]) L'ensemble des configurations récurrentes muni de l'opérateur  $\oplus$  forme un groupe abélien fini.

Une autre manière de voir les configurations récurrentes est de considérer la clôture symétrique  $\equiv$ . En effet, si l'on considère les classes d'équivalence de cette clôture alors on peut montrer le résultat suivant :

**Proposition 2.** [4] *Dans chaque classe d'équivalence il existe une et une seule configuration récurrente.*

De plus les classes d'équivalence peuvent être vues comme le quotient

$$\mathbb{Z}^{(2n+1)^2}/\{\Delta_{1,1}, \dots, \Delta_{2n+1,2n+1}\}$$

## 2 Éboulement d'une pile de sable sur une grille uniforme

Dans cette partie, nous travaillons sur une grille infinie dont chaque case sera numérotée par  $(i, j) \in \mathbb{Z}^2$  où  $i$  est l'abscisse et  $j$  l'ordonnée de la case. Il n'y a donc pas de puits dans cet automate.

Par raison de symétrie, nous travaillerons la plupart du temps sur un huitième de plan  $\Pi$  défini par les cellules  $(i, j)$ ,  $i \geq 0, j \geq 0, i \geq j$ .

Nous étudions dans cette section la chute d'une pile de  $n$  grains sur la case  $(0, 0)$  d'une grille infinie uniforme remplie de 0. Remarquons tout d'abord que l'éboulement d'une telle pile est fini.

Nous utilisons l'algorithme parallèle d'éboulement à savoir qu'à chaque étape  $t$  on éboule tous les sommets instables en parallèle le nombre maximal de fois : Par exemple si un sommet possède 9 grains de sable, alors on l'éboule 2 fois donnant 2 grains à chacun de ses voisins et gardant 1 grain.

### 2.1 Étude de la forme du champ d'éboulement

$u_{i,j}(t)$  est le nombre de grains sur la cellule  $(i, j)$  à l'instant  $t$ .

Nous noterons  $n_{i,j}(t)$  le nombre de grains de sable tombés dans la cellule  $(i, j)$  à l'instant  $t$  et par  $c_{i,j}(t)$  le nombre de grains de sable qui sont tombés dans la cellule  $(i, j)$  entre l'instant 0 et l'instant  $t$ . Par raison de symétrie, les suites sont définies sur les cellules de  $\Pi$ .

**Remarque 1.**

$$c_{i,j}(t) = \sum_{\tau=0}^t n_{i,j}(\tau) \quad (1)$$

$$c_{i,j}(t+1) = \lfloor \frac{c_{i+1,j}(t)}{4} \rfloor + \lfloor \frac{c_{i-1,j}(t)}{4} \rfloor + \lfloor \frac{c_{i,j+1}(t)}{4} \rfloor + \lfloor \frac{c_{i,j-1}(t)}{4} \rfloor \quad (2)$$

$$n_{i,j}(t) = \lfloor \frac{u_{i+1,j}(t)}{4} \rfloor + \lfloor \frac{u_{i-1,j}(t)}{4} \rfloor + \lfloor \frac{u_{i,j+1}(t)}{4} \rfloor + \lfloor \frac{u_{i,j-1}(t)}{4} \rfloor \quad (3)$$

$$u_{i,j}(t+1) = u_{i,j}(t) - 4 \lfloor \frac{u_{i,j}(t)}{4} \rfloor + \sum_{v \text{ voisin de } (i,j)} \lfloor \frac{u_v(t)}{4} \rfloor \quad (4)$$

**Théorème 1.** *Pour tout  $t, l, m$  les suites  $(c_{l,2i+m})_{i \in \mathbb{N}}, (c_{2i+m,l})_{i \in \mathbb{N}}, (c_{l+i,l-i})_{i \in \mathbb{N}}$  et  $(c_{l+i,i})_{i \in \mathbb{N}}$  restreintes à  $\Pi$  sont décroissantes.*

*Démonstration.* Le résultat est vrai pour  $t = 1$ . Supposons maintenant qu'il est vrai pour  $t \leq t_0$ .

La première partie de la démonstration consiste à regarder le cas des deux dernières suites.

Soient deux cellules voisines  $a$  et  $b$  correspondant à des termes consécutifs dans une des suites - par exemple  $(l+i, l-i)$  et  $(l+i+1, l-i-1)$ . Pour  $t = t_0$ , on a  $c_{l+i, l-i}(t_0) \geq c_{l+i+1, l-i-1}(t_0)$ . Pour montrer cette relation pour  $t_0 + 1$  on met en correspondance par une bijection  $\phi$  les voisins de la première cellule avec ceux de la deuxième cellule de sorte que

$$c_v(t_0) \geq c_{\phi(v)}(t_0) \quad (5)$$

pour tout  $v$  voisin de la cellule  $a$ .

Dans le cas de la suite  $c_{l+i, l-i}$  la bijection  $\phi$  est définie par :

$$\phi(x, y) = (x + 1, y - 1)$$

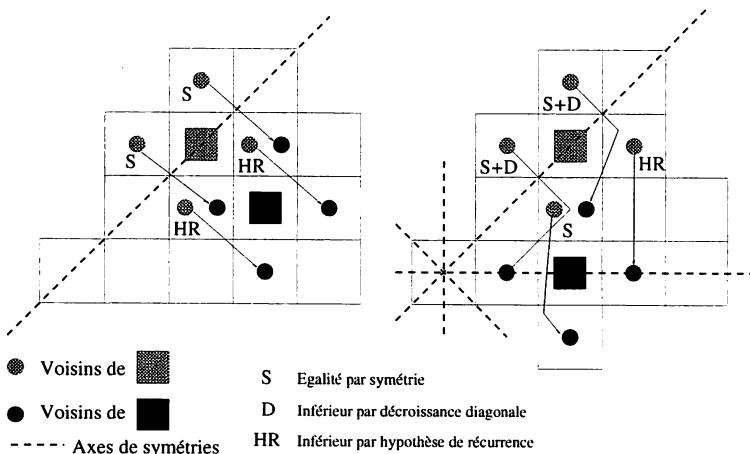
pour les cellules  $(x, y)$  strictement à l'intérieur de  $\Pi$ .

Pour les cellules situées au bord de  $\Pi$  il faut tenir compte des symétries du problème comme décrit sur la figure ci-dessous.

Ainsi, par l'égalité (2) et l'inégalité (5), on déduit l'inégalité pour  $t_0 + 1$ .

La figure suivante décrit certains cas particuliers de la bijection  $\phi$ .

Dans ces figures, les flèches indiquent les images par la fonction  $\phi$ , leur étiquetage la raison de la décroissance ou de l'égalité des suites. Une flèche étiquetée par  $S$  indique que les cellules correspondantes ont la même valeur de  $c$  par symétrie du problème. L'étiquette  $HR$  ou  $D$  indique que l'on a décroissance par l'hypothèse de récurrence du théorème 1 à l'instant  $t$ .



Notons que pour les suites horizontales et verticales, nous avons montré la décroissance pour des cases éloignées de 2. En effet, si l'on considère plutôt des cases adjacentes alors il n'existe pas de fonction  $\phi$  vérifiant les propriétés requises. Néanmoins il apparaît expérimentalement que le théorème est vrai pour les suites  $(c_{l,i})_{i \in \mathbb{N}}$  et  $(c_{i,l})_{i \in \mathbb{N}}$ .

□

## 2.2 Rayon d'éboulement sur une grille vide

Soit une grille infinie et une configuration  $u$  telle que  $u_{i,j} = 0$  pour tout  $(i,j) \neq (0,0)$  et  $u_{0,0} = n$ . On définit le support de l'éboulement par

$$S_n = \{(i,j) \in \mathbb{Z}^2, \exists t \in \mathbb{N}, n_{i,j}(t) > 0\}$$

Nous allons ici borner ce support à l'intérieur de deux boules centrées à l'origine. Nous définissons ainsi :

$$I_r = \{(x,y), |x| + |y| \leq r\}$$

$$B_r = \{(x,y), |x| \leq r, |y| \leq r\}$$

Remarquons que  $I_r$  est la boule de rayon  $r$  pour la distance de Manhattan et  $B_r$  la boule de rayon  $r$  pour la norme infinie.

**Théorème 2.** *Il existe  $r_0$  fini tel que :*

$$I_{r_0-1} \subseteq S_n \subseteq B_{r_0}$$

*On appellera  $r_0$  le rayon d'éboulement de cette configuration.*

*Démonstration.* On appellera cellule *active* une cellule qui s'est éboulée pendant la relaxation de la pile de  $n$  grains.

**Lemme 1.** *Si  $(i,j) \in \Pi$  est active alors  $(i-1,j)$  est active.*

*Démonstration.* Si  $(i,j)$  active alors par le théorème 1  $(i-2,j)$  est active. Par suite  $(i-1,j+1)$  est active. Enfin,  $(i-1,j-1)$  est donc active.

Par conséquence comme les 4 voisins de la cellule  $(i-1,j)$  sont actifs cette cellule est aussi active.  $\square$

Soit  $r_0$  la distance maximale (pour la norme infinie) des cellules actives au centre de la grille.

Par définition de  $r_0$  l'inclusion  $S_n \subseteq B(0, r_0)$  est évidente.

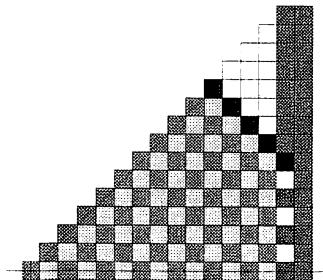
Comme l'éboulement de cette configuration est fini, on note  $(i,j)$  une cellule active à distance maximale  $r_0$  de l'origine. Par symétrie on suppose que cette cellule est située dans  $\Pi$ .

Par le théorème 1, toutes les cellules  $(i',j') \in \Pi$  avec  $i' + j' = i + j$  sont aussi actives. Par le lemme 1 on déduit que les cellules  $(i',j') \in \Pi$  avec  $i' < i$  et  $i' + j' + 1 = i + j$  sont actives.

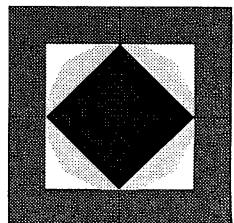
Par  $(0, -2)$  croissance et  $(-1, -1)$  croissance on déduit que toutes les cellules  $(i',j') \in \Pi$  avec  $i' < i$  et  $i' + j' \leq i + j$  sont actives.

Par symétrie du problème on déduit que toutes les cellules situées dans un octogone sont actives donc à fortiori celles de  $I_{r_0-1}$ .

La figure ci-dessous montre les encadrements précédents.



- Distance maximale du centre
- Certainement en dehors du support
- (0,2)-décroissance
- (1,-1)-décroissance
- (1,1)-décroissance
- Tout les voisins éboulés
- Appartenance indéterminée.



- Pas de support dans cette zone
- Inclu dans le support
- + ■ Un support possible

□

Muni de cet encadrement nous pouvons maintenant trouver une majoration du rayon d'éboulement  $r_0$ . Pour cela nous admettrons le lemme suivant :

**Lemme 2.** Soit une configuration  $u$ . Soit  $S$  l'ensemble des cellules actives lors du calcul de l'éboulement de la configuration  $u$  en  $\hat{u}$ .

Alors la configuration  $\hat{u}$  restreinte aux cellules de  $S$  est récurrente.

Ainsi, la configuration restreinte à  $I_r$  est récurrente. Or le nombre minimal de grains de sable sur une configuration récurrente est au moins égal au nombre d'arêtes internes du graphe. Or, dans  $I_r$ , il y a  $2r(r-1)$  sommets internes donc  $4r(r-1)$  arêtes internes. Donc on a  $4r(r-1) \leq n$ , le nombre de grains de sable initialement sur la grille, d'où la proposition suivante :

**Proposition 3.** Le rayon d'éboulement sur une grille 0-uniforme satisfait :

$$r_0 \leq \sqrt{n}/2$$

## 2.3 Extension aux cas des grilles 1 et 2-uniformes

Le résultat précédent peut s'étendre assez facilement au cas de la grille 1 ou 2-uniforme. On obtient les majorations suivantes :

**Théorème 3.**

$$r_0 \leq \frac{1}{2}\sqrt{n}$$

$$r_1 \leq \frac{3}{4}\sqrt{n}$$

$$r_2 \leq \frac{3}{2}\sqrt{n}$$

**Remarque 2.** Expérimentalement on obtient les approximations suivantes :

$$r_0 \approx 0.383\sqrt{n}$$

$$r_1 \approx 0.484\sqrt{n}$$

$$r_2 \approx 0.800\sqrt{n}$$

*Démonstration.* Considérons par exemple le cas d'une grille 1-uniforme avec une pile de  $n$  grains située en  $(0, 0)$ . Pour calculer la configuration stable obtenue après éboulement, nous utiliserons l'algorithme suivant :

1. Ébouler la configuration formée d'une pile de  $n - 1$  grains sur une grille 0-uniforme.
2. Ajouter 1 grain sur toutes les cellules
3. ébouler cette nouvelle configuration

Le premier point de cet algorithme a été traité précédemment et conduit à une configuration stable dont le rayon est majoré par  $\sqrt{n}/2$ .

On peut majorer cette configuration stable par la configuration  $us$  suivante :

$$\begin{cases} us_{i,j} = 3 \text{ si } i \leq r_0 \text{ et } j \leq r_0 \\ us_{i,j} = 0 \text{ sinon} \end{cases}$$

Pour cette démonstration nous allons coder une configuration  $u$  par la suite  $(v_i)_{i \in \mathbb{N}}$  où

$$v_i = \max\{\{u_{x,y}, (x = \pm i \text{ et } |y| \leq i)\} \quad (6)$$

$$\bigcup\{u_{x,y}, (y = \pm i \text{ et } |x| \leq i)\} \quad (7)$$

Ainsi la configuration  $us$  peut être représentée par  $\underbrace{333 \dots 3}_{r_0} 000 \dots$ . Après la deuxième phase de l'algorithme on obtient la configuration  $\underbrace{444 \dots 4}_{r_0} \underbrace{1111 \dots}_{3r_0/2}$ . On peut démontrer très simplement les règles de réécriture suivantes :

- $41 \rightarrow 32, 42 \rightarrow 33$
- $43 \dots 32 \rightarrow 43 \dots 33$
- $43 \dots 31 \rightarrow 43 \dots 32$

Ces règles sont une conséquence directe des règles d'éboulement du Tas de Sable et de la majoration de la valeur d'un sommet par une valeur plus élevée. Cette majoration intervient pour le cas des coins des carrés centrés en  $(0, 0)$ .

Grâce à ces règles, on peut écrire le mot associé à  $us$  en  $\underbrace{3 \dots 3}_{3r_0/2} 11 \dots$  démontrant ainsi la deuxième majoration. Le cas de la dernière majoration est identique au précédent si l'on ajoute quelques règles de réécriture.

□

### 3 Identité sur la grille rectangulaire

Dans cette partie, nous allons donner les différentes méthodes de calcul de l'identité puis nous en adapterons une pour donner une nouvelle caractérisation de cette configuration. Cette caractérisation se traduira par un nouvel algorithme rapide de calcul de l'identité.

#### 3.1 Calcul direct de l'identité

Dhar montre que l'identité du groupe peut s'écrire comme une somme de trois configurations de la manière suivante :

### Proposition 4.

$$Id = \delta \oplus \overline{\delta \oplus \delta}$$

où  $\bar{u}$  est la configuration  $v$  telle que  $v_i = 4 - 1 - u_i$ .

## 3.2 Calcul par classe d'équivalence

Notons  $C_{Id}$  la classe de l'identité, celle-ci contient la configuration 0 ainsi que la configuration  $\beta$  tel que  $\beta_{i,j}$  est le nombre d'arêtes reliant le sommet  $(i, j)$  au puits.

De proche en proche, on peut montrer que  $k\beta \in C_{Id}$  avec  $k$  arbitrairement grand. On peut montrer qu'à partir d'un certain  $k$ , la configuration  $\widehat{k\beta}$  est récurrente, stationnaire et dans  $C_{Id}$ . Ceci donne lieu à un algorithme de calcul de l'identité appelé algorithme de combustion.

## 3.3 Calcul sur une grille rectangulaire

Nous restreignons maintenant notre discussion au cas de la grille rectangulaire de taille  $p \times q$ . Nous prendrons  $q \geq (p(2 + 3\sqrt{2})/2)$ .

Nous allons montrer dans cette partie le théorème suivant :

**Théorème 4.** *L'identité sur une grille rectangulaire est égale par  $\hat{D}$  où  $D$  est la configuration suivante :*

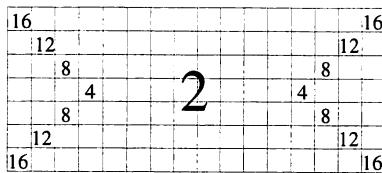


FIG. 1: Configuration  $D$

*Démonstration.* Nous allons définir les configurations  $D^{(i)}$  de la manière suivante :

- Sur un sommet à distance  $i$  du puits il y a 1 grain de sable.
- $D_{j,j}^{(i)} = 2$  si  $j \leq i$
- $D_{n-j,j}^{(i)} = 2$  si  $j \leq i$
- $D_{j,p-j}^{(i)} = 2$  si  $j \leq i$
- $D_{n-j,p-j}^{(i)} = 2$  si  $j \leq i$

Les premières valeurs de  $D^{(i)}$  sont égales à :

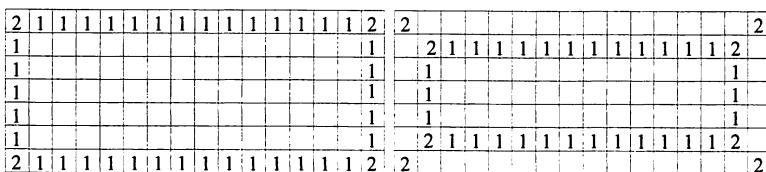


FIG. 2: Configurations  $D^{(1)}$  et  $D^{(2)}$

On remarque que  $D^{(1)} = \beta$  donc  $D^{(1)} \in C_{Id}$ .

**Proposition 5.** Les configurations  $D^{(i)}$  sont équivalentes à l'identité.

*Démonstration.* Supposons que la proposition soit vraie pour  $D^{(1)}, D^{(2)}, \dots, D^{(l)}$ . Considérons maintenant la configuration  $\lambda = D^{(1)} + D^{(l)}$  et forçons l'éboulement des sommets à distance plus petite ou égale à  $l$ . On obtient  $D^{(l+1)}$ . Comme  $\lambda \in C_{Id}$  on a  $D^{(l+1)} \in C_{Id}$ .  $\square$

La fin de la preuve du théorème 4 découle du fait que

$$D = \sum_{i=1}^{p/2} 2D^{(i)}$$

$\square$

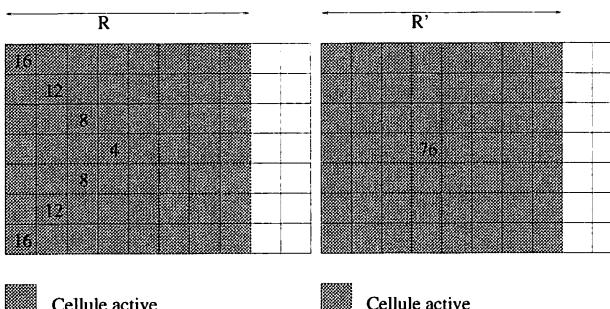
La configuration  $D$  est telle que  $D_i \geq 2$  pour tout  $i$ . Or la configuration 2-uniforme est récurrente, il en est donc de même pour  $\hat{D}$ .

Pour démontrer l'existence d'une bande de cellule ayant pour valeur 2 au centre de l'identité nous allons montrer que l'éboulement des cellules instables de  $D$  ne se propage pas jusqu'au centre.

Restreignons notre discussion à une moitié de la grille  $(p \times q/2)$ . Par symétrie, le résultat s'étend à l'ensemble de la grille.

Il est possible de traiter l'éboulement de ces sommets par le même mécanisme que l'éboulement d'une pile de grains sur une grille uniforme.

Une autre méthode consiste à considérer les deux rayons suivants :



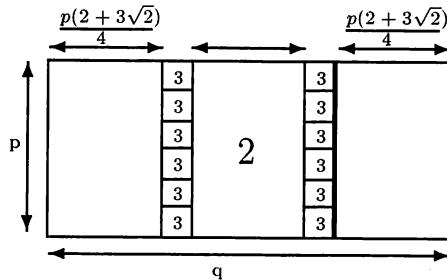
$R$  est la distance à laquelle va s'ébouler la configuration  $D$  tandis que  $R'$  est la distance à laquelle va s'ébouler la configuration formée par une pile unique sur une grille 2-uniforme. La hauteur de cette pile est la même que la somme des valeurs de tous les sommets instables de  $D$ .

En conjecturant que  $R' \geq R$ , on peut utiliser les encadrements trouvés dans le théorème 3 pour majorer le rayon  $R$ .

Cette conjecture est basé sur des intuitions provenant du phénomène physique modélisé. Nous avons une preuve plus complète et rigoureuse qu'il serait trop long d'exposer ici.

La pile de grains ayant ici une hauteur de  $p^2/2$ , on en déduit le théorème suivant :

**Théorème 5.** L'identité sur une grille rectangulaire de côtés  $p$  par  $q$  ( $q \geq \frac{p(2+3\sqrt{2})}{2}$ ) est formée de deux bords symétriques l'un de l'autre de taille  $\frac{p(2+3\sqrt{2})}{4} \times p$  et d'une bande centrale 2-uniforme de taille  $q - 2\frac{p(2+3\sqrt{2})}{4}$  bordée de deux colonnes formées de trois grains dans chaque case.



*Démonstration.* Il reste à démontrer l'existence des deux colonnes de 3 à la bordure de la zone d'éboulement.

Notons  $A$  l'ensemble des cellules actives lors de l'éboulement de  $D$ . Soit une cellule extérieure à  $A$ . Cette cellule avait deux grains à l'origine dans la configuration  $D$ . Comme cette cellule ne s'est pas éboulée elle n'a pas perdu de grains pendant le calcul de  $\hat{D}$ .

Cette cellule a donc gagné au maximum un grain de sable. Cela veut dire qu'au plus 1 voisin de cette cellule est actif.

Cela implique que la frontière des cellules actives est une ligne. Dans ce cas les cellules non actives voisines de la frontière ont gagné un grain donc en possèdent 3.  $\square$

## 4 Conclusion

Nous avons étudié ici l'éboulement d'une pile de sable de hauteur  $n$  dans le modèle en dimension 2. Ainsi, nous avons prouvé le fait que l'on obtient un support d'aire en  $\mathcal{O}(\sqrt{n})$  comme observé expérimentalement par D. Dhar [5].

De plus l'algorithme de calcul de l'identité du groupe que nous proposons semble plus performant que les algorithmes connus -par addition ou algorithme de combustion- comme l'attestent les résultats expérimentaux suivants.

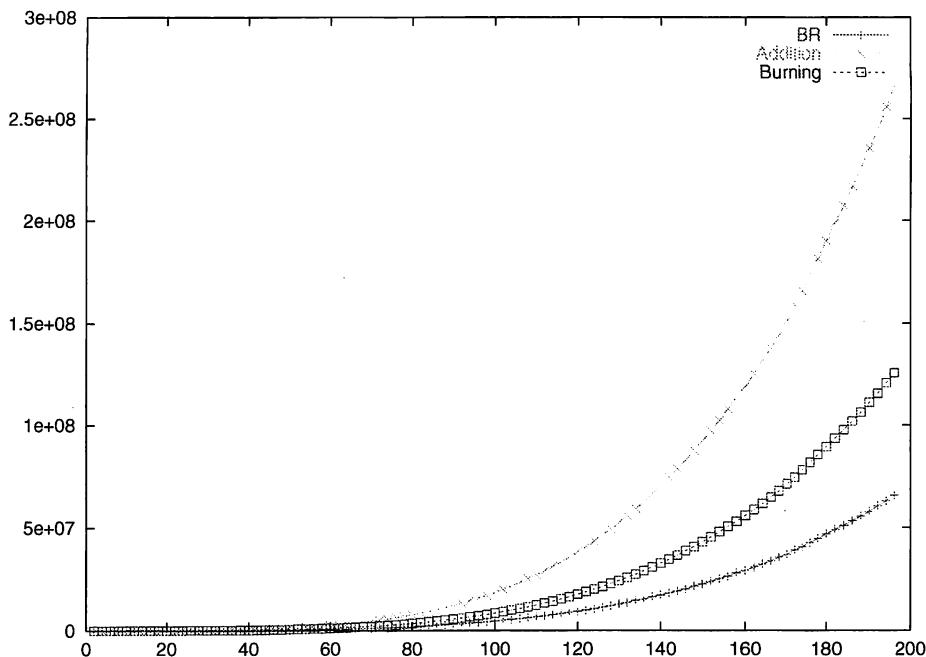
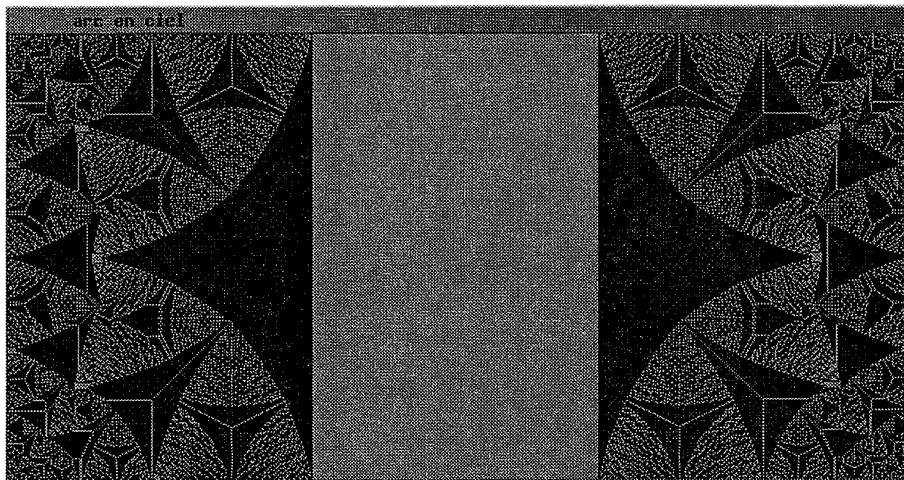


FIG. 3: Nombre d'éboulements dans le calcul de l'identité

L'algorithme nommé *BR* représente l'algorithme basé sur l'éboulement de la configuration *D*.

Enfin, nous avons pu caractériser la forme de la configuration identité pour une grille rectangulaire.

La forme de l'identité sur la grille carrée reste encore à ce jour une question ouverte.



## Références

- [1] P. Bak and K. Chen. The physics of fractals. *Physica D*, 38 :5–12, 1989.
- [2] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality : An explanation of  $1/f$  noise. *Physical Review Letters*, 59(4) :381–384, July 1987.
- [3] K. Chen, P. Bak, and S.P. Obukhov. Self-organized criticality in a crack propagation model of earthquakes. *Phys. Rev. A*, 43 :625–30, 1991.
- [4] R. Cori and D. Rossin. On the sandpile group of a graph. *to appear in European Journal of Combinatorics*, 2000.
- [5] D. Dhar. Studying self organized criticality with exactly solved models. Cours donné à l’École Polytechnique de Lausanne. Disponible sur la page www de l’auteur.
- [6] D. Dhar. Self-organized critical state of sandpile automaton models. *Physical Review Letters*, 64(14) :1613–1616, 1990.
- [7] D. Dhar, P. Ruelle, S. Sen, and D. Verma. Algebraic aspects of abelian sandpile models. *Journal of Physics A*, 28 :805–831, 1995.
- [8] A. Sornette and D. Sornette. Self-organised criticality of earthquakes. *Europhys. Lett.*, 9 :197–202, 1989.

# Optimal Performance Comparisons for Massively Parallel Multiprocessors

Håkan Lennerstad\*  
 Dept. of Mathematics  
 Univ. of Karlskrona/Ronneby  
 S-371 79 Karlskrona

Lars Lundberg\*  
 Dept. of Computer Science  
 Univ. of Karlskrona/Ronneby  
 S-372 25 Ronneby

## Abstract

We compare the performance of a system consisting of one large cluster containing  $q$  processors with a system where processors are grouped into  $k$  clusters containing  $u$  processors each. A parallel program, consisting of any number of processes and of any set of synchronization signals except deadlock, is executed on this system. Processes may be relocated between the processors within a cluster only. We have previously calculated an explicit formula for a function  $G(k, u, q)$  representing the worst case ratio of the completion times in the two cases. Provided that we can find optimal or near optimal schedules,  $G(k, u, q)$  gives an optimal upper bound on the gain of using one cluster containing  $q$  processors compared to using  $k$  clusters containing  $u$  processors each. However, the complexity of this formula is exponential in the variables  $k$  and  $q$ . Hence the formula can be used only for moderately sized multiprocessors.

In this paper we present an iterative technique allowing the calculation of the values of  $G(k, u, q)$  in polynomial time. The computational complexity is logarithmic in  $k$ , cubic in  $q$  and independent of  $u$ . This allows optimal comparisons of the performance of massively parallel multiprocessors. One immediate practical consequence is that one should avoid building large systems with clusters of unit size. Increasing the cluster size to 4 or 8 processors improves the worst-case behavior significantly.

## Résumé

Nous comparons un système contenant un grand cluster composé de  $q$  processeurs avec un système dont les processeurs sont groupés en  $k$  clusters ayant chacun  $u$  processeurs. Un programme parallèle, constitué d'un nombre arbitraire de processus ainsi que d'un ensemble arbitraire de contraintes de synchronisation sans deadlocks, est exécuté sur ce système. On permet la relocalisation des processus à condition qu'elle s'effectue entre processeurs d'un même cluster. Nous avons déjà donné une formule explicite pour une fonction  $G(k, u, q)$  représentant le rapport en pire cas des temps de complétion pour les deux cas. En supposant qu'on puisse trouver des ordonnancements optimaux ou presqu'optimaux,  $G(k, u, q)$  donne une borne supérieure optimale pour le gain réalisé par l'utilisation d'un seul cluster composé de  $q$  processeurs par rapport à l'utilisation de  $k$  clusters composés de  $u$  processeurs chacun. La complexité de cette formule est exponentielle en les variables  $k$  et  $q$ , ce qui la rend utilisable uniquement pour les multiprocesseurs de

---

\*Both authors were supported in part by the Blekinge Research Foundation.

taille modeste. Dans cet article, nous exposons une technique itérative permettant le calcul de  $G(k, u, q)$  en temps polynomial. La complexité du calcul est logarithmique en  $k$ , cubique en  $q$  et indépendante de  $u$ . Ceci permet des comparaisons optimales de la performance des multiprocesseurs massivement parallèles. Une conséquence pratique immédiate est que l'on doit éviter de construire des systèmes de grande taille avec des clusters de taille 1. L'augmentation de la taille des clusters à 4 ou 8 processeurs améliore significativement la performance en pire cas.

## 1 Introduction

It is relatively easy and cheap to build large multiprocessors, i.e. systems with a large number of processors, by connecting a number of small clusters with a communication network. The MPI (Message Passing Interface) [16] and PVM (Parallel Virtual Machine) [2] environments make it possible to write a parallel program which runs on a number of clusters. Some systems also offer a distributed shared memory model over a number of clusters, e.g. the Sun WildFire system [4]. Systems which consist of a number of clusters can easily scale up, i.e. one can simply connect more clusters to the communication network. Another advantage of using multiple clusters is that communication within one cluster does not interfere with communication within other clusters. An alternative to multiple small clusters is one large cluster. The processors in a cluster are often connected with a shared bus, which becomes a bottleneck if the number of processors is large. Consequently, systems with multiple small clusters have a number of advantages compared to systems consisting of one large cluster.

A major disadvantage with multiple clusters is that, since processes may not be reallocated between clusters, some clusters may be idle while others are very busy. In most cases the time when processors are idle can be reduced by careful allocation of processes to clusters.

The problem of finding schedules for minimum completion time is NP-hard [1]. However, there are heuristic scheduling techniques which result in near optimal schedules [5], [17]. There are previous results [6], [8] which make it possible to put tight bounds on the maximum performance gain of using one large cluster compared to multiple small clusters, provided that we are able to find optimal or near optimal schedules. A serious limitation of the previous result was that, due to the complexity in the formulas, we could only provide figures for configurations with up to 30-50 processors since the complexity grew exponentially with the number of processors. In this paper we present an iterative technique for calculating these formulas in polynomial time, thus making it possible to provide results for massively parallel multiprocessors.

## 2 Previous results and overview over the report

The report [8] extends the results in [6]. In [6] the formula for the function  $g$  in the case  $u = 1, k = q$  is treated only, representing static versus dynamic allocation on the same multiprocessor. Furthermore the result on  $G$  (see the next section) is new in [8].

The mathematics of the subject is focused in the reports [6] and [8]. These two reports can be viewed as the theoretical base for the reports [7], [12] and [13], which treat different applications from a computer science point of view.

Other than the reports [6] to [9], the only general results concerning allocation strategies of parallel programs where synchronization is not neglected appear to be the

results by R. L. Graham [3]. The overhead for process reallocation and synchronization is neglected also in this work. Here so called list scheduling algorithms are considered. This term is used for dynamic allocation algorithms where, when a processor becomes idle and there are waiting executable processes, one of the executable processes is immediately allocated to the idle processor. It is established in [3] that the makespan for a program allocated with a list scheduling algorithm is never higher than two times the makespan with optimal dynamic allocation.

We next give an overview of the paper.

In Section 3 the parallel program and multiprocessor model is specified in detail, and the definitions of the performance functions are given.

In Section 4 the allocation problem is described and analyzed in detail, and transformed to a mathematical problem. In Section 5 the problem is fully formulated as a combinatorial problem, and we briefly present the previously derived formulas and related properties. Our main reference for previous results is [8].

In Section 6 we state and prove the main results: a recursive polynomial time calculation method. The paper is concluded with a table for large values and a C program for the algorithms.

### 3 Programs, multiprocessors and performance functions

A parallel program consists of a set of sequential processes. A process in a parallel program can be either Blocked, Ready or Running. Processes in the Ready and Running states are referred to as *active* processes. The execution of a process is controlled by two synchronization primitives: *Wait(Event)* and *Activate(Event)*, where Event couples a certain Activate to a certain Wait. When a process executes an Activate on an event, we say that the event has occurred. If a process executes a Wait on an event which has not yet occurred, that process becomes blocked until another process executes an Activate on the same event. However, a process executing a Wait on an event which already has occurred does not become blocked. Each process can be represented as a list of sequential segments, which are separated by a Wait or an Activate. We assume that, for each process, the length and order the sequential segments are independent of the way processes are scheduled. All processes are created at the start of the execution. Some processes may, however, be initially blocked by a Wait.

We consider parallel programs consisting of a finite set of processes, some of which may run in parallel, and a finite set of synchronization signals between the processes, which introduce dependencies between the processes.

Thus, an arbitrary parallel program of  $n$  processes can be described as a set of  $n$  rational numbers, describing the total execution time of each process, together with a set of synchronization signals. This set consists of quadruplets,  $(i, j, x_i, x_j)$  - denoting the sending process  $i$ , the receiving process  $j$ , the time point  $x_i$  where the signal is sent and the time point  $x_j$  where the signal is received. The time points are measured as local time of the processors, i.e. disregarding idle periods. The significance of the synchronizing signals is that process  $j$  may not execute past the point  $x_i$  if processor  $i$  has not reached the time point  $x_i$ , in that case it is blocked. Our only condition on the synchronization is that the program is executable - it does not contain deadlock.

We thus have no restrictions on the numbers denoting processing times of the process parts or synchronization time points. The only requirement is that all numbers are

rational numbers, which from a practical point of view is a weak restriction.

The processors are assumed to have identical properties. In cluster allocation the processors are organized in  $k$  groups, the clusters, where each cluster contains  $u$  processors. Once a process is allocated to a processor in a specific cluster, it can only be executed on a processor in this cluster. It may be transferred any time, but only to a processor in the same cluster. The cost of transferring processes is neglected.

Dynamic and static allocations are both special cases of cluster allocation. Dynamic allocation represents the case of having all processors in one cluster,  $k = 1$ , hence the processes may be transferred between all processors without limitations. Static allocation can be described as the case when each cluster has one single processor,  $u = 1$ . Thus, in the case of static allocation, a process may never be transferred from the processor where it was initiated.

For a parallel program  $P$ , there is a certain number of different cluster allocations. The number of different cluster allocations is very large even for moderate values of  $n$  and  $k$ , however certainly finite.

Since we have a well defined minimal makespan for each allocation, the set of minimal makespans of the program  $P$  for all cluster allocations is finite, so it has a minimum. An allocation which results in a minimal makespan, i.e. shorter than or equal to the makespan of any other allocation, is called an optimal allocation.

For any fixed parallel program  $P$ , we compare the minimal makespans of  $P$  for two multiprocessors.

The first multiprocessor has  $q$  processors and allocates parallel programs dynamically. For a parallel program  $P$ , we denote the makespan for  $P$  with optimal dynamic allocation by  $T_d(P, q)$ .

The second multiprocessor performs cluster allocation with  $k$  clusters and  $u$  processors in each cluster. Let  $T_{c,A}(P, k, u)$  denote the makespan for the parallel program  $P$  with the cluster allocation  $A$ . The makespan for the parallel program  $P$  with optimal cluster allocation is denoted by  $T_c(P, k, u)$ . Hence,  $T_c(P, k, u) = \min T_{c,A}(P, k, u)$ , where the minimum is taken over all possible cluster allocations  $A$ .

We now define the performance function  $g$ :

$$g(n, k, u, q) = \max_{\text{all } n\text{-programs } P} \frac{T_c(P, k, u)}{T_d(P, q)}.$$

Here  $n$ -programs denotes parallel programs with  $n$  processes.

Thus,  $g(n, k, u, q) \geq T_c(P, k, u)/T_d(P, q)$  for any program  $P$ . Programs  $P$  for which we have equality, i.e. programs  $P$  where

$$g(n, k, u, q) = T_c(P, k, u)/T_d(P, q),$$

are referred to as extremal programs.

The function  $g(n, k, u, q)$  is also a tight bound for the same ratio taken over all programs with *at most*  $n$  processes. This follows from Theorem 5.5, where we show that the function  $g(n, k, u, q)$  is increasing in the variable  $n$ .

This increasing property can furthermore be used to compute a formula for the process independent performance function

$$G(k, u, q) = \sup_{\text{all parallel programs } P} \frac{T_c(P, k, u)}{T_d(P, q)} = \lim_{n \rightarrow \infty} g(n, k, u, q).$$

The performance function  $G(k, u, q)$  is applicable for a multiprocessor intended for any parallel program. This is the best possible bound in the case of no prior knowledge

of the parallel programs. In the case when the number of processes of the parallel programs is a priori bounded to at most  $n$  processes, the function  $G(k, u, q)$  is still an upper bound, but not tight. In this case the function  $g(n, k, u, q)$  gives a tight bound.

The formulas for  $g(n, k, u, q)$  and  $G(k, u, q)$  are presented in Section 5. For complete proofs we refer to [8].

## 4 From programs to matrices

Now consider a parallel program  $P$  and a multiprocessor with  $q$  processors. Assume that we have found an optimal dynamic allocation, with makespan  $T_d(P, q)$ . This optimal dynamic allocation will be kept fixed during the rest of this section. Next we introduce a discretization of the time interval in  $m$  subintervals  $(t_i, t_{i+1})$  of equal length, such that all synchronizing signals, process initiations and process terminations appear on the time points  $t_i$ , where  $t_i = \frac{i}{m} T_d(P, q)$ ,  $i = 0, \dots, m$ . Consequently, there is no time point  $t$ :  $t_i < t < t_{i+1}$  for any  $i$  so that a process of the program  $P$  starts or stops or a synchronization signal is executed at the time  $t$ . Obviously, all processes in the interval  $(t_{i-1}, t_i)$  are completed before any part of the processes corresponding to the interval  $(t_i, t_{i+1})$  start when using this allocation, since this is so without the discretization. For instance, a process which is active from time  $t_{i-1}$  to time  $t_{i+1}$  completes the processing in the interval  $(t_{i-1}, t_i)$  before any processing corresponding to the interval  $(t_i, t_{i+1})$  starts, and when the process starts processing the time interval  $(t_i, t_{i+1})$ , all processing corresponding to the interval  $(t_{i-1}, t_i)$  is completed.

Such a discretization is possible if all synchronizing signals and process terminations occur at rational time points, which we can assume. Therefore the discretization involves no approximation, exactly the same program is represented in a discrete way. Observe that  $m$  might be very large even if the program  $P$  is small and has a simple structure.

From the program  $P$  we next construct another program  $P'$  which will prolong or not affect the makespan using cluster allocation,  $T_c(P, k, u) \leq T_c(P', k, u)$ , but leave the makespan with dynamic allocation unchanged,  $T_d(P, q) = T_d(P', q)$ .

The construction of  $P'$  is obtained by two changes of the program  $P$ : we introduce new synchronizing signals and prolong certain processes. At every time point  $t_i$  we introduce all possible synchronization between the processes. This means that the synchronization structure now requires that all processes in the interval  $(t_{i-1}, t_i)$  have to be completed before any part of the processes corresponding to the interval  $(t_i, t_{i+1})$  may start. Since the execution time of synchronizing signals is neglected, this does not change the makespan with the fixed optimal dynamic allocation, which is  $T_d(P, q)$ . Furthermore, all processors are made to be busy at all time intervals. This is achieved by, if necessary, prolonging processes. However, no process is prolonged beyond  $T_d(P, q)$ , hence  $T_d(P, q) = T_d(P', q)$ . It is of no importance that the prolonging of processes can be made in many ways; many programs can play the role of  $P'$  to a specific program  $P$ .

By the construction we thus have  $T_d(P, q) = T_d(P', q)$ . However, since introducing more synchronizations and prolonging processes never shortens the makespan, for other allocations the makespan is either increased or unchanged. In particular, for optimal cluster allocation we therefore have  $T_c(P, k, u) \leq T_c(P', k, u)$ . Consequently,

$$\frac{T_c(P, k, u)}{T_d(P, q)} \leq \frac{T_c(P', k, u)}{T_d(P', q)}.$$

Certainly there are programs  $P$  which are left unchanged by the above transformation: programs such that  $P = P'$ . Since these programs constitute a subset of the parallel

programs we consider, we actually have

$$g(n, k, u, q) = \max_P \frac{T_c(P, k, u)}{T_d(P, q)} = \max_{P'} \frac{T_c(P', k, u)}{T_d(P', q)}.$$

Therefore, in order to calculate the maximum, only programs of the type  $P'$  need to be considered.

We next represent a program  $P'$  by an  $m \times n$  matrix of zeroes and ones only. Here each process is represented by a column, and each time period is represented by a row. The entry at the position  $(i, j)$  of the matrix is one if the  $j$ :th process is active between  $t_{i-1}$  and  $t_i$ ; if it is inactive the entry is zero. Each row contains exactly  $q$  ones, since each processor is constantly busy. Because of the complete synchronization, each row has to be completed before the next row may start.

Our next objective is to compute the makespan for cluster allocation of the program  $P'$ .

Since we consider a time ratio, the choice of time unit is immaterial. We then choose the time unit so that  $T_d(P', q) = m$  – hence the processing time for each time interval using the optimal dynamic allocation is one.

To compute the makespan for cluster allocation we need to decide how the  $n$  processes are to be allocated to the  $k$  clusters. In this case every process is to be executed within one cluster, so each cluster allocation corresponds to a way of grouping the  $n$  processes onto the  $k$  clusters. In the matrix formulation each cluster allocation corresponds to a way of grouping the  $n$  columns of the matrix together in  $k$  sets. Here each set represents one cluster.

Consider one such cluster allocation  $A$ . Assume that  $l$  of the processes which are allocated to a specific cluster are executable at a specific time interval. Within a cluster there are  $u$  processors and the processes are allocated dynamically. Since there is no synchronization within a time interval, the programs are independent. We will next see that the completion time of this cluster in this time interval is  $\max(l/u, 1)$ . The completion time clearly cannot be lower than  $l/u$ . Further, the completion time cannot be lower than 1 since no part of a process can be executed in parallel with another part of the same process. In the case  $l \geq u$  the limit  $l/u$  is reached, which has been proved by McNaughton (see Theorem 3.1 in [15]).

Because of the complete synchronization at the time points  $t_i$ , processing in the next time interval cannot start before completion of the slowest cluster in the current time interval. Since the clusters are identical, the slowest cluster is the cluster with a maximal number of active processes at that time interval. Hence, we obtain the completion time of a time interval as the maximum of  $\max(l_i/u, 1)$  over the  $k$  clusters, where  $l_i$ ,  $i = 1, \dots, k$  is the number of active processes in cluster  $i$  during the time interval. The makespan with the cluster allocation  $A$  is the sum of these maxima. If we have found an allocation of the  $n$  columns together in  $k$  sets which minimizes the makespan using cluster allocation, this is an *optimal* cluster allocation, and the makespan is denoted  $T_c(P', k, u)$ .

## 5 The matrix problem and the basic formulas

We have seen that for the sake of computing  $\max T_c(P', k, u)/T_d(P', q)$ , the program  $P'$  can be replaced by a certain  $m \times n$  matrix  $P'$  of zeroes and ones only. In this matrix each row has exactly  $q$  ones, and thus  $n - q$  zeroes,  $1 \leq q \leq n$ . We say that a matrix  $P'$  of this type is an  *$n, q$ -matrix*.

Assume that  $A$  is a partition of the columns in  $k$  sets. Let us also denote by  $P_i$  the  $i$ th row of  $P$ , and by  $A_j(P_i)$  the number of 1:s in the  $j$ :th partition set of  $P_i$ . We have defined the function

$$T_c(A, P, k, u) = \sum_i \max_{j=1, \dots, k} (1, A_j(P_i)/u),$$

and

$$T(P, k, u) = \min_{A \in \mathcal{A}(P)} T_c(A, P, k, u),$$

where  $\mathcal{A}(P)$  is the set of cluster allocations of a program  $P$ . We want to calculate

$$\max_{P \in \mathcal{P}_n} \frac{T_c(P, k, u)}{m(P)}$$

where  $\mathcal{P}_n$  is the set of parallel programs of  $n$  processes.

We call a  $n, q$ -matrix *complete* if all  $\binom{n}{q}$  permutations of the  $q$  ones occur equally frequently as rows. For a complete matrix, the number of rows is thus necessarily divisible by  $\binom{n}{q}$ . When considering  $n$  and  $q$  fixed, complete  $n, q$ -matrices differ from each other only in that they may have different number of copies of each row, and that the rows may occur in different order.

We refer to partitions where the sizes of the sets in the partition differ as little as possible as *uniform partitions*. If  $n/k$  is an integer  $w$ , every set in an uniform partition has  $w$  members. Denote the integer part of  $n/k$ , the floor function, by  $\lfloor n/k \rfloor$ , and the smallest integer greater than or equal to  $n/k$ , the ceiling function, by  $\lceil n/k \rceil$ . If  $n/k$  is not an integer, the sets in an uniform partition have  $\lfloor n/k \rfloor$  or  $\lceil n/k \rceil$  members.

In [8] we prove the following theorems:

**THEOREM 5.1** *If  $P'$  and  $Q'$  are complete  $n, q$ -matrices, then*

$$\frac{T_c(P', k, u)}{T_d(P', q)} = \frac{T_c(Q', k, u)}{T_d(Q', q)}.$$

Furthermore,

$$g(n, k, u, q) = \max_{\text{all } n, q\text{-matrices } P'} \frac{T_c(P', k, u)}{T_d(P', q)} = \frac{T_c(P'', k, u)}{T_d(P'', q)},$$

where  $P''$  is a complete  $n, q$ -matrix.

I.e., the complete matrices are extremal.

**THEOREM 5.2** *All uniform partitions are optimal partitions for a complete  $n, q$ -matrix.*

By Theorem 5.1 we may consider a complete matrix where each row occur exactly once, i.e. the number of rows is  $\binom{n}{q}$ . Then a complete matrix represents all possible ways to put  $q$  objects in  $n$  slots. The results of the present paper exploits this interpretation to give a recursive and more efficient way to calculate the values of  $g(n, k, u, q)$  and  $G(k, u, q)$ .

In [8] explicit formulas are calculated by counting the number of permutations corresponding to each value of  $\max(l/u, 1)$ . Since we are interested in the maximum of 1:s in a partition set, decreasing sequences are used to represent the number of 1:s in each set. Then the first entry in the sequence gives the maximum. However, the number of decreasing sequences increases exponentially in the parameters  $q$  and  $k$ .

Next we give the formulas of [8].

Let  $I = \{i_1, \dots, i_{k-1}\}$  be a decreasing finite sequence of non-negative integers. Then we define:

$b(l, I) =$  the number of distinct integers in  $\{l, i_1, \dots, i_{k-1}\}$ .

$a(l, I, j) =$  the number of occurrences of the  $j$ :th distinct integer in  $\{l, i_1, \dots, i_{k-1}\}$ , enumerated in size order,  $1 \leq j \leq b(l, I)$ .

$\pi(k, w, q, l) =$  the number of permutations of  $q$  ones distributed in  $kw$  slots, which are divided in  $k$  sets with  $w$  slots in each, such that the set with maximum number of ones has exactly  $l$  ones. N.B., all parameters to  $\pi$  are integers.

**THEOREM 5.3** *Given positive integers  $n, k, q$  and  $u$  where  $n \geq q$ .*

*In the case where  $w = n/k$  is an integer, we have:*

$$g(n, k, u, q) = \frac{1}{u \binom{n}{q}} \sum_{l=1}^{\min(w, q)} \max(l, u) \pi(k, w, q, l).$$

*If  $n/k$  is not an integer we let  $w = \lfloor n/k \rfloor$  and denote the remainder of  $n$  divided by  $k$  by  $n_k$ . Then we have:*

$$g(n, k, u, q) = \frac{1}{u \binom{n}{q}} \sum_{\substack{l_1 = \max(0, \lceil \frac{q - (k - n_k)w}{n_k} \rceil) \\ l_2 = \max(0, \lceil \frac{q - l_1 n_k}{k - n_k} \rceil)}}^{\min(w+1, q)} \max(l_1, l_2, u) \times \sum_{i=\max(l_1, q - l_2(k - n_k))}^{\min(l_1 n_k, q - l_2)} \pi(n_k, w + 1, i, l_1) \pi(k - n_k, w, q - i, l_2).$$

The formula for  $\pi$  is given in the following lemma:

**LEMMA 5.4** *In the special cases  $\min(q, w) < l$  and  $q > kl$ ,  $\pi(k, w, q, l) = 0$ . Otherwise, if  $k = 1$  and  $q = l$ ,  $\pi(1, w, q, l) = \binom{w}{l}$ . In all other cases we have*

$$\pi(k, w, q, l) = \binom{w}{l} \sum_{I \in \mathcal{I}} \binom{w}{i_1} \cdot \dots \cdot \binom{w}{i_{k-1}} \frac{k!}{\prod_{j=1}^{b(l, I)} a(l, I, j)!}.$$

*The set  $\mathcal{I}$  is the set of all sequences of non-negative integers  $I = \{i_1, \dots, i_{k-1}\}$  which are decreasing:  $i_j \geq i_{j+1}$  for all  $j = 1, \dots, k-2$ , bounded by  $l$ :  $i_1 \leq l$ , and have sum  $q - l$ :  $\sum_{j=1}^{k-1} i_j = q - l$ .*

The next theorem establishes basic properties of the function  $g(n, k, u, q)$ . Here (1) and (2) are obviously true from an application domain point of view. These results thus emphasize the relevance of the combinatorial model. The results (3) and (4) are quantitative statements valid for very large parameter values, which appear to be new for the multiprocessor society. We remark that the function  $g(n, k, 1, k)$  represents comparison of static to dynamic allocation.

- THEOREM 5.5**
1.  $g(n, k, u, q)$  is increasing in the variables  $n$  and  $q$ , and decreasing in the variables  $u$  and  $k$ .
  2.  $g(wk, k, 1, k)$  is increasing as a function of  $k$ .
  3. Given positive integers  $\nu$ ,  $w$  and  $u$ , we have:

$$\begin{aligned} \max\left(\frac{w+1}{u}, 1\right) - \left(\max\left(\frac{w+1}{u}, 1\right) - \max\left(\frac{w}{u}, 1\right)\right)(1 - w^{-(w+1)})^\nu \\ \geq \lim_{k \rightarrow \infty} g(wk + \nu, k, u, k) \\ \geq \max\left(\max\left(\frac{w}{u}, 1\right), \max\left(\frac{w+1}{u}, 1\right) - \left(\max\left(\frac{w+1}{u}, 1\right) - 1\right)(1 - w^{-(w+1)})^\nu\right). \end{aligned}$$

4. The functions  $g(n, k, u, q)$  and  $g(n, k, 1, k)$  are unbounded.

Note further that  $g(n, k, u, q) = 1$  if  $u \geq \min(\lceil n/k \rceil, q)$  or if  $k \geq n$ , and  $g(n, k, u, q) = g(n, k, u, n) = \max(\lceil n/k \rceil/u, 1)$  for all  $q \geq n$ .

If  $w > u$ , (3) reduces to

$$\begin{aligned} \frac{w+1}{u} - \left(\frac{w+1}{u} - 1\right)(1 - w^{-(w+1)})^\nu \\ \geq \lim_{k \rightarrow \infty} g(wk + \nu, k, u, k) \\ \geq \max\left(\frac{w}{u}, \frac{w+1}{u} - \frac{1}{u}(1 - w^{-(w+1)})^\nu\right). \end{aligned}$$

*Proof:* We here only include the part of the proof of (1) which is relevant for Theorem 5.6.

That  $g$  is increasing as a function of  $n$  follows by considering an extremal matrix  $P$  with  $n$  processes, i.e.  $T_c(P, k, u)/T_d(P, q) = g(n, k, u, q)$  by Theorem 5.1 and 5.2. Consider also a matrix  $P'$  which is identical to  $P$  with the exception that a new column of only 0:s is added. This new column is inactive during the entire execution of  $P'$ , i.e. there are  $n+1$  processes in  $P'$ . Obviously,  $T_c(P, k, u)/T_d(P, q) = T_c(P', k, u)/T_d(P', q)$ .

The column is added to the smallest partition set so that the new partition is still uniform. However,  $P'$  is not complete, in general. From the definition of  $g$  we therefore know that  $T_c(P', k, u)/T_d(P', q) \leq g(n+1, k, u, q)$ . Consequently,  $g(n, k, u, q) = T_c(P', k, u)/T_d(P', q) \leq g(n+1, k, u, q)$ .

Using Theorem 5.5 and Stirlings formula  $n! \approx (\frac{n}{e})^n \sqrt{2\pi n}$ , it is possible to prove the following formula for  $G(k, u, q)$ :

**THEOREM 5.6** For any positive integers  $k, u$  and  $q$ ,

$$G(k, u, q) = \lim_{n \rightarrow \infty} g(n, k, u, q) = \sup_{n \in \mathbb{N}} g(n, k, u, q),$$

where  $G(1, u, q) = \max(q/u, 1)$ , and if  $k > 1$ :

$$G(k, u, q) = \frac{k!q!}{k^q u} \sum_{l=1}^q \frac{\max(l, u)}{l!} \sum_{I \in \mathcal{I}} (\prod_{j=1}^{k-1} i_j! \prod_{j=1}^{b(l, I)} a(l, I, j)!)^{-1}.$$

The set  $\mathcal{I}$  is the set of all sequences  $I = \{i_1, \dots, i_{k-1}\}$  of non-negative integers which are decreasing;  $i_j \geq i_{j+1}$  for all  $j = 1, \dots, k-2$ , bounded by  $l$ ;  $i_1 \leq l$ , and have sum  $q-l$ ;  $\sum_{j=1}^{k-1} i_j = q-l$ .

Further, the function  $G(k, u, q)$  is decreasing as a function of  $k$  and as a function of  $u$ , and increasing as a function of  $q$ .  $G(k) = G(k, 1, k)$  is an increasing function.

We may alternatively use multinomial coefficient notation:

$$\binom{q}{i_1, \dots, i_k} = \frac{q!}{\prod_{j=1}^k i_j!} = \binom{q}{I},$$

where  $\sum_{j=1}^k i_j = q$ . Let us also denote the vector  $(a(I, 1), a(I, 2), \dots, a(I, k))$  by  $a(I)$ , and denote by  $l, I$  the extended sequence  $\{l, i_1, \dots, i_{k-1}\}$ .

Then we may write the formula as

$$G(k, u, q) = \frac{1}{u k^q} \sum_{l=1}^q \sum_{I \in \mathcal{I}} \max(l, u) \binom{q}{l, I} \binom{k}{a(l, I)}.$$

Here  $\mathcal{I}$  is the same set of decreasing sequences  $I = \{i_1, \dots, i_{k-1}\}$  as in Theorem 5.6.

## 6 A recursive calculation technique

A direct implementation of the formula in Theorem 5.3 and Theorem 5.6 results in an execution time that grows exponentially with the number of processors since the number of decreasing sequences increases exponentially. We next present a new technique for calculating  $G(2^r, u, q)$ , which is polynomial in time, thus making it possible to consider much larger systems.

**THEOREM 6.1** *Assume that  $r, u$  and  $q$  are positive integers. Then*

$$G(2^r, u, q) = \sum_{x=1}^q \max(1, x/u) f(2^r, q, x),$$

where

$$f(2^r, q, x) = F(2^r, q, x) - F(2^r, q, x-1),$$

and

$$F(2^r, q, x) = 2^{-q} \sum_{j=0}^q F(2^{r-1}, j, x) F(2^{r-1}, q-j, x) \binom{q}{j}.$$

Start values of the recursion in  $r$  is given by  $F(1, j, x) = 1$  if  $j \leq x$ , and  $F(1, j, x) = 0$  if  $j > x$ .

The number  $G(2^r, u, q)$  can be calculated with  $O(rq^3)$  multiplications and  $q+1$  binomial coefficient calculations.

Hence, with  $k = 2^r$  we need  $O(\log k q^3)$  multiplications. We further remark that if The theorem gives information about the value of  $G(k, u, q)$  for other values of  $k$  than powers of 2 by the fact that  $G(k, u, q)$  is decreasing in the variables  $k$  and  $u$ , and increasing in the variable  $q$  (see [8]).

*Proof:* We first claim that the formula for  $G(k, u, q)$  can be calculated in the following way. Consider  $k$  boxes in which we put  $q$  balls. Suppose that each ball is placed in a box randomly with an uniform distribution. The boxes are large - there is no limitation of the number of balls in each box. Then  $G(k, 1, q)$  is the expectation of the number of balls in the box with maximum number of balls. Consider all possible combinations of putting the different balls into the boxes, denote the number of such combinations by  $c$ . For each such combination  $i$ , the maximum number of balls in a box is denoted  $m_i$ . We have stated that  $G(k, 1, q) = \sum_i m_i/c$ . We also claim that

$$G(k, u, q) = \sum_i \max(m_i/u, 1)/c.$$

In [8] the problem of allocating a parallel program of  $n$  processes is transformed into a problem of balls and boxes containing in total  $n$  slots - possible ball positions. These slots are distributed evenly among the boxes, hence each box has  $\lfloor n/k \rfloor$  or  $\lceil n/k \rceil$  slots. Here the corresponding formula  $g(n, k, 1, q)$  is the mean value of the maximal number of balls ( $m$ ), summed over all different ways to distribute the balls in the slots. A similar statement is valid for  $g(n, k, u, q)$  and the function  $\max(m/u, 1)$ . In [8] the formula for  $G(k, u, q)$  is calculated essentially by letting  $n \rightarrow \infty$ . We then simultaneously obtain the interpretation of  $G(k, u, q)$  which is described previously in this proof.

Let  $f(k, q, x)$  be the share of the combinations for which  $m_i = x$ . Then we have

$$G(k, u, q) = \sum_i \max(m_i/u, 1)f(k, q, x).$$

Now we define a distribution function  $F$  corresponding to  $f$ :

$$F(k, q, y) = \sum_i f(k, q, x) = \frac{E(k, q, x)}{\binom{k+q-1}{q}}.$$

Here  $E(k, q, x)$  is the number of combinations where the maximal number of balls in a box is at most  $x$ .  $G(k, u, q)$  can be obtained by looking at two sets containing  $k/2$  boxes each - we assume that  $k$  is a power of two. The  $q$  balls can be placed into any box. The combinations of putting balls into the  $k$  boxes can be divided into  $q + 1$  categories, corresponding to the cases when there are  $j$  balls in the first set of  $k/2$  boxes and thus  $q - j$  balls in the second set of  $k/2$  boxes ( $0 \leq j \leq q$ ). Assuming we have  $j$  balls in the first set of boxes, let  $E_j(k, q, x)$  denote the number of combinations where the maximal number of balls in any of the  $k$  boxes is  $x$  or lower. The corresponding distribution function is

$$F_j(k, q, x) = \frac{E_j(k, q, x)}{\binom{k/2+j-1}{j} \binom{k/2+q-j-1}{q-j}}.$$

By comparing  $E_j(k, q, x)$  with the definition of  $E(k, q, x)$ , we then get the relation  $E_j(k, q, x) = E(k/2, j, x)E(k/2, q - j, x)$ . This gives immediately  $F_j(k, q, x) = F(k/2, j, x)F(k/2, q - j, x)$  by dividing by the binomial factors.

Since there are different ways of choosing  $j$  balls to the first set of boxes from  $q$  balls, we also have

$$F(k, q, x) = 2^{-q} \sum_j F_j(k, q, x) \binom{q}{j}.$$

Consequently,

$$F(k, q, x) = 2^{-q} \sum_{j=0}^q F(k/2, j, x)F(k/2, q - j, x) \binom{q}{j}.$$

This sum involves  $2(q+1)$  multiplications. Suppose that  $k = 2^r$ . If  $i < r$  we need all values of  $F(2^i, j, x)$ ,  $j = 1, \dots, q$  and  $x = 1, \dots, q$ . Therefore, for each  $i$ ,  $2(q+1)^2$  multiplications are needed. This means that  $F(k, q, x)$  can be obtained from the  $q+1$  functions

$F(1, j, x)$  by at most  $r(q + 1)^3$  multiplications. Hence we may compute  $F(k, q, x)$  in polynomial time. The initial values  $F(1, j, x)$  are trivial: if  $j \leq x$ ,  $F(1, j, x) = 1$ , and if  $j > x$ ,  $F(1, j, x) = 0$ . The function  $f(k, q, x)$  is then obtained as  $f(k, q, x) = F(k, q, x) - F(k, q, x - 1)$ . Finally  $G(k, u, q)$  is calculated by

$$G(k, u, q) = \sum_{x=1}^q \max(x/u, 1)f(k, q, x).$$

The theorem is proved.

Appendix A shows the C program implementing the algorithm above. The program operates on a three dimensional array. The first dimension corresponds to the doubling of  $k$ , i.e. each step in this dimension corresponds to doubling  $k$ . For each step in this dimension we have a  $(q + 1) \times (q + 1)$ -matrix. Value  $(x, y)$  in this matrix is  $F(k, x, y)$ . Consider the case when  $k = 4, q = 3$  and  $u = 1$ . Initially we consider the case when  $k = 1$ . This is a trivial case and we get the matrix  $((1,1,1,1), (0,1,1,1), (0,0,1,1), (0,0,0,1))$ . When  $k = 2$ , i.e. in the first iteration, we get the matrix  $((1,1,1,1), (0,1,1,1), (0,0,5,1,1), (0,0,0,75,1))$ . When  $k = 4$ , i.e. in the second iteration we get the matrix  $((1,1,1,1), (0,1,1,1), (0,0,75,1,1), (0,0,38,0,94,1))$ , i.e.  $F(4, 3, 0) = 0$ ,  $F(4, 3, 1) = 0.38$ ,  $F(4, 3, 2) = 0.94$ , and  $F(4, 3, 3) = 1$ . From this we get  $f(4, 3, 0) = 0$ ,  $f(4, 3, 1) = 0.38$ ,  $f(4, 3, 2) = 0.56$ , and  $f(4, 3, 3) = 0.06$ . This results in  $G(4, 1, 3) = 1.68$ .

In the following table, values in bold face correspond to optimal comparison of two multiprocessors with equal number of processors.

## 7 Conclusion

In this paper we have presented a technique for calculating  $G(k, u, q)$  for much larger configurations than previously. The technique has been implemented in a C program which is included in Appendix A. We have calculated  $G(k, u, q)$  for a number of values on  $k$ ,  $u$ , and  $q$ . These calculations show that systems with a large number of clusters of unit size can have a significantly lower performance than large one-cluster systems. E.g. a system with 1024 clusters of unit size can have a performance which is 4 times lower than a one cluster system with 512 processors even when using optimal allocations of processes to processors ( $G(1024, 1, 512) = 4.013$ ). By looking at the proofs in [6] and [8], and the results in Table 1, we conclude that there are programs which have 100% utilization (i.e. all processors are constantly busy) using a cluster with 256 processors and for which it is impossible to obtain a utilization exceeding  $1/(2^*3.7) = 13.5\%$  on a system with 512 clusters of unit size, i.e the completion time may 3.7 times longer and there are twice as many processors in the system with unit sized clusters. By looking at the proofs in [6] and [8] and Table 1, we see that the same program would yield a processor utilization of  $1/(2^*1.2) = 42\%$  on a multiprocessor with 64 clusters containing 8 processors each, i.e. there are still 512 processors but  $G(64, 8, 256) = 1.2$ . The proofs in [6] and [8] show that the worst-case program is the same for the case with 512 clusters of unit size and 64 clusters with 8 processors.

Consequently, as the number of processors in each cluster increases the maximum gain of using one large cluster becomes less. The conclusion of this is that one should avoid building large systems with clusters of unit size. Increasing the cluster size to 4 or 8 processors will improve the worst-case behavior significantly.

Table 1. Some values of  $G(k, u, q)$ .

$G(k, 1, q)$	$k = 16$	$k = 32$	$k = 64$	$k = 128$	$k = 256$	$k = 512$	$k = 1024$
$q = 8$	2.063	1.665	1.379	1.203	1.105	1.054	1.027
$q = 16$	<b>3.078</b>	2.392	1.991	1.656	1.389	1.213	1.112
$q = 32$	4.830	<b>3.533</b>	2.722	2.233	1.937	1.391	1.392
$q = 64$	7.880	5.415	<b>3.958</b>	3.078	2.456	2.125	1.904
$q = 128$	13.357	8.651	5.967	<b>4.379</b>	3.385	2.742	2.268
$q = 256$	23.436	14.384	9.370	6.493	<b>4.767</b>	3.683	3.030
$q = 512$	42.370	24.823	15.336	10.048	6.992	<b>5.158</b>	4.013
$G(k, 2, q)$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$	$k = 256$	$k = 512$
$q = 8$	1.300	1.091	1.025	1.006	1.002	1.000	1.000
$q = 16$	<b>2.101</b>	1.539	1.201	1.060	1.016	1.004	1.001
$q = 32$	3.524	<b>2.415</b>	1.766	1.361	1.123	1.035	1.009
$q = 64$	6.118	3.940	<b>2.707</b>	1.979	1.539	1.228	1.070
$q = 128$	10.955	6.678	4.325	<b>2.984</b>	2.189	1.693	1.371
$q = 256$	20.137	11.718	7.192	4.685	<b>3.246</b>	2.383	1.841
$q = 512$	37.806	21.185	12.411	7.668	5.024	<b>3.496</b>	2.579
$G(k, 4, q)$	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$	$k = 256$
$q = 8$	1.032	1.003	1.000	1.000	1.000	1.000	1.000
$q = 16$	<b>1.535</b>	1.102	1.010	1.001	1.000	1.000	1.000
$q = 32$	2.748	<b>1.762</b>	1.215	1.027	1.002	1.000	1.000
$q = 64$	5.050	3.058	<b>1.970</b>	1.354	1.058	1.005	1.000
$q = 128$	9.477	5.477	3.339	<b>2.163</b>	1.492	1.111	1.011
$q = 256$	18.081	10.068	5.859	3.596	<b>2.343</b>	1.623	1.193
$q = 512$	34.933	18.903	10.592	6.205	3.834	<b>2.512</b>	1.748
$G(k, 8, q)$	$k = 2$	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$
$q = 8$	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$q = 16$	<b>1.196</b>	1.000	1.000	1.000	1.000	1.000	1.000
$q = 32$	2.280	<b>1.374</b>	1.020	1.000	1.000	1.000	1.000
$q = 64$	4.397	2.525	<b>1.529</b>	1.050	1.001	1.000	1.000
$q = 128$	8.563	4.739	2.739	<b>1.670</b>	1.101	1.002	1.000
$q = 256$	16.797	9.040	5.034	2.930	<b>1.798</b>	1.174	1.004
$q = 512$	33.127	17.466	9.451	5.296	3.102	<b>1.917</b>	1.256

## References

- [1] M. Garey and D. Johnson, *Computers and Intractability*, W.H. Freeman and Company, 1979.
- [2] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, PVM: Parallel Virtual Machine, MIT press, 1994.
- [3] R. L. Graham, *Bounds on Multiprocessing Anomalies*, SIAM Journal of Applied Mathematics, Vol. 17, No 2, pp. 416-429, March, 1969.
- [4] E. Hagersten and M. Koster, WildFire: A Scalable Path for SMPs, in Proceedings of the Fifth International Symposium On High Performance Computer Architecture (HPCA-5), Orlando, Florida, 1999.

- [5] E.S.H. Hou, R. Hong and N. Ansari, Efficient multiprocessor scheduling based on genetic algorithms, 16th Annual Conference of the IEEE Industrial Electronics Society, Vol II, 1990, pp. 1239-1243.
- [6] H. Lennerstad, L. Lundberg, *An Optimal Execution Time Estimate of Static versus Dynamic Allocation in Multiprocessor Systems*, SIAM Jou. of Comp., Vol. 24, No. 4, pp. 751-764, August 1995.
- [7] H. Lennerstad, L. Lundberg, *Combinatorics for Multiprocessor Scheduling Optimization and Other Contexts in Computer Architecture*, Proceedings of the Conference of Combinatorics and Computer Science, Brest, France, 1995. In Lecture Notes of Computer Science 1120, Springer Verlag, 1996.
- [8] H. Lennerstad, L. Lundberg, *Optimal Combinatorial Functions Comparing Multi-process Allocation Performance in Multiprocessor Systems*, SIAM Jou. of Comp., to appear.
- [9] L. Lundberg, H. Lennerstad, *Using Recorded Values for Bounding the Minimum Completion Time in Multiprocessors*, IEEE Trans. on Para. and Distr. Syst., Vol 9, No 4, April 1998.
- [10] L. Lundberg, Predicting and Bounding the Speedup of Multithreaded Solaris Programs, Journal of Parallel and Distributed Computing, Vol. 57, No. 3, June 1999, pp. 322-333.
- [11] L. Lundberg, H. Lennerstad, *An Optimal Upper Bound on the Minimal Completion Time in Distributed Supercomputing*, Proceedings of the 8th ACM Conference on Supercomputing, Manchester, England, pp. 196-203, 1994.
- [12] L. Lundberg, H. Lennerstad, *Comparing the Optimal Performance of Different MIMD Multiprocessor Architectures*, Proceedings of the 12th International Parallel Processing Symposium, Orlando, Florida, pp. 498-502, March/April, 1998.
- [13] L. Lundberg, H. Lennerstad, *An optimal bound on the gain of using one large processor cluster instead of a number of small clusters*, Proceedings of the ISCA International Conference on Parallel and Distributed Computing Systems, Orlando, Fl, pp. 576-582, 1995.
- [14] L. Lundberg, H. Lennerstad, *Using Recorded Values for Bounding the Minimum Completion Time in Multiprocessors*, IEEE Trans. on Para. and Distr. Syst., Vol 9, No 4, April 1998.
- [15] R. McNaughton, *Scheduling with Deadlines and Loss Functions*, Management Science, Vol 6 No 1, pp. 1-12, 1959.
- [16] MPI Forum, *MPI: A message-passing interface standard*, International Journal of Supercomputer Application, 8 (3/4), pp. 165-416, 1994.
- [17] A.K. Nanda, D. DeGroot, D.L. Stenger, Scheduling Directed Task Graphs on Multiprocessors using Simulated Annealing, in Proceedings of the IEEE 12th International Conference on Distributed Systems, 1992, pp. 20-27.

## A The C code of a program for calculating $G(k, u, q)$

```
#include <stdio.h>
#define Q 129 /* Q = q+1 */
#define U 1
```

```

#define ITERATIONS 9 /* k = 2**ITERATIONS-1 */
double matrix[ITERATIONS][Q][Q];
double vect[ITERATIONS][Q];
double over[Q][Q];
int init() {
    int x,y;
    for (x=0; x < Q; x++)
        for (y=0; y < Q; y++)
            if (y < x) matrix[0][x][y] = 0.0;
            else matrix[0][x][y] = 1.0;
    for (i=0; i < Q; i++)
        for (j=0; j < Q; j++) over[i][j]=1.0;
    for (i=1; i < Q; i++)
        for (j=1; j <= i; j++) over[i][j]=over[i][j-1]*(i-j+1)/j;
}
main() {
    int i,k,x,y;
    double tmp;
    init();
    for (k=0; k < ITERATIONS-1; k++) {
        for (x=0; x < Q; x++)
            for (y=0; y < Q; y++)
                matrix[k][x][y] = matrix[k][x][y]/matrix[k][x][Q-1];
        for (x=0; x < Q; x++)
            for (y=0; y < Q; y++){
                matrix[k+1][x][y] = 0;
                for (i=0; i < (x+1); i++)
                    matrix[k+1][x][y] = matrix[k+1][x][y] +
                        matrix[k][i][y]*matrix[k][x-i][y]*over[x][i];}
    }
    for (k=0; k < ITERATIONS; k++)
        for (x=0; x < Q; x++){
            vect[k][x] = matrix[k][x][Q-1];
            for (y=0; y < Q-1; y++)
                matrix[k][x][Q-y-1] = matrix[k][x][Q-y-1] -
                    matrix[k][x][Q-y-2];}
    for (k=0; k < ITERATIONS; k++)
        for (x=0; x < Q; x++)
            for (y=0; y < Q; y++)
                matrix[k][x][y] = matrix[k][x][y]/vect[k][x];
    tmp = 0;
    for (y=1; y < Q; y++)
        if (y < U) tmp = tmp + matrix[ITERATIONS-1][Q-1][y];
        else tmp = tmp + matrix[ITERATIONS-1][Q-1][y]*y/U;
    printf("G = %f\n",tmp); }

```

# On the enumeration of Permutation Polynomials (Extended abstract)

Claudia Malvenuto and Francesco Pappalardi

## Abstract

Given a permutation  $\sigma$  of the elements of a finite field  $\mathbb{F}_q$ , the permutation polynomial  $f_\sigma \in \mathbb{F}_q[x]$  is the unique polynomial with degree less than  $q - 1$  that has the property that  $f_\sigma(t) = \sigma(t)$  for every  $t \in \mathbb{F}_q$ . We consider the natural question of enumerating the permutations in a given conjugacy class for which the permutation polynomial has degree strictly less than  $q - 2$ . We give formulas that extend existing ones. Furthermore for the case of  $k$ -cycles, we consider the harder problem of enumerating the permutations within a given conjugacy class for which the permutation polynomial has minimal degree. After giving an upper bound and a lower bound (for  $q \equiv 1 \pmod{k}$ ) we consider various examples in which interesting Galois properties arise.

## Résumé

Soit  $\sigma$  une permutation des éléments d'un corps fini  $\mathbb{F}_q$ : le polynôme de permutation  $f_\sigma \in \mathbb{F}_q[x]$  est le seul polynôme ayant degré inférieur à  $q - 1$  et tel que  $f_\sigma(t) = \sigma(t)$  pour chaque  $t \in \mathbb{F}_q$ . Nous considérons la question naturelle d'énumérer les permutations dans une classe de conjugaison donnée dont le polynôme de permutation ait un degré strictement inférieur à  $q - 2$ . Nous donnons des formules qui étendent celles déjà connues. De plus dans le cas de  $k$ -cycles, nous considérons le problème plus complexe d'énumérer les permutations dans une classe de conjugaison donnée dont le polynôme de permutation associé ait un degré minimal. Après avoir donné des bornes supérieure et inférieure (pour  $q \equiv 1 \pmod{k}$ ) nous prenons en considération divers exemples qui possèdent des propriétés de Galois intéressantes.

## 1 Introduction

Let  $q$  be a power of a prime and denote with  $\mathbb{F}_q$  the finite field with  $q$  elements. If  $\sigma$  is a permutation of the elements of  $\mathbb{F}_q$ , then one can associate to  $\sigma$  the polynomial in  $\mathbb{F}_q[x]$

$$f_\sigma(x) = \sum_{c \in \mathbb{F}_q} \sigma(c) \left( 1 - (x - c)^{q-1} \right).$$

Such a polynomial has the property that

1.  $f_\sigma(b) = \sigma(b)$  for all  $b \in \mathbb{F}_q$ ;
2. The degree  $\partial(f_\sigma) \leq q - 2$  (since the sum of all the elements of  $\mathbb{F}_q$  is zero).

$f_\sigma$  is the unique polynomial in  $\mathbb{F}_q$  with these two properties and it is called *the permutation polynomial of  $\sigma$* .

Permutation polynomials have increasingly attracted the attentions of various researchers in the past couple of decades. We suggest the inspiring survey papers by Rudolf Lidl and Gary Mullen<sup>1</sup> for an introduction. A key exchange protocol for public key cryptography based on permutation polynomials has been proposed by Joel V. Brawley<sup>2</sup>

Let us denote with  $S_\sigma$  the set of elements of  $\mathbb{F}_q$  that are moved by  $\sigma$ . Note that if  $\sigma$  and  $\sigma'$  are conjugated, then  $|S_\sigma| = |S_{\sigma'}|$ .

We have that  $\partial(f_\sigma) \geq q - |S_\sigma|$ . To see this it is enough to note that by the first property of the definition, the polynomial  $f_\sigma(x) - x$  has as roots all the elements of  $\mathbb{F}_q$  which are not in  $S_\sigma$ . Therefore, if not identically zero, it has to have degree at least  $q - |S_\sigma|$ .

An immediate consequence is that all transpositions give rise to polynomials of degree  $q - 2$  while the degree of a 3-cycle can be  $q - 2$  or  $q - 3$ .

Let  $\mathcal{C}$  be a conjugation class of permutations of a finite field  $\mathbb{F}_q$ . We consider the function  $N_C(q)$  defined as the number of permutations in  $\mathcal{C}$  for which the associated permutation polynomial has degree  $< q - 2$ .

In 1969, C. Wells<sup>3</sup> proved the formula

$$N_{[3]}(q) = \begin{cases} \frac{2}{3}q(q-1) & \text{if } q \equiv 1 \pmod{3} \\ 0 & \text{if } q \equiv 2 \pmod{3} \\ \frac{1}{3}q(q-1) & \text{if } q \equiv 0 \pmod{3} \end{cases}.$$

where  $[k]$  denotes the conjugation class of  $k$ -cycles.

## 2 Permutation Polynomials with non-maximal degree

We will prove formulas for  $N_{[k]}(q)$  where  $k = 4, 5, 6$  and for the classes of permutations of type  $[2\ 2]$ ,  $[3\ 2]$ ,  $[4\ 2]$ ,  $[3\ 3]$  and  $[2\ 2\ 2]$ .

Namely, suppose  $q$  is odd and let  $\eta$  denote the quadratic character<sup>4</sup> of  $\mathbb{F}_q$ . Then

<sup>1</sup> When does a polynomial over a finite field permute the elements of the field? Amer. Math. Mon. 95 (1988), 243–246, Amer. Math. Mon. 100 (1993), 71–74 )

<sup>2</sup> Some Cryptographic Applications of Permutation Polynomials, Cryptologia 1, Number 1, (1977) 76–92.

<sup>3</sup> The degrees of permutation polynomials over finite fields, J. Combinatorial Theory 7 (1969) 49–55

<sup>4</sup>i.e.  $\eta(a) = \begin{cases} 1 & \text{if } a \text{ is a square in } \mathbb{F}_q \\ -1 & \text{if } a \text{ is not a square in } \mathbb{F}_q \\ 0 & \text{if } a=0 \end{cases}$

$$\begin{aligned}
N_{[4]}(q) &= \frac{1}{4}q(q-1)(q-5-2\eta(-1)-4\eta(-3)) \\
N_{[2\ 2]}(q) &= \frac{1}{8}q(q-1)(q-4)\{1+\eta(-1)\} \\
N_{[5]}(q) &= \frac{1}{5}q(q-1) \\
&\quad (q^2-(9-\eta(5)-5\eta(-1)+5\eta(-9))q+26+5\eta(-7) \\
&\quad +15\eta(-3)+15\eta(-1)) \\
N_{[2\ 3]}(q) &= \frac{1}{12}q(q-1) \\
&\quad (q^2-(9+\eta(-3)+3\eta(-1))q+(24+6\eta(-3)+18\eta(-1)+6\eta(-7))) \\
&\quad +\eta(-1)(1-\eta(9))q(q-5)
\end{aligned}$$

The case of even characteristics has also been settled<sup>5</sup>. Suppose  $q = 2^n$ . Then

$$\begin{aligned}
N_{[4]}(2^n) &= \frac{1}{4}2^n(2^n-1)(2^n-4)(1+(-1)^n) \\
N_{[2\ 2]}(2^n) &= \frac{1}{8}2^n(2^n-1)(2^n-2) \\
N_{[5]}(2^n) &= \frac{1}{5}2^n(2^n-1)(2^n-3-(-1)^n)(2^n-6-3(-1)^n) \\
N_{[2\ 3]}(2^n) &= \frac{1}{12}2^n(2^n-1)(2^n-3-(-1)^n)(2^n-6).
\end{aligned}$$

Similar formulas (which we report at the end of this section) have been computed for the four conjugation classes of permutations that move 6 elements.

As an example let us produce the proof that

**Proposition 1** *If  $q$  is odd, then*

$$N_{[4]}(q) = \frac{1}{4}q(q-1)(q-5-2\eta(-1)-4\eta(-3))$$

**Proof.** The first step consists in showing that

if  $a, b, c, d \in \mathbb{F}_q$  (all distinct)  
such that the 4-cycle  $\sigma = (a\ b\ c\ d)$  is counted by  $N_{[4]}(q)$ ,  
have to satisfy the equation:

$$(a-b)a + (b-c)b + (c-d)c + (d-a)d = 0. \quad (1)$$

Indeed, if  $f_\sigma$  is the permutation polynomial of  $\sigma$  and we write

$$f_\sigma(x) = A_1x^{q-2} + A_2x^{q-3} + \cdots + A_{q-2}x + A_{q-1},$$

then

$$A_1(\sigma) = - \sum_{c \in \mathbb{F}_q} \sigma(c)c.$$

Since the squares of all the elements of  $\mathbb{F}_q$  add up to zero, the previous formula

---

<sup>5</sup>Part of the formulas for even characteristics are due to A. Conflitti

can be written as

$$A_1(\sigma) = - \sum_{c \in \mathbb{F}_q} (c - \sigma(c))c = \sum_{c \in S_\sigma} (\sigma(c) - c)c.$$

In our case,  $S_\sigma = \{a, b, c, d\}$ . Therefore  $A_1(\sigma)$  equals to the left hand side of (1). Since  $N_{[4]}(q)$  counts the  $\sigma$  for which  $A_1(\sigma) = 0$ , we conclude the first step of the proof.

Now for each of the  $q(q-1)$  fixed choices of  $a$  and  $b$  distinct in  $\mathbb{F}_q$ ,

substituting in (1),  $c = x(b-a) + a$ ,  $d = y(b-a) + a$ , we obtain the equation

$$(1-x) + (x-y)x + y^2 = 0 \quad (2)$$

Since  $a, b, c$  and  $d$  are all distinct, is equivalent to the condition  $x, y \notin \{0, 1\}$  and  $x \neq y$ , taking into account that every circular permutation of a solution gives rise to the same 4-cycle, we have

$$N_{[4]}(q) = \frac{1}{4}q(q-1)C_q$$

where

$$C_q = |\{(x, y) \mid x, y \in \mathbb{F}_q \setminus \{0, 1\}, x \neq y, (1-x) + (x-y)x + y^2 = 0\}|$$

Assume  $q$  is odd. The affine conic  $(1-x) + (x-y)x + y^2 = 0$  has

$$q - \eta(-3) \quad (3)$$

rational points over  $\mathbb{F}_q$ .

This can be seen by noticing that the projective conic associated has  $q+1$  points and its points at infinity over  $\overline{\mathbb{F}}_q$  are  $[1, \omega, 0]$ ,  $[1, \bar{\omega}, 0]$  (where  $\omega \in \overline{\mathbb{F}}_q$  is a root of  $T^2 - T + 1$ ) which are rational if and only if  $\eta(-3) \neq -1$ .

From this number we have to subtract the number of rational points  $(x, y)$  verifying the conditions

$x, y \in \{0, 1\}$  or  $x = y$ . All these conditions give rise to the following (at most) 10 points over  $\overline{\mathbb{F}}_q$ :

$$(0, i), \quad (0, -i), \quad (1+i, 1), \quad (1-i, 1)$$

$$(1, \omega), \quad (1, \bar{\omega}), \quad (\omega, 0), \quad (\bar{\omega}, 0), \quad (\omega, \omega), \quad (\bar{\omega}, \bar{\omega})$$

where  $i$  is a root of  $T^2 + 1$ .

The number of the above points which are rational over  $\mathbb{F}_q$  is

$$2[1 + \eta(-1)] + 3[1 + \eta(-3)]$$

Subtracting the above quantity from (3), we obtain the statement.  $\square$

In general if  $q > k$  is odd, the

$$N_{[k]}(q) = \frac{1}{k} q(q-1) P_k(q)$$

where

$$P_k(x) = x^{k-3} + a_1 x^{k-4} + \cdots + a_{k-3}$$

and the coefficients

$$a_i = \eta(\alpha_{i,1}) b_{i,1} + \cdots + \eta(\alpha_{i,s_i}) b_{i,s_i}$$

for appropriate integers  $\alpha_{i,j}, \beta_{i,j}$ . There are ways to compute upper bounds for  $\alpha_{i,j}, \beta_{i,j}$  and  $s_i$ .

For example we can prove that for  $k$  odd

$$\alpha_{i,j} \leq \binom{k}{i}^k.$$

This allows (in principle) to calculate formulas for  $N_{[k]}(q)$ , for any value of  $k$  as long as one has calculated it for sufficiently many values of  $q$ .

For the general case, we can give some estimates. For example

$$P_k(q) = q^{k-3} + O(k^2 q^{k-4}).$$

Therefore

$$N_{[k]}(q) = \frac{1}{k} (q^{k-1} + O(k^2 q^{k-2})).$$

We conclude this section with two tables containing the formulas announced. The first holds for  $q$  odd<sup>6</sup>.

**Table 1: Permutation that move 6 elements in odd characteristics**

$N_{[6]}(q) = \frac{1}{6} q(q-1)$	$q^3 - 14q^2 + (68 - 6\eta(5) - 6\eta(50))q - (154 + 66\eta(-3) + 93\eta(-1) + 12\eta(-2) + 54\eta(-7)) + (-2\eta(-1)q^2 + (3 + 51\eta(-1))q + 4)(1 + \eta(9))$
$35?N_{[4 \ 2]}(q) = \frac{1}{8} q(q-1)$	$q^3 - (14 - \eta(2))q^2 + (71 + 12\eta(-1) + \eta(-2) + 4\eta(-3) - 8\eta(2))q - (148 + 100\eta(-1) + 24\eta(-2) + 44\eta(-3) + 40\eta(-7))$
$3?N_{[3 \ 3]}(q) = \frac{1}{18} q(q-1)$	$q^3 - 13q^2 + (62 + 9\eta(-1) + 4\eta(-3))q - (150 + 99\eta(-1) + 42\eta(-3) + 72\eta(-7))$
$N_{[2 \ 2 \ 2]}(q) = \frac{1}{48} q(q-1)$	$q^3 - (14 + 3\eta(-1))q^2 + (70 + 36\eta(-1) + 6\eta(-2))q - (136 + 120\eta(-1) + 48\eta(-2) + 8\eta(-3) + 16(1 - \eta(9)))$

**Table 2: Permutation that move 6 elements in even characteristics**

$N_{[6]}(2^n) = \frac{1}{6} 2^n (2^n - 1)(2^n - 3 - (-1)^n)(2^{2n} - (11 - (-1)^n)2^n + (41 + 7(-1)^n))$
$N_{[4 \ 2]}(2^n) = \frac{1}{8} 2^n (2^n - 1)(2^n - 3 - (-1)^n)(2^{2n} - 11 \cdot 2^n + 37 + (-1)^n)$
$N_{[3 \ 3]}(2^n) = \frac{1}{18} 2^n (2^n - 1)(2^n - 3 - (-1)^n)(2^{2n} - (10 - (-1)^n)2^n + 45 - 3(-1)^n))$
$N_{[2 \ 2 \ 2]}(2^n) = \frac{1}{48} 2^n (2^n - 1)(2^n - 2)(2^n - 4)(2^n - 8).$

<sup>6</sup>The symbols 35? (resp. 3?) means that for characteristic 3 and 5 (resp 3), the formula needs adjustment.

### 3 $k$ -cycle Permutation Polynomials with minimal degree

As we noticed above, if a permutation  $\sigma$  moves  $c$  elements of  $\mathbb{F}_q$  then

$\partial(f_\sigma) \geq q - c$ . For example the permutation polynomial of any  $k$ -cycle has degree at least  $q - k$ . For practical applications it is very important to produce permutation polynomials with "small" degree.

This justifies the definition. For a given conjugation class  $\mathcal{C}$ , let  $c = c(\mathcal{C})$  be the number of elements moved by any element of  $\mathcal{C}$ . Then

$$m_{\mathcal{C}}(q) = \{\sigma \in \mathcal{C} \mid \partial(f_\sigma) = q - c\}$$

(i.e. the permutations in  $\mathcal{C}$  such that their permutation polynomial has minimal degree)

The first elementary lower bound is:

**Proposition 2** *With the above notations, assume that  $q > 4$ . If  $m_{\mathcal{C}}(q) \neq 0$ , then*

$$m_{\mathcal{C}}(q) \geq \frac{1}{c^2}q(q-1).$$

**Proof.** Let  $\sigma_0$  be a permutation in  $m_{\mathcal{C}}(q)$ ,  $a, b \in \mathbb{F}_q$ ,  $a \neq 0$  and  $L_{a,b}$  is the linear transformation<sup>7</sup> of  $\mathbb{F}_q$ , then  $L_{a,b}^{-1}\sigma_0 L_{a,b} \in m_{\mathcal{C}}(q)$ .

It is easy to see that the number of distinct elements in  $m_{\mathcal{C}}(q)$  produced in this way equals to the index of the centralizer  $C_{\sigma_0}$  of  $\sigma_0$  in  $\mathbb{A}^1(\mathbb{F}_q)$ .

We claim that  $|C_{\sigma_0}| \leq c^2$ . Indeed assume first that  $\sigma_0(0) \neq 0$  and  $\sigma_0(1) \neq 1$ . Then if  $a, b \in \mathbb{F}_q$ ,  $a \neq 0$  and

$$\sigma_0 L_{a,b} = L_{a,b} \sigma_0,$$

for all  $x \in \mathbb{F}_q$ ,  $\sigma_0(ax + b) = a\sigma_0(x) + b$ .

If we substitute  $x = 0$ , we obtain  $\sigma_0(b) - b = a\sigma_0(0) \neq 0$ . So  $b \in S_{\sigma_0}$ . If we substitute  $x = 1$ , we obtain  $\sigma_0(a + b) - (a + b) = a(\sigma_0(0) - 1) \neq 0$ . So  $a + b \in S_{\sigma_0}$ .

Finally  $b \in S_{\sigma_0}$  can be chosen in at most  $c$  distinct ways and for each fixed  $b$ ,  $a \in S_{\sigma_0} - b$  can also be chosen in at most  $c$  distinct ways.

If  $\sigma_0(0) = 0$  or  $\sigma_0(1) = 1$ , then one can chose  $a_0, b_0 \in \mathbb{F}_q$  such that

$$\sigma(b_0) \neq b_0 \quad \sigma(a_0 + b_0) \neq a_0 + b_0.$$

and replace  $\sigma_0$  by  $\sigma_1 = L_{a_0, b_0}^{-1}\sigma_0 L_{a_0, b_0}$ .

Therefore if not  $m_{\mathcal{C}}(q)$  is not empty, we have that

$$|m_{\mathcal{C}}(q)| \geq \frac{1}{c^2}q(q-1)$$

and this concludes the proof. □

We can improve the previous result for some  $k$ -cycles.

**Theorem 1** *If  $q \equiv 1 \pmod{k}$  then*

$$|m_{[k]}(q)| \geq \frac{\varphi(k)}{k}q(q-1).$$

---

<sup>7</sup>The group  $\mathbb{A}^1(\mathbb{F}_q)$  of linear transformations of  $\mathbb{F}_q$  consists on those maps  $L_{a,b} : \mathbb{F}_q \rightarrow \mathbb{F}_q, x \mapsto ax + b$ . This group has order  $q(q-1)$ .

Furthermore, using algebraic geometry, we prove the upper bound

**Theorem 2** If  $q \geq k$ , then

$$|m_{[k]}(q)| \leq \frac{(k-1)!}{k} q(q-1).$$

In principle one should be able to prove similar estimates for other conjugation classes of permutations. However, technical problems arise.

For permutations that move less than 5 elements we study the problem in detail.  
For example

$$|m_{[4]}(q)| = \frac{1}{4} q(q-1) K_q$$

where

$$K_q = 1 + \eta(-1) + \begin{cases} 4 & \text{if } q \equiv 1 \pmod{5} \\ 0 & \text{otherwise} \end{cases}$$

The formulas for 5-cycles are more complicated.

Next we restrict our attention to prime fields.

We deal with the problem of studying how sharp is the previous estimate. We provide evidence of the fact that the estimate is essentially the best possible. This will be a consequence of the Chebotarev Density Theorem. More precisely, let  $a$  be an integer with  $1 \leq a \leq (k-1)!$  and consider the function:

$$F_{a,k}(x) = \{p \leq x \mid p \text{ prime}, p \geq k, N_{[k]}(p) = \frac{a}{k}(p(p-1))\}.$$

If  $\pi(x)$  is the number of primes up to  $x$ ,  $F_{a,k}(x)/\pi(x)$  measures the probability that a  $\mathbb{F}_p$  has exactly  $\frac{a}{k}(p(p-1))$   $k$ -cycle permutation polynomial with minimal degree  $p-k$ . It is natural to expect that this probability tends to a finite limit as  $x$  tends to  $\infty$ . That is to say that one expects

$$F_{a,k}(x) \sim c_{a,k} \pi(x),$$

Where the condition  $c_{a,k} = 0$  should be interpreted as to say that

$F_{a,k}(x) = 0$ . We conjecture that  $c_{(k-1)!,k} \neq 0$ .

In general  $c_{(k-1)!,k} \leq 1/(k-1)!!$ . However it is difficult

to compute. We can show that  $c_{3!,4} = 1/8$  and

$c_{5!,6} \leq 1/(288 \cdot 108!)$ . Furthermore  $c_{4!,5} = 1/60000$ .

Here is a list of the first few primes that

enjoy the property  $N_{[5]}(p) = 24(p(p-1))/5$ .

471301	1695341	2134241	3676831	4845761	4938181
5892011	8276131	8748281	9589201	10922651	10996471
11208671	11622601	11683751	11794661	13910161	13950281
14679361	15379361				

Although  $1/30000 = 0.000016\%$  seems a small a small proportion of primes. It is surprisingly high respect to the bound  $c_{4!,5} \leq 1/24! \sim 1.6 \cdot 10^{-24}$ .

The bulk of our technique is based on the reduction of the problem of determining the solutions over  $\mathbb{F}_q$  of a family of homogeneous equations defined over  $\mathbb{Z}$ .

We bound this number by the number of solutions over  $\overline{\mathbb{Q}}$  which after proving finiteness is estimated with the Bezout Theorem. The algebraic structure of the minimal field which contains all the solutions and its Galois group over  $\mathbb{Q}$  will be used to attack the conjecture.

The results described in this note will appear in two papers:

1. C. Malvenuto & F. Pappalardi, *Enumerating Permutation Polynomials I: permutations with non maximal degree*, Preprint
2. C. Malvenuto & F. Pappalardi, *Enumerating Permutation Polynomials II: k-cycles with minimal degree*, Preprint

Claudia Malvenuto

Dipartimento di Scienze dell'Informazione  
 Università degli studi "La Sapienza"  
 Via Salaria, 113  
 I-00198, Roma - ITALY  
 claudia@dsi.uniroma1.it

Francesco Pappalardi

Dipartimento di Matematica  
 Università Roma Tre  
 Largo S. L. Murialdo, 1  
 I-00146, Roma - ITALY  
 pappa@mat.uniroma3.it

# CAYLEY GRAPH ENUMERATION

Marni Mishna<sup>1</sup>

## Abstract

Pólya's Enumeration Theorem is a powerful method for counting distinct arrangements of objects. J. Turner noticed that circulant graphs have a sufficiently algebraic structure that Pólya's theorem can be used to determine the number of non-isomorphic circulants of order  $p$  for prime  $p$ . Recent results on CI-groups, in combination with Turner's method, are used to enumerate a larger collection of circulants and Cayley graphs on  $\mathbb{Z}_p^2$ , with an outline for the  $\mathbb{Z}_p^n$  case.

## Résumé

Le Théorème d'Énumération de Pólya est une puissante méthode pour énumérer les arrangements distincts d'objets. J. Turner a remarqué que les graphes circulants ont une structure algébrique permettant l'utilisation du théorème de Pólya pour la détermination du nombre de circulants d'ordre  $p$  non isomorphes, pour un nombre premier  $p$ . Des résultats récents sur les groupes CI sont utilisés en combinaison avec la méthode de Turner pour l'énumération d'une large classe de circulants et de graphes de Cayley sur  $\mathbb{Z}_p^2$ , et pour l'esquisse du cas  $\mathbb{Z}_p^n$ .

## 1 The Problem

Pólya's theorem of enumeration, when it first became widely appreciated in the early 1960s, served as a key tool for many graph isomorphism problems. In 1967 J. Turner determined that circulant graphs of prime order, a class of Cayley graphs, was well suited to this approach. The graphs Turner considered possess a particular property known as the Cayley Isomorphism property and recent work in this area has found more families with this characteristic, thereby opening up the possibility of applying his methods to these new families. This work (and the extended original [10]) does precisely that; it uses Turner's approach to determine enumerative formulas for classes of CI-groups.

Cayley graphs are defined with respect to a group.

**DEFINITION.** A *Cayley subset*  $S$  of a group  $G$  is an inverse closed subset ( $s \in S \implies s^{-1} \in S$ ) of  $G$  not containing the identity. A *Cayley Graph* is represented by  $X(G; S)$  where  $G$  is a group, and  $S$  is a Cayley subset of  $G$ , also known as the *connection set*. The graph  $X(G; S)$  has vertex set  $G$  and edge set  $\{(g_1, g_2) | g_1 = g_2s, s \in S\}$ .

The cyclic group on five elements,  $\mathbb{Z}_5$ , admits four Cayley graphs, three of which are distinct, as seen in Figure 1.

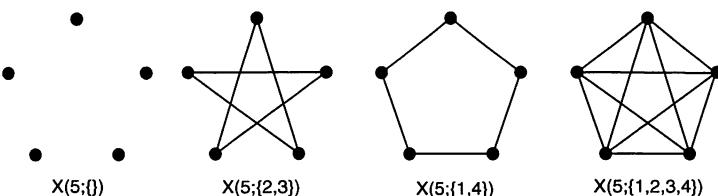


Figure 1: THE COMPLETE SET OF CAYLEY GRAPHS ON  $\mathbb{Z}_5$

<sup>1</sup>The author acknowledges the support of NSERC and Simon Fraser University; Correspondence to: mmishna@math.sfu.ca

**DEFINITION.** A *circulant* is a Cayley graph on a cyclic group. The circulant  $X(\mathbb{Z}_n; S)$  is denoted by simply  $X(n; S)$ . Thus, the Cayley graphs of the previous example are also circulants.

## 2 CI-Groups

Turner's enumeration results rely on the fact that when two circulants are isomorphic there is also a group automorphism which maps the respective Cayley subsets to each other. This is a property of cyclic groups of prime order, and is formalized with the notion of a CI-group.

**DEFINITION.** Let  $G$  be a finite group, and  $X = X(G; S)$  be a Cayley graph on  $G$ .  $S$  satisfies the *Cayley isomorphism property*, or is a *CI-subset*, if whenever  $X$  is isomorphic to  $X' = X(G; S')$ , there exists a group automorphism  $\alpha$  of  $G$  such that  $\alpha$  is also a graph isomorphism from  $X$  to  $X'$ . If every Cayley subset of  $G$  is a CI-subset, then say that  $G$  satisfies the *Cayley Isomorphism Property*, or succinctly,  $G$  is a *CI-group*.

This property is useful from several viewpoints, including enumeration. Turner [16] proved that cyclic groups of prime order were CI-groups. This led Ádám to a conjecture on the nature of cyclic CI-groups and finally M. Muzychuk resolved the issue.

**THEOREM 1 (MUZYCHUK)** *The cyclic groups which are CI-groups are precisely those of order  $n$  where  $n$  is 8, 9, 18 or  $n, n/2$  or  $n/4$  is odd and squarefree.*

Since the automorphism group of the cyclic group of order  $n$  is isomorphic to the multiplicative group of units modulo  $n$ , that is  $\mathbb{Z}_n^*$  [14], the result implies that for  $n$  in this set whenever  $X(n; S)$  is isomorphic to  $X(n; T)$ , there is an element  $a \in \mathbb{Z}_n^*$  such that  $S = aT$ .

What of the circulants of other orders? Relying on the enumerative technique presented here, nothing can be said of orders falling outside Muzychuk's condition. Although, M. Kiln, V. Liskovets and R. Pöschel. used S-Ring theory techniques to determine formulas for circulants of prime-squared order [9]. Now, the recent proof [11] of the following conjecture of Toida [15] does allow Turner's technique to count a family of circulants with restricted Cayley subsets, but no restriction on order, the so called unit-circulants [10].

**THEOREM 2 (DOBSON, MORRIS)** *For any  $S \subseteq \mathbb{Z}_n^*$ , whenever  $X(n; S)$  is isomorphic to  $X(n; T)$ , there exists an  $a \in \mathbb{Z}_n^*$  satisfying  $T = aS$ .*

C. Godsil [5] showed that the group  $\mathbb{Z}_p^2$ ,  $p$  prime is a CI-group. This CI-group is also considered in this work. An explicit formula for the number of isomorphism classes of Cayley graphs of order  $p$  on these groups is presented.

T. Dobson [3] and M.-Y. Xu [17] independently showed that The group  $\mathbb{Z}_p^3$ ,  $p$  prime is a CI-group and recently [11], it has been shown by J. Morris and T. Dobson that  $\mathbb{Z}_p^4$  is also a CI-Group. The reader, at this point, may hypothesize that  $\mathbb{Z}_p^n$  is a CI-Group for all  $n$ . In fact, L. Nowitz showed [12] that  $\mathbb{Z}_2^6$  is not a CI-Group. However, for all  $n$  such that it is true, the methods presented in this work will determine the number of isomorphism classes of a given order. For an excellent summary of CI-Groups see B. Alspach's survey [1].

### 3 Circulants of Prime Order

Circulants have been of interest in several fields. They are Cayley graphs on the simplest of groups and thus may provide direction and insight into Cayley graphs on other groups, particularly finite groups. All finite vertex-transitive graphs of prime order are circulants, hence the study of vertex transitive graphs can gain from the study of circulants.

Turner noticed that Theorem 1 provides the necessary information about the structure of circulants to appeal to Pólya's enumeration theorem to develop a formula for non-isomorphic circulants. At the time it was only known to be valid for primes, and hence his formula was restricted to the prime case.

Let  $G$  be a permutation group acting on a set  $\Omega$ . Consider the disjoint cycle decomposition of  $\sigma \in G$ . Suppose it contains precisely  $b_k$  cycles of length  $k$ , for  $1 \leq k \leq |\Omega|$ . Define a monomial  $\pi(\sigma)$  associated with the decomposition as follows:  $\pi(\sigma) = x_1^{b_1} x_2^{b_2} \dots x_m^{b_m}$ , where  $m = |\Omega|$ . Note that  $b_1 + 2b_2 + \dots + mb_m = m$ .

**DEFINITION.** The *cycle index*  $Z(G, \Omega)$  of the permutation group  $G$  acting on  $\Omega$  is defined to be the polynomial in indeterminates  $x_1, x_2, \dots, x_m$

$$Z(G, \Omega) = \frac{1}{|G|} \sum_{\sigma \in G} \pi(\sigma).$$

**DEFINITION.** Let  $a, n \in \mathbb{Z}$  with  $\gcd(a, n) = 1$ . The *order of  $a$  mod  $n$*  is the least positive integer  $k(a)$  such that  $a^{k(a)} \equiv 1 \pmod{n}$  and we denote this by  $\text{ord}_n(a)$ . Notice that this is equivalent to saying that  $a$  has multiplicative order  $k(a)$  in the group  $\mathbb{Z}_n^*$ .

Pólya's Theorem brings together the three components for counting in a very simple way. To establish context, first recall the Cauchy Fröbenius Lemma (also known by the Burnside Lemma). If  $G$  is a finite group of transformations acting on a finite set  $\Omega$  of objects, and two objects are considered equivalent if one is transformed into the other by a transformation in  $G$ , then the number of inequivalent objects is  $\frac{1}{|G|} \sum_{\sigma \in G} \text{fix}(\sigma)$ , where  $\text{fix}(\sigma)$  is the number of points in  $\Omega$  fixed by  $\sigma$ .

Pólya's Theorem generalizes this idea. Let  $D$  and  $R$  be finite sets. Let  $R^D$  denote the set of all functions from  $D$  to  $R$ , and let  $G$  be a permutation group acting on  $D$ . For each  $\sigma \in G$  define a permutation  $\bar{\sigma}$  acting on  $R^D$  by

$$\bar{\sigma}(f)(x) = f(\sigma(x))$$

for all  $f \in R^D$  and all  $x \in D$ . The mapping taking  $\sigma$  to  $\bar{\sigma}$  is a homomorphism of  $G$  to a permutation group  $\bar{G}$ .

**THEOREM 3 (PÓLYA)** With  $D$ ,  $R$ ,  $G$ ,  $\bar{G}$ , and  $R^D$  as per the above discussion, the number of inequivalent  $f \in R^D$  under  $\bar{G}$  is determined by evaluating the cycle index of  $G$  with each variable set to  $|R|$ .

The theorem is actually much stronger. Good discussions exist by De Bruijn [2] and Read [13]. To summarize, if the weight of an element  $y$  in  $R$  is  $w(y)$ , in  $\mathbb{N}$ , then the total weight off  $R$  is then  $g(u) = \sum_{y \in R} u^{w(y)}$ . The total weight of inequivalent elements in  $R^D$  under  $\bar{G}$  is obtained by substituting  $g(u^i)$  for  $x_i$  in the cycle index of  $G$ . Thus the basic case, the number of orbits of the permutation group  $\bar{G}$ , is given by setting all of the weights equal to zero. This more powerful version can be used to determine the number of graphs of a given order with a given sub-property, such as vertex valency.

To apply Pólya's Theorem to count the number of circulant graphs of order  $p$ ,  $p$  an odd prime, let the domain  $D$  be the set of unordered pairs

$$\{\{1, -1\}, \{2, -2\}, \dots, \left\{\frac{p-1}{2}, \frac{p+1}{2}\right\}\}$$

with the range  $R$  chosen to be the set  $\{0,1\}$ . A function  $f_S$  in  $R^D$  corresponds to a circulant graph  $X(p; S)$  in the following way. The set  $\{i, -i\}$  is in  $S$  if and only if  $f_S(\{i, -i\}) = 1$ . The permutation group acting on  $D$  under consideration is  $\mathbb{Z}_p^*$  with the action of multiplication. Since multiplication by  $a$  is the same as multiplication by  $-a$  with respect to  $D$ , in this case the permutation group is  $\mathbb{Z}_p^*/\{1, -1\} \cong \mathbb{Z}_{\frac{p-1}{2}}$ .

**THEOREM 4 (TURNER)** *Let  $p$  be a prime. The number of circulant graphs of order  $p$ , up to isomorphism, is*

$$\frac{2}{p-1} \sum_{d \mid \frac{p-1}{2}} \phi(d) 2^{\frac{p-1}{2d}},$$

where  $\phi$  is Euler's totient function.

**PROOF:** Theorem 1 states that two circulant graphs  $X(Z_p; S)$ , and  $X(Z_p; T)$  are isomorphic if and only if their connection sets satisfy  $S = aT$  for some  $a \in Z_p^*$ . Consider the action of  $a$  in the cyclic group of order  $(p-1)/2$  on  $D$ . This maps the set  $\{i, -i\}$  to  $\{ia, -ia\}$ . This induces an action  $\bar{a}$  on  $R^D$  defined by  $\bar{a}(f)(x) = f(ax)$ . Notice that  $f(\{ai, -ai\}) = 1$  if and only if  $\bar{a}(f)(\{i, -i\}) = 1$  for each  $i \in Z_p^*$ . Since  $\bar{a}(f) \in R^D$ ,  $a(f_S) = f_T$  for some  $T \subseteq D$ , with  $S = aT$ . Thus, counting the number of orbits of this group simultaneously counts the number of isomorphism classes. The cyclic index of the latter group [2] is

$$\frac{2}{p-1} \sum_{d \mid \frac{p-1}{2}} \phi(d) x_d^{\frac{p-1}{2d}}.$$

Now substitute  $2 = |R|$  for each variable, according to Pólya's Theorem, and obtain the desired result. ■

To illustrate this result consider the circulants on five vertices. The formula yields  $\frac{2}{4}(\phi(1)2^2 + \phi(2)2) = 3$ , as is consistent with Figure 1.

To illustrate the power of the formula, consider a more impressive result. Let  $p = 53$ . The cycle index for this case is  $\frac{2}{52}(x_1^{26} + x_2^{13} + 12x_{13}^2 + 12x_{26})$ , and it follows that the number of non-isomorphic circulant graphs of order 53 is  $\frac{1}{26}(2^{26} + 2^{13} + 12 \cdot 2^2 + 12 \cdot 2) = 2\ 581\ 428$ .

Turner's key bijection is valid whenever a group is a CI-group. In this case the problem of determining an enumeration formula amounts to determining the relevant group action and cycle index. This is the focus of the remaining work.

### 3.1 A Note about Digraphs

A Cayley digraph is a directed version.

**DEFINITION.** A *Cayley digraph*  $X(G; S)$  is defined on a group  $G$  and a set  $S \subseteq G \setminus e$ . It has vertex set  $G$  and there is a directed edge from  $g_1$  to  $g_2$  if and only if  $g_2 = g_1 s$  for

some  $s \in S$ . The theorems used here each apply also to the digraph case. A simpler version of Turner's bijection can be used as now the defining subsets have no restriction. Thus, an enumeration formula for Cayley digraphs on the group  $G$  comes directly from the cycle index of the automorphism group of  $G$  acting on  $G \setminus e$ , where  $e$  is the group identity. In all of the following cases, the digraph case can be constructed [10]. The case of prime circulant digraphs is such an example.

**THEOREM 5** *The number of non-isomorphic circulant digraphs of order  $p$ ,  $p$  prime, is*

$$\frac{1}{p-1} \sum_{d|p-1} \phi(d) 2^{(p-1)/d}.$$

Hence, the number of non-isomorphic circulant digraphs of order 29 is  $\frac{1}{28} \sum_{d|28} \phi(d) 2^{\frac{28}{d}} = 9\ 587\ 580$ .

## 4 Orders Satisfying Muzychuk's Condition

Theorem 1, implies enumeratation for a larger collection of circulants is possible using Pólya's theorem and Turner's method. Thus, we can enumerate the circulant of order 8, 9, 18 and  $2^e m$  where  $e \in \{0, 1, 2\}$  and  $m$  is odd and square-free. Here, consider only the case where  $m$  is odd and square-free. First consider circulants that have an order that is a product of two distinct primes. It is straightforward to generalize this to the larger case.

Theorem 1 tells us, that as in the case of prime order circulants, the group action to consider is multiplication by an element  $a$  in  $\mathbb{Z}_{pq}^*$ . Thus, as per the earlier discussion, the formula is a consequence of the cycle index of  $\mathbb{Z}_{pq}^*$  acting on the set  $D = \{\{1, -1\}, \dots, \{\frac{pq-1}{2}, \frac{pq+1}{2}\}\}$  by multiplication.

**THEOREM 6** *The cycle index of  $\mathcal{P}_{pq}$  acting on  $D$  is*

$$\mathcal{Z}(\mathcal{P}_{pq}, D) = \frac{2}{p-1} \sum_{d|p-1} \sum_{e|q-1/2} \phi(d)\phi(e) x_{L(d,e)}^{\frac{\phi(pq)}{2L(d,e)}} x_{H(d)}^{\frac{p-1}{2H(d)}} x_{H(e)}^{\frac{q-1}{2H(e)}}$$

where

$$H(a) = \begin{cases} a & a \text{ is odd} \\ a/2 & \text{otherwise} \end{cases}$$

and

$$L(d_1, d_2) = \begin{cases} \text{lcm}(d_1, d_2)/2 & d_1, d_2 \text{ both even and divisible by the same power of 2} \\ \text{lcm}(d_1, d_2) & \text{otherwise.} \end{cases}$$

**PROOF:(sketch)** The most important observation is that if  $m = m_1 m_2 \dots m_t$ , where the  $m_i$  are pairwise co-prime, then  $\mathbb{Z}_m^* \cong \mathbb{Z}_{m_1}^* \times \mathbb{Z}_{m_2}^* \times \dots \times \mathbb{Z}_{m_t}^*$ .

The result follows upon examining each element of  $\mathbb{Z}_{pq}^*$  as a pair from  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ . The order of the pair is a function of the order of each element in its respective group. The number of each order follows is similarly dependant on the number of elements of a given order in the smaller groups.



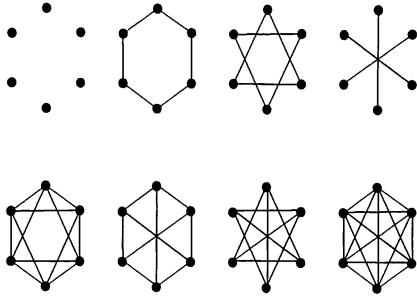


Figure 2: THE EIGHT ISOMORPHISM CLASSES OF CIRCULANTS OF ORDER 6

**COROLLARY 7** Let  $p, q$  be prime. The number of circulant graphs of order  $n = pq$  up to isomorphism is

$$\frac{2}{\phi(pq)} \sum_{d|p-1} \sum_{\substack{e|q-1 \\ e \text{ even}}} \phi(d)\phi(e/2) 2^{\frac{p-1}{2H(d)}} 2^{\frac{q-1}{e}} 2^{\frac{\phi(pq)}{2L(d,e)}}.$$

Consider now an example for some small primes. We can calculate the number of circulants of order 6. According to the corollary, this number is  $2^3 = 8$  since the cycle index is  $x_1^3$ , verified by Figure 2.

The enumeration formula can be generalized with some help from notation. Let  $\text{lcm}(\{a_i\})$  denote the least common multiple of a set. That is, for each  $a \in \{a_i\}$ ,  $a|\text{lcm}(\{a_i\})$ , and it is the smallest integer with this property. This induces a generalization of the  $L$  function from the previous theorem.

**THEOREM 8** The number of circulants of order  $p_1 p_2 \dots p_t$  with each  $p_i$  a distinct odd prime is

$$\frac{2}{\phi(p_1 p_2 \dots p_t)} \sum_{\substack{(d_1, d_2, \dots, d_t) \\ d_i|p_i-1 \\ d_i|(p_i-1)/2}} \phi(d_1) \dots \phi(d_t) \prod_{\substack{I \subseteq \{1, 2, \dots, t\} \\ I \neq \emptyset}} 2^{\frac{P(I)}{2L(\{d_i\}_{i \in I})}}.$$

and where  $P(I) = \prod_{p \in I} (p-1)$ .

## 5 Cayley Graphs over $\mathbb{Z}_p \times \mathbb{Z}_p$ with $p$ prime.

The strategy of appealing to Pólya's Theorem to count isomorphism classes can be used to count the Cayley graphs on  $\mathbb{Z}_p \times \mathbb{Z}_p$ , since this is another family of CI-groups. However, the automorphism group of  $\mathbb{Z}_p \times \mathbb{Z}_p$  is the general linear group and hence the methods to determine the cycle index are quite different than for determining the cycle index of a cyclic group.

Consider  $\mathbb{Z}_p \times \mathbb{Z}_p$  as an additive group, and consequently think of  $\mathbb{Z}_p \times \mathbb{Z}_p$  as a two dimensional vector space  $V$  over  $\mathbb{Z}_p^*$ . In general, consider  $\mathbb{Z}_p^n$  to be an  $n$ -dimensional vector space over  $\mathbb{Z}_p$ . With this view in mind, the group of automorphisms is clear.

**THEOREM 9** The automorphism group of  $\mathbb{Z}_p^n$  is isomorphic to  $GL(n, p)$ , the group of invertible  $n \times n$  matrices over  $\mathbb{Z}_p$ .

The problem of determining the cycle structure of linear transformations over a finite field was first tackled by Kung in [7]. Recently, Fripertinger [4] calculated the Pólya cycle index of the general linear group (as well as affine and projective groups) and used the cycle index to enumerate isometry classes of linear codes. The cycle index as Fripertinger calculated it is suitable to enumerate digraphs, but requires modification to be useful for the undirected graph case. Furthermore, since  $n = 2$  is specified, the expression has a far more explicit, though less compact, form.

## 5.1 Rational Normal Form

In this context matrices are viewed as permutations. Recall that permutations which are conjugate have the same cycle structure. This is the essence of our strategy. As each matrix is in a single conjugacy class, we can determine the cycle index by determining the size of each conjugacy class and the cycle structure of a representative. This section defines the rational normal form and illustrates its suitability as a representative.

The reader not immediately familiar with rational normal form is referred to Kung's paper for a quick summary or Herstein's book [6] for a more thorough treatment. For the purpose of brevity the different classes of rational normal form of dimension 2 are directly presented.

**DEFINITION.** A *partition* of  $n$  is an unordered set of integers which sum to  $n$ . Summarize a partition with a partition vector  $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots)$ , a sequence of non-negative integers with finitely many non-zero terms such that  $n = \lambda \cdot (1, 2, 3, \dots) = 1\lambda_1 + 2\lambda_2 + 3\lambda_3 + \dots$

To construct an  $n \times n$  matrix in rational normal form one needs a collection of irreducible polynomials  $f_1, f_2, \dots, f_t$  of degrees  $d_1, d_2, \dots, d_t$  each less than  $n$ , respectively, and a set of  $t$  partitions  $\lambda^{f_i}$ , or  $\lambda^{(i)}$ , such that  $\sum_i |\lambda^{(i)}|d_i = n$ . The possibilities are small when  $n = 2$ . The irreducible polynomials of degree at most two are of the form  $x^2 + ax + b$  and  $x - a$ , with  $a, b \in \mathbb{Z}_p$ .

The rational normal form is a block diagonal matrix of smaller hypercompanion matrices. The hypercompanion matrices are determined by an irreducible polynomials and a partition. The notation for a rational normal form decomposition is a sequence of such matrices denoted  $D[f(x), \lambda^f]$ . The possible rational normal forms for matrices of degree two, are summarized in Table 5.1. Kung [7] determined the size of a conjugacy class in  $GL(n, p)$ . Table 5.1 also summarizes these values for all possible cases when  $n = 2$ .

Minimal polynomial	Rational normal form	Matrix	Number in a class
$(x - a)$	$D[x - a, (2)]$	$\begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$	1
$(x - a)^2$	$D[x - a, (0, 1)]$	$\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$	$p^2 - 1$
$(x - a)(x - b), a \neq b$	$D[x - a, (1)]D[x - b, (1)]$	$\begin{pmatrix} 0 & 1 \\ -b & -a \end{pmatrix}$	$p(p + 1)$
$x^2 + ax + b$ , irreducible	$D[x^2 + ax + b, (1)]$	$\begin{pmatrix} 0 & 1 \\ -b & -a \end{pmatrix}$	$p^2 - p$

Table 1: Conjugacy Classes in  $GL(2, p)$

## 5.2 The Cycle Index of $GL(2, p)$

Having illustrated the classes, it remains to discover their cycle structure. The action of a block diagonal matrix can be decomposed into the action of its components. In this

case, the cycle structure is a product of the actions of matrices with irreducible minimal polynomials.

**DEFINITION.** Given  $A \in GL(n, p)$ , define the *order* of  $A$  to be the least integer  $k > 0$  such that  $A^k = I_n$ , the  $n \times n$  identity matrix. Let  $f \in \mathbb{F}_p[x]$  be a polynomial of degree  $m$  such that  $f(0) \neq 0$ . The *order* of  $f$ , denoted  $\text{ord}(f)$  or  $\text{ord}(f(x))$ , is the least integer  $k > 0$  such that  $f(x)$  divides  $x^k - 1$ . It can be shown that some  $k \leq p^m - 1$  exists. If  $f(0) = 0$ , define the order of  $f$  by writing  $f(x) = x^n g(x)$  such that  $n \in \mathbb{N}$ ,  $g(0) \neq 0$ , and define  $\text{ord}(f) = \text{ord}(g)$ .

**THEOREM 10** *Let  $V$  be cyclic with respect to linear transformation  $A$ . Let  $f(x)$  be an irreducible, monic polynomial and the minimal polynomial of  $A$ . As a permutation of the non-trivial elements of  $V$ ,  $A$  is a product of cycles of a length  $\text{ord}(f)$ .*

This is sufficient to give the cycle index for  $GL(2, p)$  on  $\mathbb{Z}_p \times \mathbb{Z}_p$ . However, to determine the enumerative result for graphs requires a modification of the action. As in the circulant case we are looking at the group of automorphisms acting on the quotient group, in this case  $(\mathbb{Z}_p \times \mathbb{Z}_p \setminus (0, 0)) / \{(1, 1), (-1, -1)\}$ , which we shall denote  $\mathbf{Z}_p$ . In the circulant case it simplified matters to pare down the automorphism group. In this case, it is simpler to use  $GL(2, p)$ . We have already determined the nature and size of the conjugacy classes. However, the cycle structure of a transformation is different when  $x$  and  $-x$  of  $\mathbb{Z}_p^2$  are identified.

**LEMMA 11** *Consider a monic, irreducible,  $f \in \mathbb{F}_p[x]$  with  $k = \text{ord}(f)$ . Let  $k'$  be the smallest integer such that  $f(x)$  divides  $x^{k'} + 1$ , if it exists. Then  $k' = k/2$  if  $k$  is even and does not exists if  $k$  is odd.*

This  $k'$  serves as a new form of order. If  $V$  is an  $n$ -dimensional subspace cyclic with respect to  $A \in GL(n, p)$ , then the length of a cycle of  $A$  acting on an element of  $V$  where  $x$  and  $-x$  are identified is  $k = \text{ord}(A)$  if  $k$  is even and  $k/2$  if  $k$  is odd. Capture this action with the notation

$$H(x) = \begin{cases} x & x \text{ is odd} \\ x/2 & \text{otherwise.} \end{cases}$$

**THEOREM 12** *The cycle index of  $GL(2, p)$  acting on  $\mathbf{Z}_p$  is*

$$\mathcal{Z}(GL(2, p), \mathbf{Z}) \tag{1}$$

$$= \frac{1}{(p^2 - p)(p^2 - 1)} \sum_{d|(p-1)} \phi(d) x_{H(d)}^{\frac{p^2-1}{2H(d)}} \tag{2}$$

$$+ \frac{1}{2(p^2 - 1)} \sum_{\substack{d|p^2-1 \\ p \not\equiv 1 \pmod{d}}} \phi(d) x_{H(d)}^{\frac{p^2-1}{2H(d)}} \tag{3}$$

$$+ \frac{1}{p^2 - p} \sum_{d|p-1} \phi(d) x_{H(d)}^{\frac{p-1}{H(d)}} x_{pH(d)}^{\frac{p-1}{H(d)}} \tag{4}$$

$$+ \frac{1}{(p-1)^2} \sum_{d|p-1} \sum_{\substack{e|p-1 \\ e \leq d}} \phi'(d, e) x_{H(d)}^{\frac{p-1}{H(d)}} x_{H(e)}^{\frac{p-1}{H(e)}} x_{L(d,e)}^{\frac{(p-1)^2}{L(d,e)}} \tag{5}$$

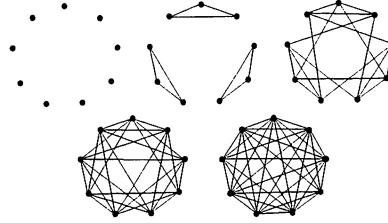


Figure 3: THE FIVE ISOMORPHISM CLASSES OF  $X(\mathbb{Z}_3 \times \mathbb{Z}_3, S)$

where  $\phi'(d, e) = \phi(d)\phi(e)$  if  $d \neq e$  and  $\phi(d)^2 - 1$  otherwise, and

$$L(d, e) = \begin{cases} \text{lcm}(d, e)/2 & \text{if both } d \text{ and } e \text{ are even and are divisible} \\ & \text{by the same powers of two} \\ \text{lcm}(d, e) & \text{otherwise.} \end{cases}$$

**PROOF:** To determine the cycle index sum over the different types of rational normal forms. An excellent reference for the nature of the orders of each type is Lidl's and Niederreiter's work [8]. The first summand, (2), gives us the cycle inventory for elements with the rational normal form  $A = D[x - a, (2)]$ , the second (3), transformations with rational normal form  $D[x^2 + ax + b, (1)]$  for irreducible polynomials  $x^2 + ax + b$ ; the third (4), accounts for the matrices with rational normal form  $A = D[x - a, (0, 1)]$ . The last, and most complicated comes from matrices with the remaining normal form,  $D[x - a, (1)]D[x - b, (1)]$  when  $a \neq b$ . We do not include the case when  $a = b$  since they are counted in (4). This gives us (5).

■

**COROLLARY 13** *The number of non-isomorphic Cayley graphs on  $\mathbb{Z}_p \times \mathbb{Z}_p$  for  $p > 2$  prime is*

$$\begin{aligned} & \frac{1}{(p^2 - p)(p^2 - 1)} \sum_{d|(p-1)} \phi(d) 2^{\frac{p^2-1}{2H(d)}} \\ & + \frac{1}{2(p^2 - 1)} \sum_{\substack{d|p^2-1 \\ p \not\equiv 1 \pmod{d}}} \phi(d) 2^{\frac{p^2-1}{2H(d)}} \\ & + \frac{1}{p^2 - p} \sum_{d|p-1} \phi(d) 2^{\frac{p-1}{H(d)}} 2^{\frac{p-1}{H(d)}} \\ & + \frac{1}{(p-1)^2} \sum_{d|p-1} \sum_{\substack{e|p-1 \\ e \leq d}} \phi'(d, e) 2^{\frac{p-1}{H(d)}} 2^{\frac{p-1}{H(e)}} 2^{\frac{(p-1)^2}{L(d, e)}}. \end{aligned}$$

The cycle index for  $GL(2, 3)$  acting on  $\mathbf{Z}_3$  is  $\frac{1}{48}(12x_2x_1^2 + 12x_4 + 6x_2^2 + 16x_3x_1 + 2x_1^4)$  hence the number of non-isomorphic Cayley graphs on  $\mathbb{Z}_3 \times \mathbb{Z}_3$  is in accordance with Figure 3,  $\frac{1}{48}(12 \cdot 2 \cdot 2^2 + 12 \cdot 2 + 6 \cdot 2^2 + 16 \cdot 2 \cdot 2 + 2 \cdot 2^4) = 5$ .

## 6 Conclusions

As more groups are identified as CI-groups, this method can be revisited as a method of enumeration. Since currently most energy in the CI-group problem is directed at products of cyclic groups, a better form for  $\mathcal{Z}(GL(n, p), \mathbb{Z}_p^n)$  would produce more satisfying enumeration results. For every case that Pólya's Enumeration Theorem is used, it may be worthwhile to investigate other uses of the cycle index apart from straight enumeration. For example, one could count the number of distinct edge colourings using  $k$  colours by substituting  $k + 1$  for each  $x_i$  in the cycle index. A different, though likely interesting, epilogue to these results here would be the computation of asymptotics. Of particular interest may be to determine the percentage of circulants which are unit circulants for a given number of factors. Crude constructive algorithms are suggested by these formulas; perhaps better are possible.

The author wishes to thank B. Alspach for presenting the problem and a careful reading the original work and also K. Meagher for her useful comments.

## References

- [1] B. Alspach, *Isomorphism and Cayley graphs on abelian groups*, Graph Symmetry (1997), 1–22.
- [2] E. Beckenbach (ed.), *Applied combinatorial mathematics*, John Wiley and Sons, New York, 1964.
- [3] E. Dobson, *Isomorphism problem for Cayley graphs of  $\mathbb{Z}_p^3$* , Discrete Math **147** (1995), 87–94.
- [4] H. Fripertinger, *Cycle indices of linear, affine and projective groups*, Lin. Alg. Apps. **263** (1997), 133–156.
- [5] C. D. Godsil, *On Cayley graph isomorphisms*, Ars Combin. **15** (1983), 231–246.
- [6] I. N. Herstein, *Topics in algebra*, Xerox College Publishing, Lexington, Massachusetts, 1975.
- [7] J. P. S. Kung, *The cycle structure of a linear transformation over a finite field*, Lin. Alg. Apps. **36** (1981), 141–155.
- [8] R. Lidl and H. Niederreiter, *Finite fields*, Addison-Wesley, London, 1983.
- [9] R. Pöschel M. Klin, V. Liskovets, *Analytical enumeration of circulant graphs with prime squared number of vertices*, Sémin. Loth. de Comb. **36** (1996), Art. B36d, (on-line journal).
- [10] M. Mishna, *Cayley Graph Enumeration*, Master's Thesis, Simon Fraser University, 2000.
- [11] J. Morris, *Isomorphisms of Cayley Graphs*, Ph.D. thesis, Simon Fraser University, 1999.
- [12] L. Nowitz, *A non-Cayley-invariant Cayley Graph of the elementary abelian group of order 64*, Discrete Math **110** (1992), 223–228.
- [13] R.C. Read, *Combinatorial enumeration for groups, graphs and chemical compounds*, Springer-Verlag, New York, 1987.
- [14] K. I. M. Rosen, *A classical introduction to modern number theory*, Springer-Verlag, New York, 1983.

- [15] S. Toida, *A note on Ádám's conjecture*, J. Combin. Theory Ser. B **23** (1977), 239–246.
- [16] J. Turner, *Point-symmetric graphs with a prime number of points*, J. Combin. Theory **3** (1967), 136–145.
- [17] M.-Y. Xu, *On isomorphism of Cayley digraphs and graphs of groups of order  $p^3$* , Adv. in Math (China) **17** (1988), 427–428.

# Tirage à pile ou face de mots de Fibonacci

Jean Guy Penaud, Olivier Roques\*

*LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France*

## Résumé

Dans cet article, on donne un semi-algorithme qui permet d'engendrer de façon aléatoire et rigoureusement uniforme un mot du langage de Fibonacci,  $L = \epsilon + La + Lbb$ . Il utilise un générateur uniforme de bits et la complexité moyenne en temps et en espace est linéaire. On montre comment transformer ce semi-algorithme en algorithme à terminaison assurée, tout en gardant une complexité linéaire en moyenne. Enfin, on donne une famille de langages rationnels pour lesquels notre méthode s'applique.

## Abstract

In this paper, we give a semi-algorithm to sample uniformly at random a word of the Fibonacci language  $L = \epsilon + La + Lbb$ . It needs an uniform bit generator and the average complexity is linear. We show that we can transform this semi-algorithm into an algorithm which always ends, with a linear complexity again. At the end, we discuss a generalization for a class of rational languages.

## Introduction

La génération aléatoire de mots d'un langage rationnel est un problème que l'on sait résoudre par une méthode récursive, en considérant ces langages comme cas particulier de langages algébriques [6] ou de structures décomposables [5]. Cependant, la mise en œuvre de cette méthode conduit à manipuler des nombres exponentiels en la longueur du mot que l'on souhaite engendrer, et ne permet pas d'engendrer des mots de grande taille. Récemment, Denise a donné dans [3] une amélioration de l'algorithme de [6] pour les langages rationnels dont la série génératrice selon la longueur possède un seul pôle dominant. Néanmoins, dans certains cas le calcul du nombre de mots de longueur  $n$  est nécessaire. Dans cet article, nous proposons une méthode de complexité linéaire en moyenne, permettant d'engendrer en quelques secondes un mot de plusieurs millions de lettres du langage de Fibonacci. Pour cela, nous ne faisons aucune opération arithmétique sur des grands nombres et n'utilisons pas d'arithmétique flottante. Notre méthode est fondée sur l'utilisation d'un semi-algorithme à rejet [1], [9], [8], qui construit le mot motif par motif avec les probabilités appropriées et a la possibilité de rejeter le mot et de recommencer lorsque celui-ci n'est pas correct. Pour décider quel motif choisir entre "a" et "bb" pour le concaténer au mot courant, on effectue un tirage paresseux : il consiste à tirer juste le nombre de bits nécessaires du développement binaire d'un réel aléatoire pour pouvoir décider entre les motifs générateurs du langage. Les mots engendrés le sont de façon rigoureusement uniforme sous l'hypothèse que l'on sache également tirer des bits de façon uniforme [7]. Dans la première section, nous décrivons la méthode paresseuse du tirage à pile ou face. Dans la deuxième nous appliquons cette méthode

---

\* Email: penaud@labri.u-bordeaux.fr, roques@labri.u-bordeaux.fr.

à la construction d'un mot de Fibonacci. La troisième section présente la complexité moyenne en nombre de bits aléatoires pour engendrer un mot de longueur  $n$ . Nous montrons également comment on peut transformer notre semi-algorithme en algorithme, tout en gardant une complexité linéaire. Dans la dernière section, nous discutons les extensions possibles de notre algorithme à une famille de langages rationnels, les *étoiles de motifs*.

## 1 Tirage paresseux

L'objectif est de tirer dans un espace d'épreuve  $E$  à deux éléments mutuellement exclusifs  $A$  et  $B$ , l'événement  $A$  avec la probabilité donnée  $p$ . La façon usuelle de procéder est de tirer selon la loi continue et uniforme un nombre réel  $r$  entre 0 et 1, et de choisir  $A$  si  $0 \leq r \leq p$ , et  $B$  si  $p < r \leq 1$ . Nous présentons ici une méthode différente qui opère dans le cadre défini ci-dessous.

### Conditions d'application

Les hypothèses supplémentaires que l'on s'impose sont :

- disposition d'un générateur uniforme de bits selon Knuth [7] ;
- arithmétique binaire entière à l'exclusion des réels ;
- résultat : génération uniforme sans biais ;
- probabilité cible  $p$  donnée comme racine simple d'un polynôme  $P$  de degré  $d$  à coefficients entiers, supposé pour simplifier monotone dans l'intervalle  $[0, 1]$  (mas on n'impose pas de connaître *a priori* le développement infini de  $p$ ).

### Méthode

On tire le nombre  $r$  bit par bit. Le tirage est paresseux car on s'arrête de tirer dès que l'on peut déterminer si, quelle que soit la suite des bits tirés ultérieurement, la valeur globale du nombre  $r$  construit sera définitivement inférieure ou supérieure à la probabilité cible  $p$ , rendant donc superflu de continuer. Un tel bit  $r_i$  sera dit *bit décisif*, et on montrera que le nombre moyen de bits *utiles*, c'est-à-dire le nombre de bits tirés jusqu'au bit décisif est exactement 2.

La comparaison du nombre tiré  $r$  avec la probabilité cible  $p$  se fera en évaluant le signe du polynôme définissant la probabilité, pour la valeur de la séquence tirée. Ce calcul se fait en binaire avec une complexité (évaluée en multiplications bit à bit) finie.

### Bit décisif

Etant donné un nombre réel  $r$  compris entre 0 et 1, on note  $r_i$  le  $i$ ème bit de l'écriture binaire de sa partie fractionnaire. Inversement, étant donné un mot  $w$  (possiblement infini) sur l'alphabet  $\{0, 1\}$ , on note  $w_i$  la  $i$ ème lettre de  $w$  et  $val(w)$  le nombre qui admet  $w$  comme développement fractionnaire, soit  $val(w) = \sum_{i \geq 1} w_i \frac{1}{2^i}$ .

Avec ces notations nous pouvons préciser la notion de bit décisif.

**Définition 1** Soient deux nombres  $p$  et  $r$  dans  $[0, 1]$ . On dira que le bit  $r_i$  du développement de  $r$  est décisif si  $i$  est le rang du premier bit qui diffère dans les développements de  $p$  et de  $r$ .

Plus précisement,

1.  $1 \leq j < i \Rightarrow r_j = p_j$ ,
2.  $(r_i = 1 \wedge p_i = 0) \cup (r_i = 0 \wedge p_i = 1)$ .

Ainsi la connaissance des  $i$  premiers bits permet de décider, quelle que soit la continuation du développement de  $r$ , si  $r > p$  ou si  $r \leq p$ .

**Propriété 2** Si  $p$  est donné, et si  $r$  est tiré uniformément bit par bit, le rang moyen du bit décisif  $r_i$  est égal à  $\sum_{i \geq 1} \frac{i^d}{2^i} = 2$ .

Notons que cette somme réapparaîtra de façon généralisée,  $\sum_{i \geq 1} \frac{i^d}{2^i}$ , ci-dessous dans l'étude de la complexité binaire.

Plaçons nous maintenant dans le cadre spécifique de l'article, c'est-à-dire le développement binaire de la probabilité cible  $p$  est inconnu, mais  $p$  est déterminé uniquement comme racine d'un polynôme  $P$  donné, racine supposée unique dans l'intervalle  $[0,1]$ .

La propriété suivante montre qu'il est toujours possible de déterminer efficacement le bit décisif de  $r$ .

## Complexité bit à bit

Une conséquence directe de la définition du bit décisif conduit à la propriété suivante.

**Propriété 3** Sous les mêmes hypothèses que la Propriété 2, on a,

1.  $r_i = 1 \Rightarrow \{p_i = 0 \Leftrightarrow \{\text{val}(r_1 r_2 \dots r_i) > p\}$ ,
2.  $r_i = 0 \Rightarrow \{p_i = 1 \Leftrightarrow \{\text{val}(r_1 r_2 \dots r_{i-1} 1) \leq p\}$ .

**Preuve :**

Pour le point 1 c'est évident car précisément  $\{r_i = 1 \wedge p_i = 0\} \Leftrightarrow \{\text{val}(r_1 r_2 \dots r_i) > p\}$ . De même pour le point 2 c'est vrai *a fortiori* car  $\{\text{val}(r_1 r_2 \dots r_{i-1} 1) \leq p\} \Leftrightarrow p_i = 1$ .

Après chaque tirage d'un bit  $r_i$ , il convient de vérifier s'il est décisif. Pour cela il suffit, d'après la propriété 2, et sous les hypothèses choisies pour le polynôme  $P$  dont  $p$  est racine (à savoir  $p$  racine simple unique dans l'intervalle  $[0,1]$ ), de calculer le signe de  $P(\text{val}(r_1 \dots r_{i-1} 1))$ .

**Propriété 4** Si  $d$  est le degré du polynôme, et  $l$  le nombre maximum de bits nécessaires pour coder les coefficients, le nombre d'exécutions d'opérations binaires est majoré par la constante  $2^d l \sum_{i > 0} \frac{i^d}{2^i}$

Cette constante peut également s'écrire  $2^{d+1} l i^d A(d)$ , où  $A(d)$  est le nombre d'arrangements préférentiels introduit par Cayley [2], qui apparaît dans le développement de la fonction génératrice exponentielle  $\frac{1}{2-e^x}$ .

**Preuve :**

Selon le schéma de Horner, le polynôme  $P$  peut s'écrire,  
 $P(x) = ((a_d x + a_{d-1}) x + a_{d-2}) x + \dots + a_0$ .

Or une multiplication de nombres binaires a pour complexité bit à bit le produit des longueurs, et une somme de tels nombres la plus grande des longueurs des deux termes. D'où la première expression de la complexité.

Pour obtenir la seconde il suffit de remarquer que,

$$\sum_{n \geq 0} \left( \sum_{i \geq 0} \frac{i^n}{2^i} \right) \frac{x^n}{n!} = \sum_{i \geq 0} \left( \frac{1}{2^i} \sum_{n \geq 0} \frac{(ix)^n}{n!} \right) = \sum_{i \geq 0} \frac{1}{2^i} (e^{ix}) = \sum_{i \geq 0} \left( \frac{e^x}{2} \right)^i = \frac{1}{2 - ex}.$$

La suite des nombres d'arrangements préférentiels est la suite 1, 1, 3, 13, 75, 541, 4683, ..., que l'on trouve dans [10] avec de nombreuses références.

## Généralisation

Il est clair que le procédé s'étend au cas où la probabilité cible  $p$  est racine multiple (on prend la dérivée adéquate), ou également au cas où l'intervalle d'unicité de la racine est strictement inclus dans  $[0,1]$ . De même, la technique du résultant de deux polynômes permet de traiter le cas où l'on a plus de deux événements mutuellement exclusifs à tirer. Il suffit d'exprimer la seconde (voire la troisième ...) probabilité cible sous forme polynomiale en fonction de la première, ce qui constitue le second polynôme du résultant, le premier restant le polynôme  $P$  dont  $p$  est racine.

## 2 Génération aléatoire du langage de Fibonacci

Appelons langage de Fibonacci, l'ensemble des mots sur l'alphabet  $\{a, b\}$  défini par la grammaire,

$$\mathbf{L} = \epsilon + \mathbf{L}a + \mathbf{L}bb. \quad (1)$$

Le nombre de mots de  $\mathbf{L}$  à  $n$  lettres, noté  $L_n$ , est le nombre de Fibonacci qui s'exprime en fonction du *nombre d'or*  $\phi = \frac{1+\sqrt{5}}{2}$ , par  $L_n = \frac{\sqrt{5}}{5}(\phi^{n+1} - \phi^{-n-1})$ .  $L_n$  est solution de la récurrence linéaire  $L_0 = L_1 = 1$ ,  $L_n = L_{n-1} + L_{n-2}$  pour  $n \geq 2$ .

Pour tirer un mot de Fibonacci de façon équiprobable parmi l'ensemble des mots de longueur  $n$ , on construit le mot, motif par motif, en choisissant de concaténer au mot courant le motif  $a$  avec la probabilité  $p$  et le motif  $bb$  avec la probabilité  $q$ . On doit avoir d'une part  $q = p^2$  pour respecter l'uniformité selon la longueur, et  $p + q = 1$  car ce sont les deux seuls motifs possibles. D'où l'équation polynomiale  $1 - p - p^2 = 0$  dont  $p = \phi^{-1}$  est racine dans l'intervalle  $[0,1]$ .

Par ce procédé, chaque mot de  $L$  de longueur  $n$  est engendré avec la probabilité  $\phi^{-n}$ . Si la longueur du mot obtenu est  $n+1$ , on le rejette et on recommence le procédé. Ceci est illustré par l'Algorithme 1.

Dans la suite, on appellera *essai*, le tirage sans rejet d'un mot de longueur  $n$  ou  $n+1$  (ce qui revient à supprimer les lignes 5, 6 et 7 de l'Algorithme 1).

A chaque étape de la construction du mot on effectue un tirage paresseux afin de choisir entre les deux motifs, et ceci sans connaître le développement binaire de  $\phi^{-1}$ , comme expliqué dans la section 1.

L'Algorithme 2 met en oeuvre ce procédé. Pour comparer la valeur de la suite binaire courante avec  $\phi^{-1}$ , il suffit d'évaluer le signe du polynôme  $P(t) = 1 - t - t^2$ . Ce polynôme, décroissant sur  $[0, 1]$ , vérifie bien les hypothèses de validité de la méthode.

---

**Algorithme 1** Tirage uniforme d'un mot de Fibonacci (semi-algorithme)

---

**Entrée** un entier  $n$ **Sortie** un mot de Fibonacci aléatoire et uniforme de longueur  $n$ 

```

1:  $w \leftarrow \epsilon$ 
2: tant que  $\|w\| < n$  faire
3:    $l \leftarrow \text{Motif\_aléatoire}()$ 
4:    $w \leftarrow w l$ 
5:   si  $\|w\| = n + 1$  alors
6:      $w \leftarrow \epsilon$ 
7:   fin si
8: fin tant que

```

---



---

**Algorithme 2** Motif\_aléatoire()

---

**Entrée**  $P(t) = 1 - t - t^2$ **Sortie** le motif  $a$  avec probabilité  $\phi^{-1}$  ou le motif  $bb$  avec probabilité  $1 - \phi^{-1}$ 

```

1:  $r \leftarrow \epsilon$ 
2: tant que on n'a pas décidé entre  $a$  et  $bb$  faire
3:    $z$  un bit aléatoire uniforme.
4:    $r \leftarrow rz$ 
5:   si  $z = 1$  alors
6:     si  $P(\text{val}(r)) < 0$  alors
7:       retourner  $bb$ 
8:     fin si
9:   sinon  $\{z = 0\}$ 
10:    si  $P(\text{val}(r_1 \dots r_{k-1} 1)) > 0$  alors
11:      retourner  $a$ 
12:    fin si
13:  fin si
14: fin tant que

```

---

### 3 Complexité

On évalue la complexité de l'Algorithme 1, en nombre d'appels à la fonction Motif\_aléatoire() (Algorithme 2) nécessaires à l'obtention d'un mot de Fibonacci de longueur  $n$  (i.e. à la terminaison de l'Algorithme 1).

De l'équation (1) du langage de Fibonacci, on déduit la série génératrice de  $\mathbf{L}$  selon la longueur

$$\mathbf{L}(x) = \sum_{n \geq 0} L_n x^n = \frac{1}{1 - x - x^2} = \frac{1}{(1 - \phi x)(1 - \phi^{-1}x)}.$$

où  $\phi^{-1}$  est le pôle de plus petit module de  $\mathbf{L}(x)$ . On en déduit immédiatement le comportement asymptotique du nombre de mots de Fibonacci de longueur  $n$ ,  $L_n \sim \frac{\sqrt{5}}{5} \phi^{n+1}$ .

La complexité moyenne de l'Algorithme 1 s'exprime comme le produit du nombre moyen d'appels à la fonction Motif\_aléatoire() par le nombre moyen d'essais nécessaires à l'obtention d'un mot de Fibonacci de longueur  $n$ .

Pour calculer cette complexité, nous définissons une variable aléatoire  $X$  qui compte le nombre de motifs  $a$  et de motifs  $bb$  lors d'un essai de l'Algorithme 1. La moyenne de  $X$ , notée  $E(X)$  s'exprime comme la somme du nombre moyen de motifs  $a$  et de motifs  $bb$  dans les mots de  $\mathbf{L}$  de longueur  $n$  et du nombre moyen de motifs  $a$  et de motifs  $bb$  dans les mots de  $\mathbf{L}bb$  de longueur  $n + 1$ .

Notons  $\mathbf{L}_n$ , l'ensemble des mots de Fibonacci de longueur  $n$ . Soit  $A_n = \sum_{f \in \mathbf{L}_n} |f|_a + |f|_{bb}$ , le nombre de motifs  $a$  et de motifs  $bb$  dans les mots de longueur  $n$ . D'après (1)  $\mathbf{L}_n = \mathbf{L}_{n-1}a + \mathbf{L}_{n-2}bb$ . On peut donc écrire

$$\begin{aligned} A_n &= \sum_{f \in \mathbf{L}_{n-1}a} |f|_a + |f|_{bb} + \sum_{f \in \mathbf{L}_{n-2}bb} |f|_a + |f|_{bb} \\ &= \sum_{f \in \mathbf{L}_{n-1}} (|f|_a + 1) + |f|_{bb} + \sum_{f \in \mathbf{L}_{n-2}} |f|_a + (|f|_{bb} + 1) \\ &= A_{n-1} + A_{n-2} + L_{n-1} + L_{n-2}. \end{aligned}$$

Puisque  $L_n = L_{n-1} + L_{n-2}$ , on obtient

$$A_0 = 0, A_1 = 1, A_2 = 3, A_3 = 7, \text{ et pour } n \geq 4$$

$$A_n = 2A_{n-1} + A_{n-2} - 2A_{n-3} - A_{n-4}.$$

On en déduit alors (voir [4])

$$A_n = [x^n] \frac{x + x^2}{(1 - x - x^2)^2} \sim \frac{\phi^2}{(1 + 2\phi^{-1})^2} \phi^n n,$$

ce qui donne

$$\begin{aligned} E(X) &= A_n \phi^{-n} + (A_{n-1} + L_{n-1}) \phi^{-n-1}, \\ &\sim \frac{\phi^2 + 1}{5}. \end{aligned} \tag{2}$$

On calcule maintenant le nombre moyen d'essais nécessaires à l'obtention d'un mot de Fibonacci. Soit  $Y$  la variable aléatoire qui compte le nombre moyen d'essais pour

obtenir un mot correct.  $Y$  suit une loi géométrique avec comme probabilité de succès  $L_n \phi^{-n}$ . On a immédiatement,

$$E(Y) \sim \sqrt{5} \phi^{-1}. \quad (3)$$

En multipliant (2) par (3) on obtient le nombre moyen de tirages de motifs  $a$  ou de motifs  $bb$ , ce qui fait l'objet de la proposition suivante.

**Proposition 5** *Notons  $\mathcal{C}_n$ , la complexité moyenne de l'Algorithme 1 en nombre d'appels à la fonction Motif\\_aléatoire().*

$$\mathcal{C}_n \sim n.$$

D'après la Propriété 2, il faut en moyenne 2 bits pour tirer un motif aléatoire. Ainsi, la complexité moyenne de l'Algorithme 1 en nombre de bits aléatoires est asymptotiquement égale à  $2n$ . Si l'on compte les opérations binaires, cette complexité est majorée par  $2kn$  où  $k$  est la constante calculée dans la section 1.

## 4 Algorithme à terminaison assurée

La complexité dans le pire des cas de l'Algorithme 1 est infinie. Ceci est une caractéristique des algorithmes à rejet, et peut ne pas convenir à certaines applications qui requièrent une réponse sûre. Néanmoins, puisque l'on connaît une algorithme dont le pire des cas est  $O(n^2)$  [3], on peut décider de lancer l'algorithme quadratique si le semi-algorithme (Algorithme 1) n'est pas terminé au bout de  $T$  essais (Algorithme 3).

**Algorithme 3** Tirage uniforme d'un mot de Fibonacci (algorithme)

**Entrée** un entier  $n$ , un entier  $T$

**Sortie** un mot de Fibonacci aléatoire et uniforme de longueur  $n$

- 1:  $w \leftarrow \epsilon$
- 2:  $i \leftarrow 0$
- 3: **tant que**  $i < T$  et  $\|w\| < n$  **faire**
- 4:    $l \leftarrow \text{Motif\_aléatoire}()$
- 5:    $w \leftarrow w l$
- 6:   **si**  $\|w\| = n + 1$  **alors**
- 7:      $w \leftarrow \epsilon$
- 8:   **fin si**
- 9: **fin tant que**
- 10: **si**  $\|w\| = n + 1$  **alors**
- 11:   Lancer un autre algorithme (qui termine).
- 12: **fin si**

Soit  $K(n) = O(n^2)$  la complexité moyenne d'un algorithme fondé sur la méthode récursive (par exemple l'algorithme proposé dans [3]) pour engendrer un mot de Fibonacci

de longueur  $n$ . La complexité moyenne de l'Algorithme 3 est alors

$$\begin{aligned} & E(X) \sum_{i=1}^T i \operatorname{Prob}(Y = i) + \operatorname{Prob}(Y > T)(K(n) + TE(X)) \\ &= E(X) \frac{1 - (T+1)q^T + Tq^{T+1}}{1-q} + q^T(K(n) + TE(X)). \end{aligned}$$

où  $q = \frac{L_{n-1}}{L_n + 1} \sim \phi^{-2}$ . En prenant par exemple  $T = n$ , la complexité de l'Algorithme 3 est  $O(n)$ .

## 5 Extensions

L'Algorithme 1 peut aisément s'étendre aux langages rationnels du type *étoile de motifs* (Fig. 1),

$$\mathbf{L} = (m_1 + m_2 + \cdots + m_k)^*,$$

où  $m_1, \dots, m_k$  sont des motifs de longueur quelconque. Dans ce cas, chaque motif de longueur  $l$  doit être tiré avec une probabilité  $p^l$ , où  $p$  est racine réelle positive du polynôme

$$1 - \sum_{i=1}^k x^{\|m_i\|}.$$

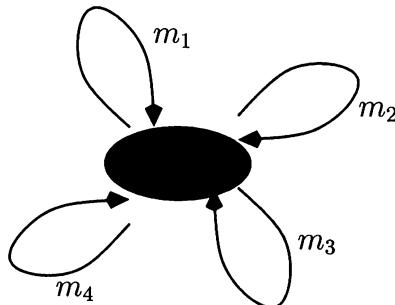


FIG. 1 – Une étoile à quatre motifs.

Cette racine est l'unique racine positive d'après la loi des signes de Descartes.

Le nombre de rejets reste en  $O(1)$  et le nombre moyen de motifs à choisir pour construire un mot de longueur  $n$  diminue bien évidemment lorsque la longueur des motifs est grande. Pour choisir un motif avec une probabilité donnée, on tire d'abord la longueur  $l$  du motif par la méthode étendue du tirage paresseux (cf section 1). Puis on choisit parmi les motifs de même longueur  $l$  à nouveau par un tirage bit à bit (cf [7]).

## Références

- [1] Alonso (L.) et Schott (R.). – *Random Generation of Trees*. – Kluwer Academic Publishers, 1995.

- [2] Cayley (A.). – On the theory of the analytical forms called trees ii. *Math. Papers*, vol. 4, 1859, pp. 112–115.
- [3] Denise (A.). – Génération aléatoire et uniforme de mots de langages rationnels. *Theoretical Computer Science*, vol. 159, 1996, pp. 43–63.
- [4] Flajolet (Ph.). – *Introduction à l'analyse des algorithmes*. – International Thomson Publishing, 1996.
- [5] Flajolet (Ph.), Zimmerman (P.) et Cutsem (B. Van). – A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science, all*, vol. 132, 1994, pp. 1–35.
- [6] Hickey (T.) et Cohen (J.). – Uniform random generation of strings in a context-free language. *Siam.J. Comput*, vol. 12, 1983, pp. 645–655.
- [7] Knuth (D.E.) et Yao (A.C.). – The complexity of nonuniform random number generation. *[CA] Algorithms and Complexity, Proc.Symp., Pittsburgh*, 1976, pp. 357–428.
- [8] Penaud (J.G.), Pergola (E.), Pinzani (R.) et Roques (O.). – Chemins de Schröder et hiérarchies aléatoires. *Theoretical Computer Science*, 2000. – à paraître.
- [9] Penaud (J.G.) et Roques (O.). – Génération de chemins de Dyck à pics croissants. In : *11<sup>ème</sup> Colloque Séries Formelles et Combinatoire Algébrique*, pp. 438–449. – Soumis à Discrete Math.
- [10] Sloane (N.J.A) et Plouffe (S.). – *The Encyclopedia of integer sequences*. – Academic Press, 1995.

# The Yuri Manin ring and its $\mathcal{B}_n$ -analogue

MARGARET A. READDY \*

## Abstract

The Manin ring is a family of quadratic algebras describing pointed stable curves of genus zero whose homology gives the solution of the Commutativity Equations. This solution was first observed by the physicist Losev. We show the Manin ring is the Stanley-Reisner ring of the standard triangulation of the  $n$ -cube modulo a system of parameters. Thus, the Hilbert series of the Manin ring is given by the Eulerian polynomials. One can also view the Manin ring as the Stanley-Reisner ring of the dual of the permutohedron modulo a system of parameters. Furthermore, we develop a  $\mathcal{B}_n$ -analogue of the Manin ring. In this case the signed Manin ring is the Stanley-Reisner ring of the barycentric subdivision of the  $n$ -cube (equivalently, the dual of the signed permutohedron) modulo a system of parameters and its Hilbert series is the descent polynomial of augmented signed permutations.

## Résumé

L'anneau de Manin est une famille d'algèbres quadratiques décrivant des courbes pointées et stables de genre zéro dont l'homologie donne la solution des Équations de Commutativité. Cette solution a été observée pour la première fois par le physicien Losev. Nous montrons que l'anneau de Manin est le quotient de l'anneau de Stanley-Reisner de la triangulation standard de l'hypercube par un système de paramètres. Par conséquent la série d'Hilbert de l'anneau de Manin est donnée par les polynômes Eulériens. On peut aussi regarder l'anneau de Manin comme le quotient de l'anneau de Stanley-Reisner du dual du permutoèdre par un système de paramètres. De plus, nous développons un  $\mathcal{B}_n$  analogue de l'anneau de Manin. Dans ce cas-là l'anneau de Manin signé est le quotient de l'anneau de Stanley-Reisner de la subdivision barycentrique de l'hypercube (ou, d'une manière équivalente, du dual du permutoèdre signé) par un système de paramètres, et sa série d'Hilbert est le polynôme de descentes des permutations augmentées et signées.

## 1 Introduction

Losev and Manin [15] showed the solutions to the Commutativity Equations (the pencils of formal flat connections) correspond to the homology of a family of pointed stable curves of genus zero, denoted  $\bar{L}_n$ . Their work parallels that of the solutions to the Associativity Equations, that is, Frobenius manifolds, first discovered by string theorists and later verified mathematically [10, 11, 12]. The Losev-Manin solution of the Commutativity Equations required verifying the dimension of  $\bar{L}_n$  is  $n!$ .

---

\*This work was done while the author was a Visiting Professor at Stockholm University during the academic year 1999–2000.

In this paper we determine the Hilbert series of this family of commutative quadratic algebras which we call the Manin ring and hence conclude the dimension result. We also give a combinatorial description of this algebra as well as a  $\mathcal{B}_n$ -analogue. The techniques we use are familiar to combinatorialists, but do not seem as well-known in the commutative algebra community. It is our hope that this paper will remedy this deficiency and at the same time give insight to develop and study new interesting quadratic algebras.

Recall for  $\Delta$  a simplicial complex with vertex set  $V(\Delta)$ , the Stanley-Reisner ring  $\mathbf{k}[\Delta]$  is the polynomial ring in the variables  $x_v$  for  $v \in V(\Delta)$  modulo the ideal of non-faces of  $\Delta$ . Stanley used this ring to prove the Upper Bound Theorem for simplicial spheres [18]. In the case that a simplicial complex is shellable, and hence, having Stanley-Reisner ring which is Cohen-Macaulay, Kind and Kleinschmidt [9] gave necessary and sufficient conditions for  $(\theta_0, \dots, \theta_d)$  to be a linear system of parameters (s.o.p.). Additionally, they exhibit an explicit basis for the Stanley-Reisner ring of  $\Delta$  modulo an s.o.p. which comes from the shelling.

Using the Kind-Kleinschmidt characterization, we show the Manin ring  $\overline{L}_n$  is the familiar Stanley-Reisner ring of the standard triangulation of the  $n$ -cube modulo a system of parameters. As this simplicial complex is shellable, it is then a straightforward matter to compute the Hilbert series of the associated ring. We do this in Section 3.

In Section 4 we give a  $\mathcal{B}_n$ -analogue of the Manin ring, denoted  $\overline{L}_n^\pm$ . Geometrically this signed Manin ring is the Stanley-Reisner ring of the barycentric subdivision of the  $n$ -cube modulo an s.o.p. As in the case of the Manin ring, we again compute the associated Hilbert series. It would be interesting to see if this signed analogue has any physical meaning as was known for the Commutativity Equation solutions [13, 14]. In the last section of the paper we suggest some possible directions of generalization.

## 2 Definitions

We now review some elementary concepts in commutative algebra which can be found in [21]. Throughout  $[n]$  will denote the set  $[n] = \{1, \dots, n\}$ . Let  $\mathbf{k}$  be a field of characteristic zero and let  $A = \bigoplus_{i \geq 0} A_i$  be a *standard graded  $\mathbf{k}$ -algebra*, that is,  $A$  is generated by a finite number of degree 1 homogeneous elements. The *Hilbert series* is defined to be

$$\mathcal{H}(A) = \sum_{i \geq 0} \dim_{\mathbf{k}}(A_i) \cdot t^i.$$

A basic result of Noether's is the existence of a linear system of parameters. Namely, for  $A$  a standard graded  $\mathbf{k}$ -algebra, there exists a finite number of homogeneous degree 1 elements  $\theta_0, \dots, \theta_d$  which are algebraically independent over  $\mathbf{k}$ . Furthermore, there exists a finite number of homogeneous elements  $\eta_1, \dots, \eta_s$  such that for all  $x \in A$

$$x = \sum_{i=1}^s \eta_i \cdot p_i(\theta_0, \dots, \theta_d), \quad (2.1)$$

where  $p_i$  is a polynomial in the  $\theta_0, \dots, \theta_d$  depending on  $x$ . The elements  $\theta_0, \dots, \theta_d$  are called a (*linear*) *system of parameters*, abbreviated s.o.p. An s.o.p. is *regular* if for some choice of homogeneous elements  $\eta_1, \dots, \eta_s$  the representation in (2.1) is unique, while a standard  $\mathbf{k}$ -algebra is *Cohen-Macaulay* if some (equivalently, every) s.o.p. is regular.

The *Stanley-Reisner ring* or *face ring* of a finite simplicial complex  $\Delta$  on the vertex

Facet $F_i$	$\pi(F_i)$	$R(F_i)$
$F_1 = \{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}\}$	123	$\emptyset$
$F_2 = \{\emptyset, \{1\}, \{1, 3\}, \{1, 2, 3\}\}$	132	$\{\{1, 3\}\}$
$F_3 = \{\emptyset, \{2\}, \{1, 2\}, \{1, 2, 3\}\}$	213	$\{\{2\}\}$
$F_4 = \{\emptyset, \{2\}, \{2, 3\}, \{1, 2, 3\}\}$	231	$\{\{2, 3\}\}$
$F_5 = \{\emptyset, \{3\}, \{1, 3\}, \{1, 2, 3\}\}$	312	$\{\{3\}\}$
$F_6 = \{\emptyset, \{3\}, \{2, 3\}, \{1, 2, 3\}\}$	321	$\{\{3\}, \{2, 3\}\}$

Table 1: Shelling restrictions for the 3-cube using the lexicographic shelling order

set  $V(\Delta)$  is the quotient ring

$$\mathbf{k}[\Delta] = \mathbf{k}[x_v : v \in V(\Delta)]/I(\Delta),$$

where  $I(\Delta)$  is the face ideal generated by all squarefree monomials  $x_{v_1} \cdots x_{v_j}$  satisfying the condition that the vertices  $v_1, \dots, v_j$  do not lie on a common face of  $\Delta$ .

Consider the  $n$ -cube  $C_n$  as  $[0, 1]^n$ . The *standard triangulation of the  $n$ -cube*, denoted  $\Delta(C_n)$ , is the triangulation induced by cutting  $[0, 1]^n$  with the  $\binom{n}{2}$  hyperplanes  $x_i = x_j$ ,  $1 \leq i < j \leq n$ . The resulting triangulation gives  $n!$  simplices of dimension  $n$ .

The *Stanley-Reisner ring of the triangulated  $n$ -cube* is given by  $\mathbf{k}[\Delta(C_n)] = \mathbf{k}[y_t : t \subseteq [n]]/I(\Delta(C_n))$ , where the ideal  $I(\Delta(C_n))$  is generated by the degree 2 monomials  $y_t \cdot y_u$  with the vertices  $t$  and  $u$  not lying on the same facet of  $\Delta(C_n)$ . Alternatively, one can think of the triangulated  $n$ -cube as being the Boolean algebra  $B_n$ , that is, the lattice of all subsets of  $[n]$  ordered with respect to inclusion. The  $n$ -dimensional simplices of  $\Delta(C_n)$  correspond to maximal chains in the Boolean algebra. More formally, the maximal chain  $c : \emptyset = P_0 \subset P_1 \subset \cdots \subset P_{n-1} \subset P_n = [n]$  in the Boolean algebra  $B_n$  corresponds to the facet  $F$  spanned by the vertices  $V(F) = \{P_0, \dots, P_n\}$ . Thus the Stanley-Reisner ring is simply the polynomial ring having variables the  $2^n$  elements of the Boolean algebra and the face ideal is generated by pairs of non-comparable elements in the poset.

Let  $F_1, \dots, F_s$  be a linear ordering of the facets of a pure simplicial complex  $\Delta$  of dimension  $d$  and set  $\Delta_j = \overline{F}_1 \cup \cdots \cup \overline{F}_j$ . Such a linear ordering of the facets is called a *shelling order*, or *shelling*, if  $\Delta_{k-1} \cap F_k$  is pure and  $(d-1)$ -dimensional for  $2 \leq k \leq s$ . If such a facet order exists, we say  $\Delta$  is *shellable*. Given a shelling order, for a facet  $F_k$  the *restriction*  $R(F_k)$  is the minimal new face added in the  $k$ th shelling step, that is,  $R(F_k) = \min\{F \in \Delta_k : F \notin \Delta_{k-1}\}$ .

Given a facet  $F \in \Delta(C_n)$  with corresponding maximal chain  $c : \emptyset = P_0 \subset \cdots \subset P_n = [n]$  in the Boolean algebra  $B_n$ , let  $\pi_i$  denote the unique element in the set difference  $P_i - P_{i-1}$  for  $1 \leq i \leq n$ . Then  $\pi(F) = \pi_1 \cdots \pi_n \in S_n$  is the *permutation representation of the facet  $F$* . For  $\pi = \pi_1 \cdots \pi_n$  a permutation in the symmetric group  $S_n$ , the *number of descents* of  $\pi$ , denoted  $\text{des}(\pi)$ , is the number of indices  $i$  such that  $\pi_i > \pi_{i+1}$ . The *descent set* of  $\pi$ , denoted  $\text{Des}(\pi)$ , is the set of indices  $i$  such that  $\pi_i > \pi_{i+1}$ .

A shelling order for the triangulated  $n$ -cube is to take the facet order given by the lexicographic order inherited by the permutation representation of each facet. For the 3-cube this shelling order is given in Table 1. Observe there is one restriction of size 0, four of size 1 and one of size 2. In general, a result due to Björner on *EL*-labelings [1] implies the facet restriction  $R(F_i)$  depends on the descent set of the permutation  $\pi(F)$ . More specifically, if  $\pi(F_i) = \pi_1 \cdots \pi_n$  and  $\text{Des}(\pi(F_i)) = \{s_1, \dots, s_k\}$  with  $s_1 < \cdots < s_k$  then  $R(F_i) = \{T_1, \dots, T_k\}$  where  $T_i = \{\pi_1 \cup \cdots \cup \pi_{s_i}\}$ .

One well-known example of shellable objects is due to Bruggesser and Mani, namely the boundary complex of convex polytopes [2]. Implicit from Hochster's work [8] is the result that if a simplicial complex is shellable then its Stanley-Reisner ring is Cohen-Macaulay.

Let  $\Delta$  be a pure shellable simplicial complex of dimension  $d$  and  $\theta_0, \dots, \theta_d$  be homogeneous degree 1 elements from  $\mathbf{k}[\Delta]$  of the form  $\theta_i = \sum_{v \in V(\Delta)} m_{i,v} \cdot x_v$  with  $m_{i,v} \in \mathbf{k}$ . Let  $M$  denote the matrix of the linear forms  $\theta_i$ , that is,  $M = (m_{i,v})_{0 \leq i \leq d, v \in V(\Delta)}$ . For  $F$  a facet of  $\Delta$ , let  $M|_F$  denote the matrix formed by restricting the  $\theta_i$  to the variables  $x_v$  for  $v \in F$ . Observe that  $M|_F$  is a square matrix of order  $d+1$ .

A beautiful result of Kind and Kleinschmidt [9] characterizes systems of parameters for shellable simplicial complexes.

**Theorem 2.1 (Kind-Kleinschmidt)** *Let  $\Delta$  be a pure shellable simplicial complex of dimension  $d$  with shelling order  $F_1, \dots, F_s$ . Let  $\theta_0, \dots, \theta_d$  be homogeneous degree 1 elements in  $\mathbf{k}[\Delta]$  and let  $M$  be the matrix of the  $\theta_i$ 's. Then*

- (a) *The elements  $\theta_0, \dots, \theta_d$  are an s.o.p. for  $\mathbf{k}[\Delta]$  if and only if the matrix  $M|_F$  is invertible for any facet  $F \in \Delta$ .*
- (b) *A basis for  $\mathbf{k}[\Delta]/(\theta_0, \dots, \theta_d)$  is given by the facet restriction monomials, that is,*

$$\eta_i = \prod_{v \in R(F_i)} x_v, \quad 1 \leq i \leq s.$$

In the case that a standard graded  $\mathbf{k}$ -algebra is Cohen-Macaulay, the Hilbert series has an explicit form.

**Theorem 2.2** [21, page 35] *Let  $A$  be a Cohen-Macaulay standard graded  $\mathbf{k}$ -algebra with an s.o.p.  $(\theta_0, \dots, \theta_n)$  and let  $\eta_1, \dots, \eta_s$  be homogeneous elements of  $A$  which form a  $\mathbf{k}$ -basis for  $A/(\theta_0, \dots, \theta_n)$ . Then*

$$\mathcal{H}(A) = \frac{\sum_{i=1}^s t^{\deg \eta_i}}{(1-t)^{n+1}}$$

Returning to the example, the Hilbert series of the Stanley-Reisner ring of the triangulated 3-cube is  $\mathcal{H}(\mathbf{k}[\Delta(C_3)]) = (1 + 4t + t^2)/(1 - t)^3$ . In general, the numerator  $\mathbf{k}[\Delta(C_n)]$  is the *Eulerian polynomial*, that is, the degree  $n$  polynomial having the coefficient of  $t^i$  equal to the number of permutations in the symmetric group having exactly  $i$  descents. This definition of the Eulerian polynomial differs slightly from the usual one which instead has each term having degree one more than we have stated. For a general reference on the Eulerian numbers, we refer the reader to [19].

Putting all these results together, we have the following theorem which is also a consequence of work of Hetyei [6]

**Theorem 2.3** *Let  $\theta_0, \dots, \theta_n$  be an s.o.p. of  $\mathbf{k}[\Delta(C_n)]$ , the Stanley-Reisner ring of the triangulated  $n$ -cube. The Hilbert series of the quotient ring  $\mathbf{k}[\Delta(C_n)]/(\theta_0, \dots, \theta_n)$  is given by*

$$\mathcal{H}(\mathbf{k}[\Delta(C_n)]/(\theta_0, \dots, \theta_n)) = \sum_{\pi} t^{\text{des}(\pi)},$$

where the sum is over all permutations  $\pi$  in the symmetric group  $S_n$ .

### 3 The Manin ring $\overline{L}_n$

Let  $n$  be an integer greater than or equal to 2 and let the index set  $N_n$  consist of all ordered pairs  $s = (s_1, s_2)$  with  $s_1 \cup s_2 = [n]$  and  $s_1 \neq \emptyset, [n]$ . Let the ideal  $I_n$  be generated by:

1. The linear relations

$$\lambda(i, j) = \sum_{(s_1, s_2)} x_{(s_1 \cup \{i\}, s_2 \cup \{j\})} - x_{(s_1 \cup \{j\}, s_2 \cup \{i\})},$$

where  $i \neq j$  are fixed integers satisfying  $1 \leq i, j \leq n$  and the ordered pairs  $(s_1 \cup \{i\}, s_2 \cup \{j\})$  and  $(s_1 \cup \{j\}, s_2 \cup \{i\})$  belong to the index set  $N_n$ .

2. The quadratic relations

$$x_{(s_1 \cup \{i\}, s_2 \cup \{j\})} \cdot x_{(t_1 \cup \{j\}, t_2 \cup \{i\})},$$

where  $i \neq j$  are fixed integers satisfying  $1 \leq i, j \leq n$  and the ordered pairs  $(s_1 \cup \{i\}, s_2 \cup \{j\})$  and  $(t_1 \cup \{j\}, t_2 \cup \{i\})$  belong to the index set  $N_n$ .

The *Manin ring*, denoted  $\overline{L}_n$ , is defined by  $\overline{L}_1 = \mathbf{k}$  and for  $n \geq 2$  it is the quotient

$$\overline{L}_n = \mathbf{k}[x_s : s \in N_n]/I_n.$$

Manin used the notation  $\sum_s (x_{isj} - x_{jsi})$  to denote the linear relation  $\lambda(i, j)$  and  $x_{isi} \cdot x_{jti}$  to denote the quadratic relation for  $i$  and  $j$  fixed integers.

**Proposition 3.1** *The following identity holds among the differences  $\lambda(i, j)$ :*

$$\lambda(k, j) = \lambda(i, j) - \lambda(i, k).$$

*In particular, the quantity  $\lambda(i, j)$  can be written as a linear combination of the  $n - 1$  quantities  $\lambda(1, 2), \dots, \lambda(1, n)$ .*

We now introduce the  $2^n$  variables  $y_t$  where  $t \subseteq [n]$ . Let  $y_t = x_{(t, [n] - t)}$  for  $t \neq \emptyset, [n]$ . Define the  $n + 1$  linear forms

$$\theta_0 = y_\emptyset, \quad \theta_i = \lambda(1, i + 1) \text{ for } 1 \leq i \leq n - 1, \quad \text{and} \quad \theta_n = y_{[n]}. \quad (3.2)$$

Here the relations  $\lambda(1, i + 1)$  are written using the variables  $y_t$ .

We now state the main results of this section.

**Lemma 3.2** *Let  $M$  be the  $(n + 1) \times 2^n$  matrix of the  $\theta_i$  given in (3.2). Assume the columns of the matrix  $M$  are ordered using any linear extension of the Boolean algebra  $B_n$ . Let  $F$  be a facet of the triangulated  $n$ -cube. Then*

$$\det(M|_F) = (-1)^{\pi(F)},$$

*that is, the determinant of the matrix  $M|_F$  is precisely the signature of the permutation  $\pi(F)$ .*

**Theorem 3.3** *Let the linear forms  $(\theta_0, \dots, \theta_n)$  be as in (3.2). Then*

- (a) *The  $(\theta_0, \dots, \theta_n)$  form a linear system of parameters for  $\mathbf{k}[\Delta(C_n)]$ , the Stanley-Reisner ring of the triangulated  $n$ -cube.*

$$\begin{aligned}
\mathcal{H}(\bar{L}_1) &= 1 \\
\mathcal{H}(\bar{L}_2) &= 1 + t \\
\mathcal{H}(\bar{L}_3) &= 1 + 4t + t^2 \\
\mathcal{H}(\bar{L}_4) &= 1 + 11t + 11t^2 + t^3 \\
\mathcal{H}(\bar{L}_5) &= 1 + 26t + 66t^2 + 26t^3 + t^4
\end{aligned}$$

Table 2: Values of  $\mathcal{H}(\bar{L}_n)$  for  $1 \leq n \leq 5$ .

(b)  $\bar{L}_n = \mathbf{k}[\Delta(C_n)]/(\theta_0, \dots, \theta_n)$ , that is, the Manin ring is precisely the Stanley-Reisner ring of the triangulated  $n$ -cube modulo a system of parameters.

**Theorem 3.4** *The Hilbert series of the Manin ring is given by*

$$\mathcal{H}(\bar{L}_n) = \sum_{\pi} t^{\text{des}(\pi)},$$

where the sum is over all permutations  $\pi$  in the symmetric group  $S_n$ .

Some values of the Hilbert series are given in Table 2.

As an aside, the permutohedron  $\Pi_n$  is the polytope formed by the convex hull of the vertices  $(\pi_1, \dots, \pi_n) \in \mathbb{R}^n$  with  $\pi_1 \cdots \pi_n \in S_n$ . Since the permutohedron is a simple polytope, its dual polytope  $\Pi_n^*$  is a simplicial polytope. Note that the vertices of  $\Pi_n^*$  correspond to ordered partitions of the elements  $[n]$  into two parts, and in general, the  $i$ -dimensional faces of  $\Pi_n^*$  correspond to ordered partitions of the elements  $[n]$  into  $i+2$  parts. It is not difficult to convince oneself that the Manin ring  $\bar{L}_n$  coincides with the Stanley-Reisner ring of the dual of the permutohedron modulo the  $n-1$  linear relations  $\lambda(1, 2)$  through  $\lambda(1, n)$ .

## 4 A $\mathcal{B}_n$ -analogue of the Manin ring

We now introduce a signed analogue of the Manin ring. In order to do this, we first review the notions of signed sets and signed permutations. Let  $\mathcal{S}_n^\pm$  denote the set of all signed subsets from the set  $\{\pm 1, \pm 2, \dots, \pm n\} = \{1, \bar{1}, 2, \bar{2}, \dots, n, \bar{n}\}$ , that is,  $\{a_1, \dots, a_k\} \in \mathcal{S}_n^\pm$  if  $\{|a_1|, \dots, |a_k|\}$  is a  $k$  element subset of  $[n]$ . We say  $\sigma = \sigma_1 \cdots \sigma_n$  is a *signed permutation* if the set  $\{\sigma_1, \dots, \sigma_n\}$  has cardinality  $n$  and belongs to  $\mathcal{S}_n^\pm$ . An *augmented signed permutation* is of the form  $\tau = \sigma_1 \cdots \sigma_n 0$  where the zero element has been adjoined to the end of the signed permutation  $\sigma$ , in other words,  $\tau_{n+1} = 0$ . We denote the set of all signed permutations on  $n$  elements by  $S_n^\pm$  and the set of all augmented signed permutations by  $S_{n,\text{aug}}^\pm$ . The number of descents of  $\tau \in S_{n,\text{aug}}^\pm$ , denoted  $\text{des}(\tau)$ , is the number of indices  $i$  such that  $\tau_i > \tau_{i+1}$  for  $1 \leq i \leq n$ . Finally, the *signature* of a signed permutation  $\sigma \in \mathcal{S}_n^\pm$  is the signature of the permutation  $\pi = |\sigma_1| \cdots |\sigma_n| \in S_n$  times  $(-1)^\epsilon$ , where  $\epsilon$  is the number of negative signs in  $\sigma$ .

Let  $\mathcal{L}(C_n)$  denote the face lattice of the  $n$ -dimensional cube or cubical lattice. The dual of the rank  $n$  cubical lattice, denoted  $\mathcal{L}(C_n)^*$ , is simply the elements of  $\mathcal{S}_n^\pm$  ordered by inclusion with a maximal element  $\hat{1}$  adjoined. The maximal chains of the cubical lattice correspond to signed permutations in the following manner. Given a maximal chain  $c : \emptyset = Q_0 \subset Q_1 \subset \cdots \subset Q_n < \hat{1}$  in  $\mathcal{L}(C_n)^*$ , let  $\sigma_i$  denote the unique element

in the set difference  $Q_i - Q_{i-1}$  for  $1 \leq i \leq n$ . The permutation representation of the maximal chain  $c$  is the signed permutation  $\sigma = \sigma_1 \cdots \sigma_n$ . Geometrically, the barycentric subdivision of the  $n$ -cube gives  $n! \cdot 2^n$  number of  $n$ -simplices each corresponding to a permutation in  $S_n^\pm$ .

We now define an ideal using the signed variables  $y_s$  for  $s \in S_n^\pm$ . Let  $I_n^\pm$  be the ideal generated by:

### 1. The linear relations

$$\lambda^\pm(i, j) = \sum_s (y_{s \cup \{i\}} + y_{s \cup \{\bar{i}\}}) - (y_{s \cup \{j\}} + y_{s \cup \{\bar{j}\}}),$$

where  $i \neq j$  are fixed positive integers satisfying  $1 \leq i, j \leq n$  and the sum is over all signed subsets  $s$  of  $S_n^\pm$  not containing the elements  $i, \bar{i}, j$  or  $\bar{j}$ .

### 2. The quadratic relations

$$y_s \cdot y_t,$$

where  $s$  and  $t$  are signed subsets of  $S_n^\pm$  that are incomparable elements in the dual of the cubical lattice.

### 3. The relation

$$\lambda_n^\pm = \sum_s (-1)^{\# \text{ signs in } s} \cdot y_s,$$

where  $s$  ranges over all signed subsets of  $S_n^\pm$  with  $|s| = n$ .

Define the *signed Manin ring*  $\overline{L}_n^\pm$  by  $\overline{L}_0^\pm = \mathbf{k}$  and for  $n \geq 1$  it is  $\overline{L}_n^\pm = \mathbf{k}[y_t : t \in S_n^\pm]/I_n^\pm$ .

By virtually the same proofs as in Section 3, we have the following results.

**Proposition 4.1** *The following identity holds among the differences  $\lambda^\pm(i, j)$ :*

$$\lambda^\pm(k, j) = \lambda^\pm(i, j) - \lambda^\pm(i, k).$$

In particular, the quantity  $\lambda^\pm(i, j)$  can be written as a linear combination of the  $n - 1$  quantities  $\lambda^\pm(1, 2), \dots, \lambda^\pm(1, n)$ .

Define the  $n + 1$  linear forms  $\theta_0^\pm, \dots, \theta_n^\pm$  by

$$\theta_0^\pm = y_\emptyset, \quad \theta_i^\pm = \lambda^\pm(1, i+1) \text{ for } 1 \leq i \leq n-1, \quad \text{and } \theta_n^\pm = \lambda_n^\pm. \quad (4.3)$$

**Lemma 4.2** *Let  $M$  be the matrix of the  $\theta_i^\pm$  given in (4.3), where the columns are ordered using any linear extension of the dual of the cubical lattice  $\mathcal{L}(C_n)^*$ . Let  $F$  be an  $n$ -simplex of the barycentric subdivision of the  $n$ -cube with signed permutation representation  $\sigma(F) = \sigma_1 \cdots \sigma_n$ . Then*

$$\det(M|_F) = (-1)^{\sigma(F)},$$

that is, the determinant of the matrix  $M|_F$  is precisely the signature of the signed permutation  $\sigma(F) \in S_n^\pm$ .

**Theorem 4.3** *Let  $(\theta_0^\pm, \dots, \theta_n^\pm)$  be given as in (4.3). Then*

- (a) *The  $(\theta_0^\pm, \dots, \theta_n^\pm)$  form a linear system of parameters for  $\mathbf{k}[\text{Bar}(C_n)]$ , the Stanley-Reisner ring of the barycentric subdivision of the  $n$ -cube.*
- (b)  *$\overline{L}_n^\pm = \mathbf{k}[\text{Bar}(C_n)]/(\theta_0^\pm, \dots, \theta_n^\pm)$ , that is, the signed Manin ring is precisely the Stanley-Reisner ring of the barycentric subdivision of the  $n$ -cube modulo a system of parameters.*

$$\begin{aligned}
\mathcal{H}(\overline{L}_0^\pm) &= 1 \\
\mathcal{H}(\overline{L}_1^\pm) &= 1 + t \\
\mathcal{H}(\overline{L}_2^\pm) &= 1 + 6t + t^2 \\
\mathcal{H}(\overline{L}_3^\pm) &= 1 + 23t + 23t^2 + t^3 \\
\mathcal{H}(\overline{L}_4^\pm) &= 1 + 76t + 230t^2 + 76t^3 + t^4
\end{aligned}$$

Table 3: Values of  $\mathcal{H}(\overline{L}_n^\pm)$  for  $0 \leq n \leq 4$ .

We now state a signed analogue of Theorem 3.4. The fact that the sum is over all augmented signed permutations again follows from the aforementioned result of Björner on EL-labelings.

**Theorem 4.4** *The Hilbert series of the signed Manin ring is given by*

$$\mathcal{H}(\overline{L}_n^\pm) = \sum_{\tau} t^{\text{des}(\tau)},$$

where the sum is over all augmented signed permutations  $\tau \in S_{n,\text{aug}}^\pm$ .

The first few values of the Hilbert series of the signed Manin ring are displayed in Table 3.

The signed permutohedron  $\Pi_n^\pm$  is the polytope formed by taking the convex hull of the vertices  $(\sigma_1, \dots, \sigma_n) \in \mathbb{R}^n$  with  $\sigma_1 \cdots \sigma_n \in S_n^\pm$ . Its face lattice is the lattice of regions of the braid arrangement  $\mathcal{B}_n$  given by  $x_i = \pm x_j$ , for  $1 \leq i < j \leq n$ , and  $x_i = 0$ , for  $1 \leq i \leq n$ . This lattice may also be described as the lattice of ordered signed partitions, denoted  $\mathcal{L}(\Pi_n^\pm)$ . We refer the reader to [4, Section 6] for a more detailed description of the ordered signed partition lattice than the one we will give here.

The elements of  $\mathcal{L}(\Pi_n^\pm)$  are ordered signed partitions  $Z/\widetilde{B}_1/\cdots/\widetilde{B}_k$ , where  $Z$  is a (possibly empty) unsigned set called the zero set, and  $\widetilde{B}_1, \dots, \widetilde{B}_k$  are signed non-empty sets. The order relation in the lattice is merging two adjacent blocks. Hence the coatoms are of the form  $Z/\widetilde{B}_1$  with  $Z = \{i_1, \dots, i_m\}$  and  $\widetilde{B}_1 = \{i_{m+1}, \dots, i_n\}$ . Using the convention  $x_i$  to mean  $-x_i$ , this determines the ray  $0 = x_{i_1} = \cdots = x_{i_m} < x_{i_{m+1}} = \cdots = x_{i_n}$  in the arrangement.

Dualizing all of this, we see the dual of the signed permutohedron has vertices corresponding to elements of the form  $Z/\widetilde{B}_1$ , which in turn corresponds to the variable  $y_{\widetilde{B}_1}$  in the signed Manin ring. In analogy to the Manin ring, it is straightforward to verify that the signed Manin ring is the Stanley Reisner ring of the dual of the signed permutohedron modulo the  $n$  linear relations  $\lambda^\pm(1, 2), \dots, \lambda^\pm(1, n), \lambda_n^\pm$ .

## 5 Concluding remarks

One obvious question to ask is to find other naturally occurring analogues of the Manin ring and the Commutativity Equations. The permutohedron and signed-permutohedron correspond respectively to the Weyl groups  $\mathcal{A}_{n-1}$  and  $\mathcal{B}_n$ . In a forthcoming paper, the author studies the Manin rings arising from other Weyl groups and root systems.

Two other examples which may lead to further results are worth mentioning. First, the notion of augmented signed permutations has a generalization to augmented  $\mathbf{r}$ -signed permutations which corresponds to the  $\mathbf{r}$ -cubical lattice; see [3]. Secondly, Reiner and Ziegler [17] studied the Coxeter-Associahedra, a class of convex polytopes interpolating between the permutohedron and associahedron. Can anything be said about associahedron types of examples?

The Hetyei ring [6] is a cubical analogue of the Stanley-Reisner ring defined for cubical polytopes, and more generally, cubical complexes. It is an example of a ring whose defining ideal is binomial. See [16] for further investigations by the author regarding s.o.p.'s of the Hetyei ring, as well as [5, 7] for the relation of the Hetyei ring with the Ron Adin  $h$ -vector. In [20] Stanley generalizes the Hetyei ring to polytopal complexes.

## 6 Acknowledgements

The author would like to thank Richard Ehrenborg and Gábor Hetyei for reading an earlier draft of this paper.

## References

- [1] A. BJÖRNER, Shellable and Cohen-Macaulay partially ordered sets, *Trans. Amer. Math. Soc.* **260** (1980), 159–183.
- [2] H. BRUGGESSER AND P. MANI, Shellable decompositions of cells and spheres, *Math. Scand.* **29** (1971), 197–205.
- [3] R. EHRENBORG AND M. READDY, The  $\mathbf{r}$ -cubical lattice and a generalization of the cd-index, *European J. Combin.* **17** (1996), 709–725.
- [4] R. EHRENBORG AND M. READDY, On flag vectors, the Dowling lattice and braid arrangements, *Discrete Comput. Geom.* **21** (1999), 389–403.
- [5] J. HAGLUND, On the nonnegativity of the coefficients of some polynomials occurring in the theory of cubical spheres, *Ann. Sci. Math. Québec* **1** (1998), 43–45.
- [6] G. HETYEI, On the Stanley ring of a cubical complex, *Discrete Comput. Geom.* **14** (1995), 305–330.
- [7] G. HETYEI, Invariants des complexes cubiques, *Ann. Sci. Math. Québec* **20** (1996), 35–52.
- [8] M. HOCHSTER, Rings of invariants of tori, Cohen-Macaulay rings generated by monomials, and polytopes, *Annals of Math.* **96** (1972), 318–337.
- [9] B. KIND AND P. KLEINSCHMIDT, Schälbare Cohen-Macauley-Komplexe und ihre Parametrisierung, *Math. A.* **167** (1979), 173–179.
- [10] M. KONTSEVICH AND Y. MANIN, Gromov-Witten classes, quantum cohomology and enumerative geometry, *Comm. Math. Phys.* **164:3** (1994), 525–562.
- [11] M. KONTSEVICH AND Y. MANIN, Quantum cohomology of a product (with Appendix by R. Kaufmann), *Inv. Math.* **124**, f. 1–3 (1996), 313–339.
- [12] M. KONTSEVICH AND Y. MANIN, Relations between the correlators of the topological sigma-model coupled to gravity, *Comm. Math. Phys.* **196** (1998), 385–398.
- [13] A. LOSEV, Commutativity equations, operator-valued cohomology of the “sausage” compactification of  $(\mathbb{C}^*)^N/\mathbb{C}^*$  and SQM, preprint ITEP-TH-84/98, LPTHE-61/98.
- [14] A. LOSEV, Passing from solutions to Commutativity Equations to solutions to Associativity Equations and background independence for gravitational descendants, preprint ITEP-TH-85/98, LPTHE-62/98.

- [15] A. LOSEV AND Y. MANIN, New modular spaces of pointed curves and pencils of flat connections, preprint arXiv: math.AG/0001003 January 1, 2000.
- [16] M. READDY, Systems of parameters for cubical complexes, preprint 1999.
- [17] V. REINER AND G. ZIEGLER, Coxeter-Associahedra, *Mathematika* **41** (1994), 364–393.
- [18] R. P. STANLEY, The Upper Bound Conjecture and Cohen-Macaulay rings, *Studies in Appl. Math.* **54** (1975), 135–142.
- [19] R. P. STANLEY, “Enumerative Combinatorics, Vol. I,” Wadsworth and Brooks/Cole, Pacific Grove, CA, 1986.
- [20] R. P. STANLEY, A monotonicity property of  $h$ -vectors and  $h^*$ -vectors, *European J. Combin.* **14** (1993), 251–258.
- [21] R. P. STANLEY, “Combinatorics and commutative algebra, second edition,” Birkhäuser Boston, Inc., Boston, MA, 1996.

MARGARET A. READDY  
DEPARTMENT OF MATHEMATICS  
STOCKHOLM UNIVERSITY  
S-106 91 STOCKHOLM  
SWEDEN  
[readdy@matematik.su.se](mailto:readdy@matematik.su.se)

# Squashing animals and anisotropic generating functions.

Andrew Rechnitzer

Department of Mathematics and Statistics  
The University of Melbourne, Vic 3010, Australia.  
email: andrewr@ms.unimelb.edu.au

## Abstract

We present a method for examining some of the properties of the anisotropic generating function of many (commonly studied) lattice bond animals. In particular if one considers the anisotropic generating function  $P(x, y)$ , then the coefficient of  $y^n$  is a rational function whose denominator is a product of cyclotomic polynomials. The method can determine which cyclotomic factors can occur and also bound their exponents. In certain cases, most notably self-avoiding polygons (SAPs), these bounds can be sharpened to equalities. This proves that the anisotropic SAP generating function is not solvable in terms of D-finite functions.

## Résumé

Nous présentons une méthode pour étudier la fonction génératrice anisotope de plusieurs familles classiques d'animaux sur un réseau. En particulier, nous prouvons que le coefficient de  $y^n$  dans la fonction génératrice anisotope  $P(x, y)$  est une fonction rationnelle de  $x$  dont le dénominateur est un produit de polynômes cyclotomiques. Cette méthode permet de déterminer quels facteurs cyclotomiques interviennent et également de borner leurs exposants. Dans certains cas, notamment pour les polygones auto-évitants, ces bornes sont atteintes, ce qui prouve que la fonction génératrice anisotope des polygones auto-évitants n'est pas D-finie.

## 1 Properties of anisotropic generating functions

The enumeration of lattice animal models is arguably one of the most famous problems in combinatorics, and is of considerable importance in the study of lattice models in statistical physics and theoretical chemistry. Considering the intensive study that these models have been subjected to over their 40+ year history, it is perhaps a little surprising that the number of rigorous results concerning these models is very small, and the number of models that have been solved exactly (either implicitly or explicitly) smaller still.

There are two basic types of lattice animals; site animals and bond animals. A site animal is a finite connected set of lattice vertices (or sites), while a bond animal is a finite connected set of lattice edges (or bonds); both are defined up to translation. In this paper we will only study bond animals on the square lattice. We will write “animal” to mean “square lattice bond animal”.

Ideally one would like to enumerate a family of bond animals, such as self-avoiding polygons or directed bond animals, according to the number of bonds contained. The

history of animal enumeration on the square lattice would suggest that our attempts are likely to be frustrated<sup>1</sup>. In fact, of the models, listed in section 1.2 below, exact solutions are only known for those models with severe topological restrictions: spiral walks (see [1] for example), 3-choice polygons [2] and a number of families of column-convex polygons (see [3] for example). Rather than commencing an almost certainly doomed attempt to find the generating function of a family of animals we will attempt to determine some of the properties of the solution by examining their *anisotropic* generating function.

## 1.1 Anisotropic generating functions

The *isotropic* generating function of a family of animals enumerates the animals according to the total number of bonds. The *anisotropic* generating function, on the other hand, distinguishes between horizontal and vertical bonds. For a given bond animal  $B$ , we denote the number of horizontal (resp. vertical) bonds it contains by  $|B|_{\leftrightarrow}$  (resp.  $|B|_{\uparrow\downarrow}$ ).

Let  $\mathcal{G}$  be a set of bond animals on the square lattice. Let us enumerate the elements of  $\mathcal{G}$  according to the number of horizontal and vertical bonds and form the *anisotropic* generating function:

$$\begin{aligned} gf(\mathcal{G}) &= \sum_{Q \in \mathcal{G}} x^{|Q|_{\leftrightarrow}} y^{|Q|_{\uparrow\downarrow}} \\ &= \sum_{n,m=0}^{\infty} c_{n,m} x^n y^m, \end{aligned}$$

where  $c_{n,m}$  is the number of elements of  $\mathcal{G}$  containing exactly  $n$  horizontal bonds and  $m$  vertical bonds.

If the animal is a polygon, then the numbers of vertical and horizontal bonds are always even numbers, and so rather introducing extraneous factors of 2 we will enumerate families of polygons according to their horizontal and vertical *half-perimeters* (being exactly half the number of horizontal and vertical bonds). Rather than defining different notation for polygons we will simply take  $|B|_{\leftrightarrow}$  and  $|B|_{\uparrow\downarrow}$  to mean the horizontal and vertical half-perimeters (respectively) of any polygon  $B$ .

Let us start by summing this generating function to make it a power series in  $y$  with coefficients that are power series in  $x$ . Writing  $\mathcal{G}_n = \{Q \in \mathcal{G}_n : |Q|_{\uparrow\downarrow} = n\}$  we have:

$$gf(\mathcal{G})(x, y) = \sum_{n=1} y^n \sum_{Q \in \mathcal{G}_n} x^{|Q|_{\leftrightarrow}} = \sum_{n=1} H_n(x) y^n.$$

The coefficient of  $y^n$  in the above generating function,  $H_n(x)$ , is a power series in  $x$  that enumerates all bond animals in  $\mathcal{G}$  containing  $n$  vertical bonds (the set  $\mathcal{G}_n$ ), according to the number of horizontal bonds.

Very generally speaking it is not too difficult to extend isotropic techniques (be they exact solutions or numerical expansions) to the anisotropic case, so one can obtain the anisotropic generating function without having to do much more work. The anisotropic generating function is somehow a more manageable object than the isotropic. Splitting the set of animals  $\mathcal{G}$ , into separate simpler subsets,  $\mathcal{G}_n$ , gives us smaller pieces, each of which is easier to study than the whole.

---

<sup>1</sup>This is an understatement of the extreme difficulty that has been encountered by those venturing into this area of enumerative combinatorics.

As noted above, the only families of bond animals that have been solved are those with severe topological restrictions. If one seeks to understand the *isotropic* generating function then one must somehow examine *all* possible topologies or configurations<sup>2</sup> that can occur in  $\mathcal{G}$ . On the other hand, if we examine the generating function of  $\mathcal{G}_n$ , then the number of different topologies that can occur is always finite. For example, consider self-avoiding polygons with  $2n$  vertical bonds. If  $n = 1$  all configurations are rectangles. If  $n = 2$ , then all configurations are vertically *and* horizontally convex, while if  $n = 3$  all configurations are vertically *or* horizontally convex. The anisotropy allows one to study the effect that these configurations have on the generating function in a more controlled manner.

Similarly, instead of trying to study the properties of the whole generating function (that may not be known), the anisotropy breaks the generating function into separate simpler pieces,  $H_n(x)$ , that can be calculated exactly. By studying the properties of these coefficients we can obtain some idea of the properties of the generating function as a whole. This procedure has been proposed [4] as a test of the solvability of a model (see below), and has been applied to a wide variety of models [5].

It has also been observed [6] that the anisotropic generating functions of many (solved and unsolved) families of animals obey a type of functional symmetry known as *reciprocity* or *inversion relations*. These are well known in statistical mechanics and in certain cases can lead to an exact solution. These symmetries are not exhibited in the isotropic generating function.

## 1.2 A few definitions

We will apply the techniques of this paper to several families of animals. Here are some brief definitions of animals we will consider.

**Definition 1.** Some families of bond animals on the square lattice:

1. bond animals — connected set of bonds.
2. directed bond animals — a bond animal for which all bonds can be reached by a directed path (taking only north and east steps) from the origin.
3. lattice trees — bond animals that contain no loops or cycles.
4. self-avoiding walks (SAWs) — bond animals, for which all vertices are degree 2, except two of degree 1.
5. spiral walks — a SAW for which clockwise turns are forbidden.
6. 3-choice walks — a SAW for which clockwise turns are forbidden after a step in the  $\pm x$  direction.
7. 2-choice walks — a 3-choice walk that is forbidden to make  $> 1$  successive steps in the  $\pm y$  direction.
8. self-avoiding polygons (SAPs) — a bond animal for which all vertices are of degree 2, or equivalently the embedding of a simple closed loop in the square lattice.
9. 3-choice polygons — a SAP whose perimeter obeys the 3-choice walk rule. One can similarly define 2-choice polygons.
10. column-convex polygons — a SAP which contains only 2 horizontal bonds in each column.

---

<sup>2</sup>We deliberately use imprecise words here. We will be more precise below.

11. staircase polygons — a column-convex polygon whose upper and lower edges are directed walks (use only  $+x$  and  $+y$  steps).

Examples of these animals are illustrated in figure 1.

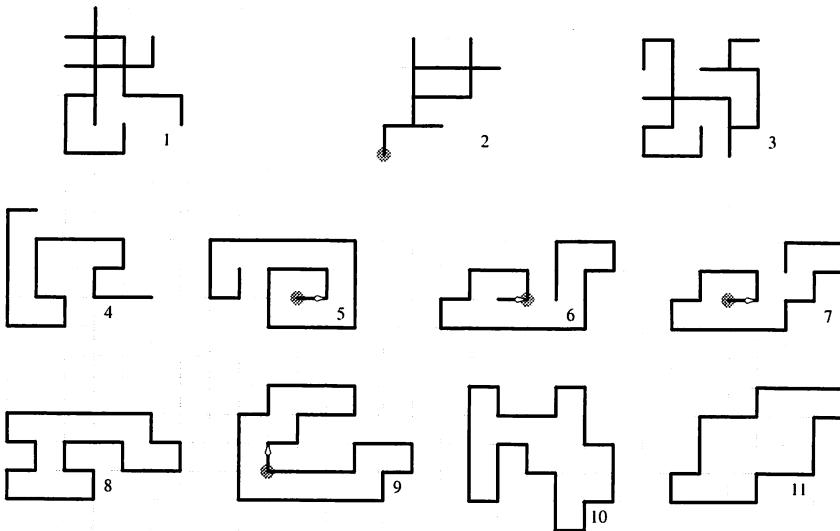


Figure 1: Examples of bond-animals. The starting vertices of the spiral walk, 2-choice walk, 3-choice walk and polygon, and the root vertex of directed bond animal are highlighted.

In the work that follows we will show that the generating functions,  $H_n(x)$ , can be expressed in terms of generating functions of the form  $\frac{x^k}{1-x^k}$ . The denominator polynomial,  $(1-x^k)$ , can be factored into *cyclotomic polynomials*.

**Definition 2.** The cyclotomic polynomials,  $\Psi_k(x)$ , are the factors of the polynomials  $(1-x^n)$ . In particular,  $(1-x^n) = \prod_{k|n} \Psi_k(x)$ .

Hence for any given integers  $\{c_i\}$ , the we have the following factorisation

$$\prod_{n=1} \left(1 - x^n\right)^{c_n} = \prod_{n=1} \prod_{k|n/d} \Psi_k(x)^{c_n} = \prod_{k=1} \Psi_k(x)^{\sum_{d=1} c_k \times d}$$

The first few cyclotomic polynomials are below in bold.

$$\Psi_1 : (1-x) = \mathbf{(1-x)}$$

$$\Psi_2 : (1-x^2) = (1-x)(1+x)$$

$$\Psi_3 : (1-x^3) = (1-x)(1+x+x^2)$$

$$\Psi_4 : (1-x^4) = (1-x)(1+x)(1+x^2)$$

$$\Psi_5 : (1-x^5) = (1-x)(1+x+x^2+x^3+x^4)$$

$$\Psi_6 : (1-x^6) = (1-x)(1+x)(1+x+x^2)(1-x+x^2)$$

We call  $\Psi_k(x)$  the  $k$ th cyclotomic polynomial, and say that it its *order* is  $k$ .

### 1.3 Some examples

Let us consider two examples of these anisotropic generating functions; staircase polygons and self-avoiding polygons.

**Example 1 (Staircase polygons).** The anisotropic generating function of staircase polygons is known in closed form [7]:

$$P(x, y) = \frac{1}{2} \left( 1 - x - y - \sqrt{(1 - x - y)^2 - 4xy} \right)$$

Expanding  $P(x, y)$  as a power series in  $y$  gives:

$$\begin{aligned} P(x, y) &= \frac{x}{1-x}y + \frac{x}{(1-x)^3}y^2 + \frac{x(1+x)}{(1-x)^5}y^3 \\ &\quad + \frac{x(1+3x+x^2)}{(1-x)^7}y^4 + \frac{x(1+6x+6x^2+x^3)}{(1-x)^9}y^5 + \dots \end{aligned}$$

The coefficients,  $H_n(x)$ , have the following properties:

- $H_n(x)$  is a rational function of  $x$ ,
- the degree of the numerator of  $H_n(x)$  is  $(n - 1)$ , for all  $n \geq 2$ .
- the denominator of  $H_n(x)$  is  $(1 - x)^{2n-1}$ .
- the coefficients of the numerators are positive, symmetric and unimodal.

All other polygon models for which a closed form solution is known (all of which are subsets of column-convex polygons) have similar properties (though the symmetry of numerator coefficients is lost in some cases) [8].

**Example 2 (Self-avoiding polygons).** The anisotropic generating function of self-avoiding polygons,  $P(x, y)$ , remains elusive, and to date the best algorithm for computing the coefficients of  $P$  is the finite lattice method, which still requires exponential time [9], but is exponentially more efficient than direct enumeration.

Expanding  $P(x, y)$  as a power series in  $y$ , one finds [10] that the coefficients (which have been computed up to order  $y^{14}$ ),  $H_n(x)$  have the following properties:

- $H_n(x)$  is a rational function of  $x$ ,
- the degree of the numerator of  $H_n(x)$  is equal to the degree of its denominator.
- the coefficients of the numerators are positive and unimodal, but not symmetric.
- if we write the denominator of  $H_n(x)$  as  $D_n(x)$ , then the first ten are:

$$\begin{aligned} D_1(x) &= (1 - x) \\ D_2(x) &= (1 - x)^3 \\ D_3(x) &= (1 - x)^5 \\ D_4(x) &= (1 - x)^7 \\ D_5(x) &= (1 - x)^9(1 + x)^2 \\ D_6(x) &= (1 - x)^{11}(1 + x)^4 \\ D_7(x) &= (1 - x)^{13}(1 + x)^6(1 + x + x^2) \\ D_8(x) &= (1 - x)^{15}(1 + x)^8(1 + x + x^2)^3 \\ D_9(x) &= (1 - x)^{17}(1 + x)^{10}(1 + x + x^2)^5 \\ D_{10}(x) &= (1 - x)^{19}(1 + x)^{12}(1 + x + x^2)^7(1 + x^2) \end{aligned}$$

These suggests that a new cyclotomic factor enters every third coefficient, and that it enters with exponent 1, and increases by 2 in each subsequent coefficient (with the exception of  $(1+x)$  which has exponent 1 less than this pattern predicts).

The denominator structure of self-avoiding polygons is starkly different to that of the previous (solvable) example. The denominators of staircase polygons contain only a single cyclotomic factor,  $(1-x)$ , and so  $H_n(x)$  contains only a finite number of poles as  $n \rightarrow \infty$ . On the other hand, if one extrapolates from the observed pattern of self-avoiding polygon denominators, then any cyclotomic factor will appear eventually, and so  $H_n(x)$  has a dense set of poles on  $|x|=1$  as  $n \rightarrow \infty$ . This has strong (pessimistic) implications for the analytic nature and solvability of the self-avoiding polygon generating function (which we will discuss below).

Similar properties have been observed in the generating functions of many families of bond animals on the square lattice including self-avoiding walks, bond animals, directed bond animals and lattice trees [10]. These properties have also been observed in the thermodynamic functions of the Ising and Potts models [4]. Such functions can be interpreted as enumerating families of graphs on the square lattice with complicated weights (the weights can be negative and the graphs can be disconnected). It would be *very nice indeed* if we were able to extend the techniques outlined in this paper to such problems, however the author suspects that the complicated nature of the weights will make this far from trivial.

We can summarise the (observed) properties of  $H_n(x)$  for general families of bond animals as follows:

1.  $H_n(x)$  is a rational function of  $x$ ,
2. the degree of the numerator of  $H_n(x)$  is less than or equal to the degree of the denominator,
3. the denominators contain only *cyclotomic* polynomial factors,
4. new cyclotomic factors enter the generating functions in a simple predictable way, and
5. the exponents of these factors seem to satisfy simple predictable patterns.

Unfortunately we are not able to prove all of these properties for very general families of animals. We are able to prove the following for any<sup>3</sup> family of animals:

1.  $H_n(x)$  is a rational function of  $x$ ,
2. the degree of the numerator of  $H_n(x)$  is less than or equal to the degree of its denominator,
3. the denominators contain only *cyclotomic* polynomial factors, and
4. only certain cyclotomic factors can occur in the the denominator of  $H_n(x)$ .

We also offer some explanation as to why:

5. new cyclotomic factors enter the generating functions in a predictable way, and
6. the exponents of these factors satisfy simple predictable patterns.

In fact, we have been able to prove (5) (and most of (6)) for self-avoiding polygons and hope to do so for bond animals, directed bond animals and lattice trees in the near future.

---

<sup>3</sup>There are a few technical restrictions, but most animals of interest do satisfy them.

## 2 Squashing animals with partial orders

Consider the set of self-avoiding polygons that contain 2 vertical bonds — this is simply the set of all rectangles of height 1. The horizontal half-perimeter generating function of these polygons is clearly  $\sum_{n \geq 1} x^n = \frac{x}{1-x}$ .

The smallest polygon (or the minimal polygon) in this set is the unit square. We can then obtain the other polygons from the unit square by “stretching” or “growing” the horizontal bonds. The unit square has generating function simply given by  $x$ , stretching the horizontal bonds to length  $n$  gives an  $n$  by 1 rectangle that contributes  $x^n$  to the generating function. Summing over all possible “stretches” gives  $\sum_{n \geq 1} x^n = \frac{x}{1-x}$  as required.

By reversing the stretching process, we can think of squashing the rectangles into shorter and shorter rectangles until we reach the unit square. This squashing process gives a (total) order on this set. The smallest element of this set under this order is the unit square. This idea can be extended to other animals, and we will introduce two different ways of “squashing” general animals. By “stretching” the squashed animals we can examine properties of anisotropic generating functions.

### 2.1 Columns, sections and partial orders

**Definition 3.** We will define a *column* of an animal to be the horizontal bonds within a single horizontal lattice spacing of an animal. See figure 2. If the column contains  $k$  horizontal bonds we say it is a  $k$ -*column*. The number of  $k$ -columns in an animal,  $A$ , is denoted by  $\gamma_k(A)$ .

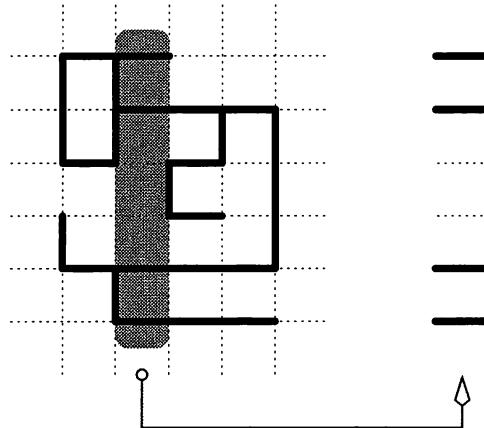


Figure 2: A 4-column of an animal.

**Definition 4.** We construct the *section lines* of an animal in the following way. Draw horizontal lines from the extreme left and the extreme right of the lattice towards the animal so that the lines run through the middle of each lattice cell. The lines are terminated when they first touch a vertical bond (see figure 3).

An animal is split into *pages* in the following way. Cut the lattice along each section line from infinity until it terminates at a vertical bond. Then cut vertically (in both

directions) from this vertical bond to the next section line. If the section line arrives from the left, then vertical bonds along the cut remain on the right hand side (and *vice-versa*). Each separate region obtained in this way is a *page* (see figure 3).

We call a *section* the set of horizontal bonds within a single column of a given page. Equivalently, it is the set of horizontal bonds of a column of an animal between two neighbouring section lines. A section containing  $k$  horizontal bonds is a  $k$ -*section*. The number of  $k$ -sections in an animal,  $A$ , is denoted by  $\sigma_k(A)$ .

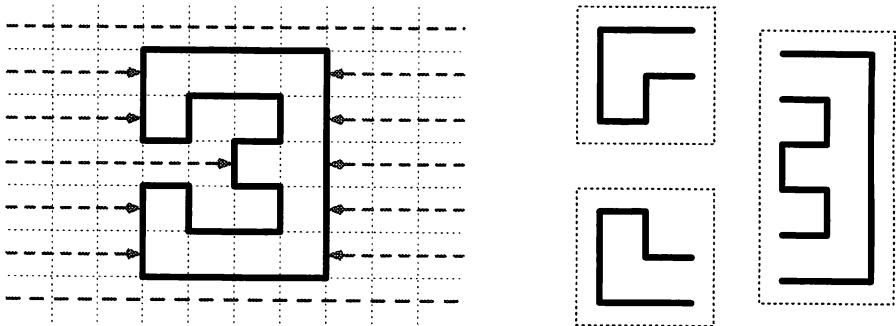


Figure 3: *Section lines* (the heavy dashed lines) split the animal (in this example it is a polygon) into *pages*. Each column in a page is a *section*. This polygon is split into 3 pages, each with 2 sections.

**Definition 5.** We say that a column is a *duplicate column* if the column immediately on its left (without loss of generality) is identical (see figure 4) and there are no vertical bonds between them. We similarly define a *duplicate section*.

One can squash or reduce animals by *deletion* of duplicate columns by slicing the animal on either side of the duplicate column, removing the column and recombining the animal, as illustrated in figure 4). By reversing the column deletion process we define *duplication* of a column. We define *section-deletion* and *section-duplication* in an analogous manner.

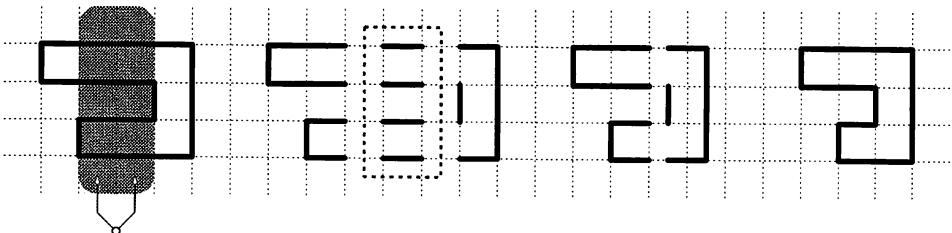


Figure 4: Animal surgery — the process of column deletion. The two indicated columns are identical. Slice either side of the duplicate and separate the polygon into three pieces. The middle piece, being the duplicate, is removed and the remainder of the polygon is recombined. Reversing the steps leads to column duplication.

Let us take an animal,  $B$ , and delete one (or more) of its columns or sections to obtain another animal,  $A$ . We can construct some ordering on the set of animals so that when we examine  $A$  and  $B$ , we can state that  $A \preceq B$ .

**Definition 6.** For any two animals  $P, Q \in \mathcal{G}_n$ , we define the binary relations  $\preceq_c$  and  $\preceq_s$  by stating that:

- $P \preceq_c Q$  if  $P = Q$  or  $P$  can be obtained from  $Q$  by a sequence of column deletions, and
- $P \preceq_s Q$  if  $P = Q$  or  $P$  can be obtained from  $Q$  by a sequence of section deletions.

See figure 5 for example.

From this definition we immediately obtain the following lemma

**Lemma 1.** *The binary relations  $\preceq_c$  and  $\preceq_s$  are partial orders on the set of animals.*

## 2.2 What can we do with these partial orders?

If we take an animal and start to remove duplicate columns, then we cannot reduce the animal to nothing. At some point an animal that contains no duplicate columns must be reached. It does not matter in what order the duplicate columns are removed, we will always obtain the same animal. We say that an animal that is minimal for  $\preceq_c$  is *column-minimal*, while an animal that is minimal for  $\preceq_s$  is *section-minimal*. We note that if an animal is section-minimal then it must be column-minimal, but the converse is false (see figure 5).

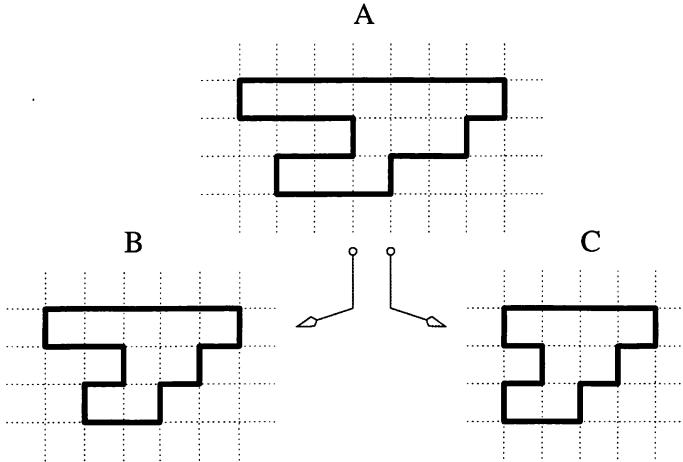


Figure 5: Polygon  $A$  is reduced by a sequence of column deletions to polygon  $B$  (which is column-minimal) and a sequence of section deletions to polygon  $C$  (which is section-minimal).  $B$  can be reduced to  $C$  by section deletions, and hence is not section-minimal.

It can be shown that the set of bond animals with  $n$  vertical bonds,  $\mathcal{G}_n$ , contains only a finite number of minimal animals. Further, since each animal reduces to a unique minimal animal, we can consider  $\mathcal{G}_n$  to be the disjoint union of a finite number of sets, each containing a single minimal animal — each of these sets forms an equivalence class.

Let us denote the horizontal bond generating function of the equivalence class with column-minimal animal  $P$ , by  $G_c(P)$ . Similarly denote the generating function of the equivalence class with section-minimal animal,  $Q$  by  $G_s(Q)$ . These generating functions are given by:

$$G_c(P) = \prod_k \left( \frac{x^k}{1-x^k} \right)^{\gamma_k(P)} \quad (1)$$

$$G_s(P) = \prod_k \left( \frac{x^k}{1-x^k} \right)^{\sigma_k(P)} \quad (2)$$

One has to be a little careful to ensure that the animals do not have any “strange” restrictions placed on them (e.g. having a prime number of horizontal bonds). Most animals of interest do satisfy this technical requirement.

Since  $\mathcal{G}_n$  is split into these equivalence classes, the generating function  $H_n(x)$  can be written as the sum of the generating function of each equivalence class. Further, since there are only a finite number of these equivalence classes, the sum is finite and we have the following theorem:

**Theorem 1.** *If  $P(x, y) = \sum_{n \geq 0} H_n(x)y^n$  is the anisotropic generating function of some set of animals,  $\mathcal{G}$ , then*

- $H_n(x)$  is a rational function,
- the degree of the numerator of  $H_n(x)$  is at most equal to the degree of its denominator, and
- the denominator of  $H_n(x)$  is a product of cyclotomic polynomials.

We can then sharpen the above result to determine which cyclotomic factors can appear in the denominator of  $H_n(x)$ , by noting that the denominator of  $G_s(P)$  can only contain the cyclotomic factor  $\Psi_k(x)$  if  $P$  contains a  $K$ -section, where  $K$  is some integer multiple of  $k$ .

### 2.3 The poles of $H_n(x)$

**Theorem 2 (Poles, columns and sections).** *If  $H_n(x)$  has a denominator factor  $\Psi_k(x)$ , then  $\mathcal{G}_n$  must contain a column-minimal animal, that contains a  $K$ -column for some  $K \in \mathbb{Z}^+$  divisible by  $k$ . Further if  $H_n(x)$  has a denominator factor  $\Psi_k(x)^\alpha$ , then  $\mathcal{G}_n$  must contain a column-minimal animal, that contains a total of  $\alpha$   $K$ -columns for some (possibly different)  $K \in \mathbb{Z}^+$  divisible by  $k$ .*

*Similar results hold for  $k$ -sections and section-minimal animals.*

## 3 Results

Theorem 2 tells us that if there is no animal in  $\mathcal{G}_n$  that contains a  $k$  section (or a  $K$ -section for any  $K$  being an integer multiple of  $k$ ), then the denominator of  $H_n(x)$  does not contain a factor of  $\Psi_k(x)$ . Further, if there is no minimal animal with  $\alpha$   $k$ -sections in  $\mathcal{G}_n$  (or a total of  $\alpha$   $K$ -sections for any  $K$  being an integer multiple of  $k$ ), then the denominator of  $H_n(x)$  cannot contain a factor of  $\Psi_k(x)^\alpha$ .

**Corollary 1.** *We have the following results on the denominators of various families of bond animals:*

- Any subset of column convex polygons can only contain denominator factors  $(1-x)$ .
- Any subset of row convex polygons can only contain denominator factors  $(1-x)$ .
- 3 choice polygons can only contain denominator factors  $(1-x)$  and  $(1+x)$ .
- The factor  $\Psi_k(x)$  for  $k > 1$  cannot enter the denominator of  $H_n(x)$  of spiral, 3-choice and 2-choice walks for  $n < \binom{k}{2} + 1$ . The factor,  $\Psi_1(x) = (1-x)$ , first enters in  $H_0(x)$ .
- For any family of animals containing  $n$  vertical bonds,  $\mathcal{G}_n$ , the horizontal bond generating function  $H_n(x)$ , cannot contain the denominator factor  $\Psi_k(x)$  if  $n < 2k - 2$ .
- The exponent of  $\Psi_k(x)$  in the denominator of  $H_{2k-2}(x)$  is at most  $k$ .
- The factor  $\Psi_k(x)$  cannot enter the denominator of  $H_n(x)$  of self-avoiding polygons for  $n < 3k - 2$ .
- The exponent of  $\Psi_k(x)$  in the denominator of  $H_{3k-2}(x)$  for SAPs is at most 1.

It is worth noting that in proving the above corollary we found that the animals that first contain a  $k$ -section tend to have a relatively simple characterisation. For example, the first bond animals to contain a  $k$ -section contain  $k-1$  rows, each containing 2 vertical bonds. By performing relatively simple “surgery” on these simple animals we can find section-minimal animals with multiple  $k$ -sections. Examination of these allows us to extend the above results to find upper bounds for the exponents of  $\Psi_k(x)$  in general  $H_n(x)$ . These upper bounds follow simple patterns. e.g. for SAPs, the exponent of  $\Psi_k(x)$  in  $H_{3k-2+l}$  is at most  $2l+1$ .

## 4 From bounds to equalities

How good are these bounds? In the case of the exactly solved models (column convex, row convex polygons) expansion of the solution shows that the denominator of  $H_n(x)$  contains only factors of  $(1-x)$ , and it has been shown that the denominators of 3-choice polygons contain factors of  $(1-x)$  and  $(1+x)$  but no other factors [8].

Unsolved models must be examined numerically and the author thanks I. Jensen for providing his recent unpublished series for bond animals, directed bond animals and lattice trees and self-avoiding polygons [10]. These indicate (as far as has been calculated) that the lower bounds of corollary 1 are equalities; e.g. for bond animals the factor  $\Psi_k(x)$  first appears in the denominator  $H_{2k-2}(x)$  with exponent *exactly equal* to  $k$ , while for SAPs  $\Psi_k(x)$  appears in  $H_{3k-2}(x)$  with exponent *exactly equal* to 1 (except when  $k=2$ ). This indicates that the denominators predicted by the above techniques are very close to the exact denominators (sharing no common factors with the numerators). This gives some hope that one might be able to *prove* that there are no cancellations using some extension of these techniques. We have made some progress in this direction.

Proving that there are no cancellations at all between numerators and (predicted) denominators appears to be a very difficult problem. It is far simpler to consider the first occurrence of  $\Psi_k(x)$ . As noted above, one finds that the animals that first contain a  $k$ -section have relatively simple characterisations, and on the basis of this it is not unreasonable to hope that they (or a simple superset of them) can be enumerated. Once enumerated, analysis of their generating function will show whether or not there are any cancellations, which will determine the exponent of  $\Psi_k(x)$ .

We have done exactly this for self-avoiding polygons. We have shown that the factor  $\Psi_k(x)$  appears in the denominator of  $H_{3k-2}(x)$  with exponent *exactly equal to 1* (excepting when  $k = 2$ ). This implies that the singularities of the coefficients of  $y^n$ , form an infinite set that is dense on the unit circle,  $|x| = 1$ , as  $n \rightarrow \infty$ . Numerical evidence [10, 4] would suggest such behaviour in many unsolved models.

Such a nasty analytic property means that the *anisotropic* SAP generating function,  $P(x, y)$ , is not solvable in terms of algebraic functions (those satisfying an algebraic equation), nor D-finite functions [11, 12]. *i.e.* when considered a power series in  $y$  with coefficients that are power series in  $x$ , it does not satisfy any linear differential equation with polynomial coefficients. In fact  $P(x, y)$  is excluded from an even larger class of functions [5]<sup>4</sup>.

In contrast, all explicitly solved column-convex animals are algebraic (see [3] for example) and 3-choice polygons are D-finite [8]. This points to a very marked difference in analyticity between the models that we really would like to solve and the ones that we have been able to solve to date. This difference between the denominators of solved and unsolved models led Guttmann & Enting to propose examination of the anisotropic generating function as a numerical test of the solvability of a model [4]. If one finds that  $H_n(x)$  has only a few poles for all  $n$ , then the model is probably solvable, while if the poles of  $H_n(x)$  become dense as  $n \rightarrow \infty$ , then the model probably has a non-D-finite solution and is going to be much harder to tackle.

## 5 Summary of results

We have introduced a method for the examination of the anisotropic generating function of general families of animals,  $P(x, y)$ . The coefficient of  $y^n$  in this function is found to be a rational function whose denominator is a product of cyclotomic polynomials. We can determine upper bounds for the exponents of the factors appearing in the denominators of these coefficients by examining section-minimal and column-minimal animals.

In the case of self-avoiding polygons these upper bounds can be tightened into equalities, which prove that the singularities of the coefficients of  $y$  in the anisotropic generating function form an infinite set that becomes dense on  $|x| = 1$ . This implies that self-avoiding polygons, enumerated anisotropically, are not solvable in terms of D-finite functions. We hope to prove similar results for bond animals, directed bond animals and lattice trees.

## References

- [1] H. W. J. Blöte and H. J. Hilhorst. Spiral walks i think? *Journal of Physics A*, 17:L111, 1984.
- [2] M. Delest A. R. Conway and A. J. Guttmann. On the number of three choice polygons. *Mathematical and Computer Modelling*, 26:51–58, 1997.
- [3] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Mathematics*, 154:1–25, 1996.
- [4] A. J. Guttmann and I. G. Enting. Solvability of some statistical mechanical systems. *Physical Review Letters*, 76:344–347, 1996.

---

<sup>4</sup>What this result implies about the *isotropic* generating function  $P(x, x)$  is unclear; it is (unfortunately) all too easy to construct a non-D-finite function  $P(x, y)$  for which  $P(x, x)$  is D-finite

- [5] A. J. Guttmann. Indicators of solvability for lattice models. *Discrete Mathematics*, To appear.
- [6] W. P. Orrick M. Bousquet-Mélou, A. J. Guttmann and A. Rechnitzer. Inversion relations, reciprocity and polyominoes. *Annals of Combinatorics*, 3:223–249, 1999.
- [7] K. Y. Lin and S. J. Chang. Rigorous results for the number of convex polygons on the square and honeycomb lattices. *Journal of Physics A*, 21:2635–2642, 1988.
- [8] M. Bousquet-Mélou. Denominators of anisotropic 3-choice polygons. Personal communication with author.
- [9] I. G. Enting. Generating functions for enumerating self-avoiding rings on the square lattice. *Journal of Physics A*, 13:3713–3722, 1980.
- [10] I. Jensen. Anisotropic series for bond animals, directed bond animals and lattice trees. Personal communication with author.
- [11] R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1:175–188, 1980.
- [12] L. Lipshitz. D-finite power series. *Journal of Algebra*, 122:353–373, 1989.

# COUNTING LATTICE PATHS WITH VARIOUS STEP SETS

ROBERT A. SULANKE  
BOISE STATE UNIVERSITY  
BOISE, ID, USA

## Abstract

Focusing on bijective methods, we count lattice paths having various sets of permitted steps. The selected step sets yield path counts expressible in terms of Narayana polynomials. Our main result answers an open problem in Stanley's recent book by giving a bijection relating paths using quite general steps to paths using the steps  $(0, 1)$ ,  $(1, 0)$ , and  $(1, 1)$ .

## Résumé

Utilisant des méthodes bijectives, nous énumérons les chemins du plan qui utilisent différents ensembles de pas permis. Les ensembles de pas choisis produisent des énumérations de chemins qui s'expriment en fonction des polynômes de Narayana. Notre résultat principal donne une réponse à un problème ouvert présenté dans le livre récent de Stanley en définissant une bijection entre des chemins qui utilisent des pas tout à fait généraux et des chemins qui utilisent seulement les pas  $(0,1)$ ,  $(1,0)$  et  $(1,1)$ .

## 1. INTRODUCTION

Let  $A(n)$  denote the set of all lattice paths from  $(0, 0)$  to  $(n, n)$  with steps from  $\mathbb{N} \times \mathbb{N} - \{(0, 0)\}$ , where  $\mathbb{N}$  denotes the nonnegative integers. Let  $D(n)$  denote the set of all lattice paths from  $(0, 0)$  to  $(n, n)$  with steps in  $\{(0, 1), (1, 0), (1, 1)\}$ . We will give a bijective proof for the identity

$$|A(n)| = 2^{n-1} |D(n)|$$

for  $n \geq 1$  as requested in Stanley's [3] Exercise 6.16. To give perspective for our proof, we will consider bijections between various sets of lattice paths with step sets which have path counts related to Narayana polynomials. The known results of the first two sections serve as background for the results of the last two sections.

Let  $A_i(n)$  denote the set of *all* lattice paths running from  $(0, -1)$  to  $(n, n)$  that use the steps in a specified step set, denoted by  $S_i$ , and that remain strictly above the horizontal line  $y = -1$  except initially. Let  $L_i(n)$  denote that subset of  $A_i(n)$  whose member paths remain strictly above the line  $y = x - 1$  except initially.

For the step set  $S_1 = \mathbb{P} \times \mathbb{P}$ , with  $\mathbb{P}$  denoting the positive integers, let  $A_1(n, k)$  ( $L_1(n, k)$ , resp.) denote the set of paths in  $A_1(n)$  ( $L_1(n)$ , resp.) with  $k$  steps.

**Proposition 1.** *For  $0 < k \leq n$ ,*

$$(1) \quad |A_1(n, k)| = \binom{n-1}{k-1} \binom{n}{k-1}$$

$$(2) \quad |L_1(n, k)| = \frac{1}{k} \binom{n-1}{k-1} \binom{n}{k-1} = \frac{1}{n} \binom{n}{k-1} \binom{n}{k}.$$

*Proof.* The right side of (1) simply counts the ways to assign the coordinates of the end points of the steps constituting a path in  $A_1(n, k)$ . Since exactly one of the  $k$  cyclic permutations of any path in  $A_1(n, k)$  lies in  $L_1(n, k)$ , the middle formula of (2) follows from (1).  $\square$

With  $H = (1, 0)$  and  $V = (0, 1)$  let  $S_2 = \{H, V\}$ . Observe that each step  $(u, v) \in S_1$  determines a vertical step  $(0, v)$ , followed by a horizontal step  $(u, 0)$ , and conversely. Hence there is an immediate matching between  $A_1(n, k)$  and  $A_2(n, k)$ , where  $A_2(n, k)$  is that subset of  $A_2(n)$  in which each path has  $k$  peaks (i.e.,  $VH$  pairs). Thus  $|A_1(n)| = |A_2(n)|$ . Likewise, there is an immediate matching between  $L_1(n, k)$  and  $L_2(n, k)$  where  $L_2(n, k)$  is that subset of  $L_2(n)$  in which each path has  $k$  peaks; thus  $|L_1(n)| = |L_2(n)|$ .

The count,  $|L_2(n, k)| = \frac{1}{n} \binom{n}{k-1} \binom{n}{k}$ , is known as a Narayana number, and

$$N_n(z) = \sum_{k=1}^n \frac{1}{n} \binom{n}{k-1} \binom{n}{k} z^k$$

is known as the  $n^{th}$ -Narayana polynomial. Studies of these polynomials are given in [1] and [4]. The sequence  $(N_0(1), N_1(1), N_2(1), \dots) = (1, 1, 2, 5, 14, \dots)$  form the Catalan numbers, while the sequence  $(N_0(2), N_1(2), N_2(2), \dots) = (1, 2, 6, 22, 90, \dots)$  gives the large Schröder numbers.

## 2. PEAKS, DOUBLE ASCENTS, AND LARGE SCHRÖDER PATHS

For positive integer  $z$ , replicate each path in  $L_2(n)$  by independently coloring its peaks with colors from a set of  $z$  colors where we require blue and red to be present whenever  $z \geq 2$ . Let  $L_2^c(n; z)$  denote the set of all such replicated paths. For example, the six paths of  $L_2^c(2; 2)$  are  $VVbHH$ ,  $VVrHH$ ,  $VVbHVbH$ ,  $VVbHVrH$ ,  $VVrHVbH$ , and  $VVrHVrH$ . From the previous section, we have

### Proposition 2.

$$|L_2^c(n; z)| = N_n(z).$$

On any path the intermediate vertex of a consecutive  $VV$  pair is called a *double ascent*. Let  $L_2^{cc}(n; z)$  denote the set of paths replicated from  $L_2(n)$  by independently coloring the intermediate vertex of each double ascent with the same  $z$  colors that were available for the peaks. We define a bijection

$$(3) \quad f : L_2^{cc}(n; z) \longrightarrow L_2^c(n; z)$$

as follows: Let  $R \in L_2^{cc}(n; z)$  be determined by the coordinates of its peaks, say,  $(x_1, y_1), \dots, (x_k, y_k)$ . (The coordinates of a peak are the coordinates of the intermediate vertex between the  $V$  and  $H$  steps.) Then  $(x'_1, y'_1), \dots, (x'_h, y'_h), \dots, (x'_{n+1-k}, y'_{n+1-k})$  will be the coordinates of the peaks of the path  $f(R)$  where

$$\begin{aligned} \{x'_1, \dots, x'_h, \dots, x'_{n+1-k}\} &= \{0, \dots, n\} - \{y_1, \dots, y_k\} \\ \{y'_1, \dots, y'_h, \dots, y'_{n+1-k}\} &= \{0, \dots, n\} - \{x_1, \dots, x_k\} \\ \text{with } x'_1 < x'_h < x'_{n+1-k} \quad \text{and} \quad y'_1 < y'_h < y'_{n+1-k} \end{aligned}$$

and the peak at  $(x'_h, y'_h)$  has the same color as the intermediate vertex of the double ascent on  $R$  with ordinate  $x'_h$ . Hence,

### Proposition 3.

$$|L_2^{cc}(n; z)| = N_n(z).$$

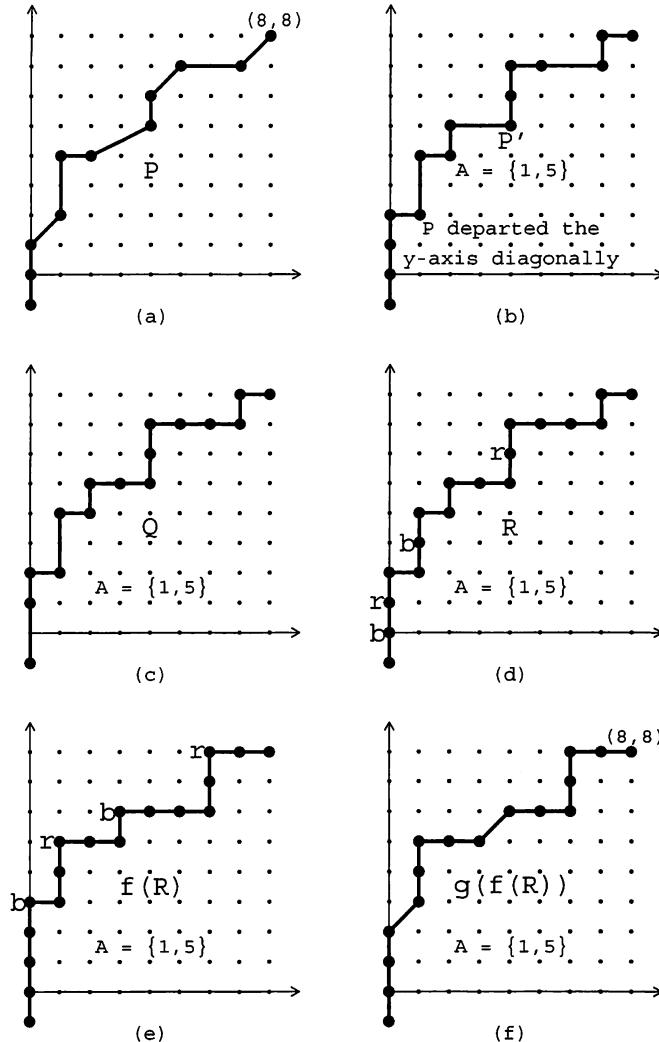


FIGURE 1. Various related paths

As an example, let  $R = VbVrVHVbVHVHHVrVHHHVH \in L_2^{cc}(8; 2)$  with peak set  $\{(0, 2), (1, 4), (2, 5), (4, 7), (7, 8)\}$ . The  $b$ 's and  $r$ 's represent the two colors,  $z = 2$ , for the double ascents. The image is  $f(R) = VVVVbHVrHHVbHHHVrHH \in L_2^c(8; 2)$  where now the colors  $b$  and  $r$  label peaks. We illustrate the bijection  $f$  for these paths in Figures 1d and 1e. (We use " $A = \{1, 5\}$ " in Section 4.)

For the step set,  $S_3 = \{V, H, D\}$ , where  $D = (1, 1)$ , the paths of  $L_3(n)$  are known as Schröder paths since  $|L_3(n)| = N_n(2)$ . (Usually in the literature the initial vertical step from  $(0, -1)$  to  $(0, 0)$  is omitted.) This last equation follows from the bijection

$$g : L_2^c(n; 2) \longrightarrow L_3(n)$$

where the image path is simply obtained by replacing each blue peak, i.e., blue VH pair, with the step  $D$  and by removing the color from each red peak. Figures 1e and 1f give an example of the map  $g$  with “ $A = \{1, 5\}$ ” ignored.

### 3. ARBITRARY VERTICAL AND HORIZONTAL STEPS

One can consider vertical and horizontal steps of varying lengths. Take  $S_4 = (\mathbb{P} \times \{0\}) \cup (\{0\} \times \mathbb{P})$ . Deutsch's [2] interest in counting  $L_4(n)$  motivated the following, which state without proof.

**Proposition 4.**

$$|L_4(n)| = N_n(4)/2.$$

### 4. ARBITRARY STEPS

For  $S_5 = \mathbb{N} \times \mathbb{N} - \{(0, 0)\}$  where  $\mathbb{N}$  denotes the non-negative integers, let  $A_5^0(n)$  ( $L_5^0(n)$ , resp.) denote the set of paths in  $A_5(n)$  ( $L_5(n)$ , resp.) that begin with a unit vertical step. Let  $A_3^0(n)$  denote the subset of those paths in  $A_3(n)$  that begin with a unit vertical step, where  $S_3 = \{V, H, D\}$ . (Without the initial vertical unit step,  $A_5^0(n)$  is essentially  $A(n)$  and  $A_3^0(n)$  is essentially  $D(n)$  of the Introduction.) The proof of (5) answers Exercise 6.16 in [3].

**Proposition 5.** For  $n \geq 1$ ,

$$(4) \quad |L_5^0(n)| = 2^{n-1} |L_3(n)| = 2^{n-1} N_n(2)$$

$$(5) \quad |A_5^0(n)| = 2^{n-1} |A_3^0(n)|.$$

*Proof of (4).* First we will obtain (4) by establishing a bijection

$$L_5^0(n) \longrightarrow L_3(n) \times 2^{[n-1]}$$

where  $2^{[n-1]}$  denotes the collection of subsets of  $\{1, 2, \dots, n-1\}$ . The intermediate bijections composing this bijection are illustrated in Figure 1.

We now define a bijection

$$(6) \quad h : L_5^0(n) \longrightarrow L_6(n) \times 2^{[n-1]}$$

where  $S_6 = (\{0\} \times \mathbb{P}) \cup \{H\}$ . Let  $P \in L_5^0(n)$  and let  $A$  be the set of abscissae of the initial points of the horizontal steps of  $P$  excepting any step that departs from the y-axis;  $A \subseteq \{1, \dots, n-1\}$ . Let  $P'$  be the path obtained from  $P$  by replacing each diagonal step  $(u, v)$  by the consecutive pair  $(0, v)(u, 0)$  and by leaving the vertical and horizontal steps of  $P$  unaltered. Let  $Q$  be obtained from  $P'$  by two operations: (1) Replace each horizontal step, say  $(u, 0)$ , by a string of  $u$   $H$  steps. (2) Whenever  $P$  departs the y-axis with a non-horizontal step, join the initial two steps of  $P'$  to create a new initial vertical step. Equivalently,  $Q$  will begin with a unit vertical step with end point at  $(0, 0)$  if, and only if,  $P$  leaves the y-axis with a horizontal step. See Figures 1a, 1b, and 1c. Define  $h(P)$  to be the pair,  $h(P) = (Q, A) \in L_6(n) \times 2^{[n-1]}$ .

Recalling the notation of Section 2, we define a bijection

$$(7) \quad K : L_6(n) \longrightarrow L_2^{cc}(n)$$

where  $Q \in L_6(n)$  is be matched with a path  $R \in L_2^{cc}(n)$  so that they trace the same points in the plane and each intermediate vertex of a double ascent on  $Q$  becomes a red double ascent of  $R$  and each lattice point that is interior to a vertical step of  $Q$  becomes a blue double ascent of  $R$ . With  $id$  denoting the identity map,

$$(g \times id) \circ (f \times id) \circ (K \times id) \circ h : L_5^0(n) \longrightarrow L_3(n) \times 2^{[n-1]}$$

is the desired bijection yielding (4).  $\square$

We remark that  $P$  departs from the  $y$ -axis with a diagonal step if, and only if, its image in  $L_3(n)$  has a diagonal step that precedes any peak.

*Proof of (5).* We modify the bijections proving (4) to prove (5). A *left turn* is intermediate point of a consecutive  $HV$  pair. Let  $A_2^\ell(n)$  ( $A_2^{cc}(n)$ , resp.) denote the set of replicated paths formed from  $A_2(n)$  so each left turn (double ascent, resp.) is independently colored blue or red. We have a bijection

$$F : A_2^{cc}(n) \longrightarrow A_2^\ell(n)$$

defined as follows: Let  $R \in A_2(n)$  be determined by the set (perhaps empty) of the coordinates of its left turns, namely  $\{(x_1, y_1), \dots, (x_k, y_k)\}$ . Then  $(x'_1, y'_1), \dots, (x'_h, y'_h), \dots, (x'_{n-k}, y'_{n-k})$  are the left turns of the path  $g(R)$  where

$$\begin{aligned} \{x'_1, \dots, x'_{n-k}\} &= \{1, \dots, n\} - \{x_1, \dots, x_k\} \\ \{y'_1, \dots, y'_{n-k}\} &= \{0, \dots, n-1\} - \{y_1, \dots, y_k\} \end{aligned}$$

with  $x'_1 < x'_2 < x'_{n-k}$  and  $y'_1 < y'_2 < y'_{n-k}$  and the left turn at  $(x'_h, y'_h)$  has the color blue if and only if  $y'_h$  is the ordinate of the intermediate vertex of a blue double ascent on  $R$ .

We have a bijection

$$G : A_2^\ell(n) \longrightarrow A_3^0(n)$$

where  $G(P)$  is obtained by replacing each blue  $HV$  pair by  $D$  and removing the color red. Finally, with the maps

$$h : A_5^0(n) \longrightarrow A_6(n) \times 2^{[n-1]} \quad \text{and} \quad K : A_6(n) \longrightarrow A_2^{cc}(n)$$

defined in the same ways as the maps are defined in (6) and (7), we have

$$(G \times id) \circ (F \times id) \circ (K \times id) \circ h : L_5^0(n) \longrightarrow L_3^0(n) \times 2^{[n-1]}$$

is a bijection yielding (5).  $\square$

## REFERENCES

- [1] J. Bonin, L. Shapiro, R. Simion, Some  $q$ -analogues of the Schröder numbers arising from combinatorial statistics on lattice paths, *J. Stat. Plann. and Infer.* 34 (1993) 35–55.
- [2] E. Deutsch, personal communication, 1999
- [3] R.P. Stanley, *Enumerative Combinatorics*, Vol II, Cambridge University, Cambridge, UK, Press, 1999.
- [4] R.A. Sulanke, The Narayana distribution, to appear in *J. Stat. Plann. and Infer.*

# A Fast Algorithm for Building the Hasse Diagram of a Galois Lattice

Petko Valtchev, Rokia Missaoui, Pierre Lebrun

Département d’Informatique, UQAM, C.P. 8888, succ. “Centre Ville”,  
Montréal, Québec, Canada, H3C 3P8

e-mail : {valtchev, missaoui, lebrun}@info.uqam.ca

## Abstract

Formal concept analysis and Galois lattices in general are increasingly used for large contexts that are automatically generated. As the size of the resulting datasets may grow considerably, it becomes essential to keep the algorithmic complexity of the analysis procedures as low as possible. This paper presents an efficient algorithm that computes the Hasse diagram of a Galois lattice from the lattice ground set, i.e., the set of all concepts. The algorithm performs an element-wise completion of the lattice according to a linear extension of the lattice order. This requires only a limited number of comparisons between concepts and therefore makes the global algorithm very efficient. In fact, its asymptotic time complexity is almost linear in the number of concepts. Consequently, the joint use of our algorithm with an efficient procedure for concept generation yields a complete procedure for building the Galois lattice.

## Résumé

L’analyse formelle de concepts et, plus généralement, la construction de treillis de Galois sont de plus en plus utilisées dans des circonstances où les données sont générées de manière automatique. La taille des jeux de données pouvant croître considérablement, il est important d’assurer une complexité algorithmique raisonnable pour les procédures d’analyse. Notre papier présente un algorithme efficace qui permet de construire le diagramme de Hasse d’un treillis de Galois en partant de l’ensemble des concepts (nœuds). La stratégie adoptée par l’algorithme est une complétion élément par élément du treillis basée sur une extension linéaire de l’ordre dans le treillis. Elle ne demande qu’un nombre très réduit de comparaisons entre concepts de sorte que l’algorithme entier reste très efficace. Ainsi, la complexité asymptotique de l’algorithme est quasiment linéaire selon le nombre de concepts dans le treillis. En conséquence, notre algorithme peut avantageusement compléter une procédure de génération de concepts au sein d’un module efficace d’analyse formelle.

## 1 Introduction

Concept analysis has proven to be a valuable tool for gaining insight into complex data [10, 13, 9]. In many applications of concept analysis, experts learn from formal contexts by inspecting their carefully layouted concept lattices. Contexts in these applications tend to have a modest size because otherwise the resulting concept lattices are hard to analyze visually. The algorithmic complexity of concept analysis for these

applications is consequently a minor concern. But concept analysis is used increasingly for applications like program analysis inside a compiler [16] or data analysis within a large database [11] where large contexts are constructed automatically. The resulting concept lattice is no longer inspected visually but is considered as a data structure that is part of the program. For these applications, the algorithmic complexity of concept analysis does matter. This paper presents an efficient algorithm for computing the concept lattice from its ground set and provides some analytical complexity results.

We first present the basics of the Galois lattices and formal analysis (Section 2). Then, we provide a row of structural properties of the Galois lattice that help devise an efficient procedure for its construction (Section 3). Next, an algorithm is sketched that implements the suggested strategy in a straightforward manner (Section 4). Finally, some improvements that make the algorithm computational cost decrease are described (Section 5).

## 2 Formal concept analysis and Galois lattices

*Formal concept analysis* (FCA) [9] is an approach towards the extraction of highly similar groups of objects from a collection  $O$  of objects described by a set of attributes  $A$ . The paradigm occurred within the lattice theory [2]: the attributes considered represent binary features, i.e., with only two possible values, *present* or *absent*. In this framework, the discovered groups represent the closed sets of the *Galois connection* [1] induced by  $I$  on the couple  $O$  and  $A$ .

### 2.1 Basics

The objects in  $O$  can be seen as binary vectors and the dataset as a binary table with an incidence relation  $I$  ( $oIa$  means that the object  $o$  has the attribute  $a$ ). The table  $(O, A, I)$ , further called *formal context* or simply context, may be processed to extract the natural groupings and relationships between objects and attributes. For that reason, two set-valued functions are defined:  $f$  extracts common attributes from a set of objects whereas  $g$  computes the dual for attribute sets. Formally, the functions are defined as follows:

- $f : \mathcal{P}(O) \rightarrow \mathcal{P}(A), f(X) = \{a \in A | \forall o \in X, oIa\}$
- $g : \mathcal{P}(A) \rightarrow \mathcal{P}(O), g(Y) = \{o \in O | \forall a \in Y, oIa\}$

Both functions constitute a Galois connection between  $\mathcal{P}(O)$  and  $\mathcal{P}(A)$ . The closed subsets of both  $O$  and  $A$  constitute two lattices with respect to the set inclusion. Both lattices are isomorphic, they can be merged into a common structure made up of couples  $(X, Y)$  of an object set  $X \in \mathcal{P}(O)$  and an attribute set  $Y \in \mathcal{P}(A)$ . The couples, called *concepts*, are complete couples with respect to  $I$ , in the sense that  $Y = f(X)$  and  $X = g(Y)$ . In the FCA framework,  $X$  is referred to as the concept *extent* and  $Y$  as the concept *intent*. The set  $\mathcal{C}_K$  of all concepts over a context  $K = (O, A, I)$  constitute a lattice  $\mathcal{L}$  with respect to  $\leq_{\mathcal{L}}$ , the product of both set inclusions,  $\subseteq_O$  and  $\supseteq_A$ :

- $(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2$ .
- $(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow Y_2 \subseteq Y_1$ .

The lattice  $\mathcal{L} = \langle \mathcal{C}, \leq_{\mathcal{L}} \rangle$  is called *Galois lattice* [1] or *formal concept lattice* [9]. Lattice operators *join* and *meet* provide the least upper bound (LUB) and the greatest lower bound (GLB) in the concept lattice respectively. They are defined as follows:

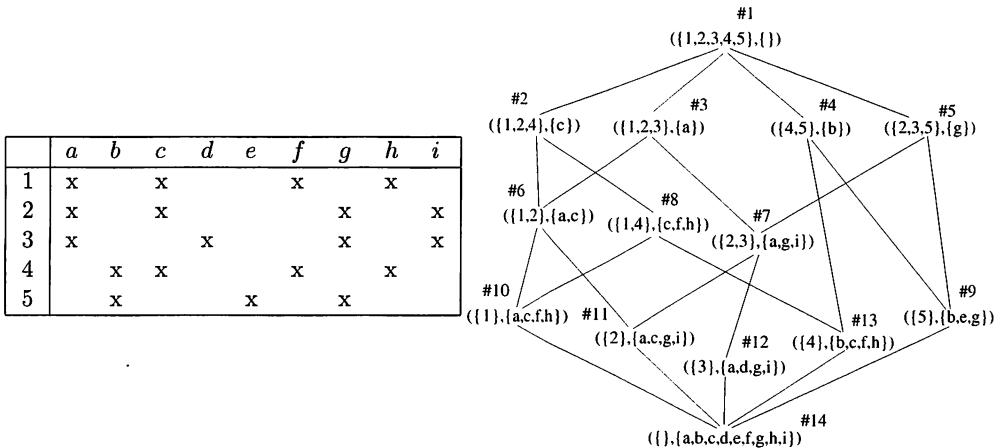


Figure 1: A binary relation and its Galois or concept lattice.

- $(X_1, Y_1) \vee (X_2, Y_2) = (g \circ f(X_1 \cup X_2), Y_1 \cap Y_2)$
- $(X_1, Y_1) \wedge (X_2, Y_2) = (X_1 \cap X_2, f \circ g(Y_1 \cup Y_2))$

For conciseness reasons, both basic operations will be denoted by a quote ('), and the closure operators by a double quote ("). Thus,  $X'$  and  $Y'$  will stand for  $f(X)$  and  $g(Y)$  respectively, whereas  $X''$  and  $Y''$  will stand for  $g \circ f(X)$  and  $f \circ g(Y)$ . These notations reflect the symmetric role of both initial operations.

Furthermore, given an object  $o$ , there is always a most specific concept whose extent contains  $o$ . Dually, for each attribute  $a$ , there is a most general concept whose intent includes  $a$ . These concepts will be further referred to as *object concept* and *attribute concept* respectively.

## 2.2 Computing the lattice nodes only

The concepts of a context may be seen as maximal rectangles in the table of the binary relation which represents the context. These are called *complete primary submatrices* of the global matrix. From a graph-theoretical viewpoint, the concepts represent complete maximal sub-graphs of the bipartite graph which is described by the context. The wide range of problems that may be reduced to the discovery of the closed sets of a Galois connection explains the variety of the existing algorithms. A detailed description of those algorithms is out of the scope of our paper. Interested readers may refer to [12] for a very comprehensive presentation. We only sketch the basic principles of those algorithms whereas a deeper, performance-oriented comparison may be found in [10].

The first algorithm that may be used to discover the concept set (see Chein [5]) comes from the linear algebra. It generates the concepts in an iterative manner, starting by the most specific ones, i.e., the object concepts. At each step, new concepts are generated as couples where the intent is the intersection of the intents of two already existing concepts. As there are numerous ways of obtaining a concept as the join of two other concepts, the algorithm performs a lot of redundant generations which consists a major drawback.

Another algorithm that only computes the concept set has been proposed by Nor-

ris [15]. The underlying principle is an incremental generation of all the concepts over the sub-tables made up of the first  $i$  rows, i.e., attributes. The algorithm iterates over  $i$ : it starts by the first row and generates, at each step, the possible "enhancements" of the already generated concepts with the entries of the  $i$ -th row.

The algorithm suggested by Ganter [7] is undoubtedly the most sophisticated one as it uses a deeper insight in the structure of the concepts to speed-up the computation. Actually, in order to avoid the most expensive part of the concept generation, i.e., the lookup for redundant generations, the algorithm uses a specific order on the concept sets called *lectic*. The concepts are generated according to the lectic order which is total and therefore, each concept is computed only once. The main drawback of the approach, as in the two previous cases, remains the lack of structure over the concept set.

## 2.3 Computing both the nodes and the Hasse diagram

The Bordat algorithm [3] generates both the concept set and the Hasse diagram of the lattice. The structural properties of the precedence relation between concepts are used to generate the concepts in the appropriate order. Thus, from each concept the algorithm generates its upper covers. The drawback is that a concept is generated a number of times that corresponds to the number of its lower covers (one concept generation per link in the Hasse diagram).

Finally, Godin *et al.* [10] suggested an incremental procedure that maintains lattice structure, i.e., both concept set and Hasse diagram, upon the insertion of a new object into the table. The principle of the algorithm consists of locally modifying the lattice structure (insertion of new concepts, completion of existing ones, deletion of redundant links, etc.) while keeping large parts of the lattice untouched. The procedure may be applied to both incremental and non-incremental concept lattice problems.

## 2.4 What is missing?

When asymptotic complexity is only considered, the Ganter algorithm shows the best score since it only generates every concept once. However, the algorithm requires an additional effort to build the diagram, a post-processing which may significantly increase the overall complexity. In fact, a naive algorithm that builds the precedence relation of a partially ordered set from its order relation, has a complexity which is cubic in the size of the ground set (the number of concepts here). This exceeds by far the complexity of the main concept building algorithms and, finally, makes all optimization efforts over these algorithms useless. In addition, the expensive diagram building prevents sensible performance-based comparisons between algorithms that compute only the concept set and those which build the diagram. In particular, it is impossible to correctly asset the incremental strategy versus the batch ones.

In a recent paper, Ganter *et al.* made a suggestion [8] about how the Hasse diagram may be efficiently extracted from the concept set. The basic idea is to search for minimal concepts among all the super-concepts of a given concept. The super-concepts are dynamically generated and compared. The idea has been further developed by Lindig [14] who presented an effective algorithm that computes both the concept set and the diagram.

We choose another direction based on a dynamic view of the lattice as a construct which is gradually completed by linking one concept at a time. The completion is done in a top-down manner, starting from the T node and processing the rest of the nodes according to a total order which is a linear extension of the lattice order. At each step,

the current element is connected to each of its immediate successors in the final lattice, further called upper covers. An important feature of such a procedure is that at the time a node is processed, all its upper covers are already completely integrated in the partial structure.

### 3 Basic facts about precedence relation in $\mathcal{L}$

In the following we use some basic notions of the lattice and partially ordered set theory that we shall not define here. As the vocabulary used is consistent with the excellent work of Davey and Priestley [6], the interested readers may refer to their book.

#### 3.1 Auxiliary definitions

In the following we define the border of a partially constructed lattice. First, the lattice construction is achieved through an element-wise insertion of the lattice nodes. The construction is supported by a structure that, once the algorithm has finished its work, contains the entire lattice. At any time point, the structure represents a partial order (poset) that is a sub-order of the lattice  $\mathcal{L}$ .

**Definition 3.1.** Let  $P_i$  denote the partial ordered set represented by the structure after the insertion of the  $i$ -th node. Clearly,  $P_0 = \langle \emptyset, \emptyset \rangle$ , the empty lattice, and  $P_l = \mathcal{L}$ , where  $l = ||\mathcal{C}||$ .

In the following, the poset  $P_i$  will not be distinguished from its ground set. Of course, the poset  $P_i$  depends on the particular (linear) order  $\leq^e$  in which the nodes are inserted. Our approach relies heavily on some interesting properties of  $P_i$  that follow from the choice of  $\leq^e$ . In fact, several facts concerning  $P_i$  may be observed. First, when the order  $\leq^e$  is a linear extension of the lattice order  $\leq_{\mathcal{L}}$ , i.e., both orders match on common couples (noted  $\leq_{\mathcal{L}} \in \leq^e$ ), then the set  $P_i$  is an *upper set* of  $\mathcal{L}$ .

**Property 3.1.** Whenever  $\leq_{\mathcal{L}} \in \leq^e$ ,  $\uparrow_{\mathcal{L}} P_i = P_i$  for all  $i$ .

In other words, with every element  $c$  already in  $P_i$ , the poset includes its upper bounds as well. Of course, we can assume the elements in  $P_i$  are correctly linked with respect to  $\leq_{\mathcal{L}}$ , so that  $P_i$  is the sub-order of  $\mathcal{L}$  induced by its elements (no links are missing). Moreover, the above property may be extended to  $c_{i+1}$  ( $i \leq l - 1$ ), i.e., the element in  $\mathcal{C}$  which is to be integrated into  $P_i$  on the next step of the construction.

**Property 3.2.** Whenever  $\leq_{\mathcal{L}} \in \leq^e$ ,  $\forall c \in \mathcal{C}, c_i \leq_{\mathcal{L}} c \text{ implies } c \in P_i$ , for all  $i \leq l - 1$ .

Our attention is drawn by the elements that cover  $c_i$  in  $\mathcal{L}$  since they have to be physically connected to  $c_i$  in order to achieve their integration into  $P_{i-1}$ . These elements will be further referred to as *upper covers*, as opposed to the *lower covers* which are the elements covered by  $c_i$ .

**Definition 3.2.** The set of upper covers of  $c_i$  in  $\mathcal{L}$  is  $c_i^{\nabla} = \{c | c_i \prec_{\mathcal{L}} c\}$ .

Clearly, if a linear extension of  $\leq_{\mathcal{L}}$  is used, the entire set is already in  $P_i$ . In the following, we assume  $\leq_{\mathcal{L}} \in \leq^e$  so that our aim will be to devise an algorithm that efficiently detects  $c_i^{\nabla}$  within  $P_{i-1}$ .

### 3.2 Where are the upper covers?

As we noted previously, all the upper covers of a  $c_i$  are already in  $P_{i-1}$ . With a similar reasoning, one may show that  $P_{i-1}$  contains no lower bound of  $c_i$  (due to the linear order). This means that  $c_i$  is a minimal element of  $P_i$ . One may expect, that with some tricky choice of  $\leq^e$ , all the upper covers of  $c_i$  remain minima of  $P_i$ , a fact that would greatly simplify their look-up. Unfortunately, a general definition of such an order  $\leq^e$  is impossible. However, the idea may be re-tailored in order to fit to the general case of non-minimal upper covers. For that reason, we first define the set of minima in  $P_i$  as the "border" of the current lattice construction.

**Definition 3.3.** *The border of  $P_i$  is the set of its minima:*

$$\perp(P_i) = \{c \in P_i \mid \forall c' \in P_i, c' \leq_{P_i} c \Rightarrow c' = c\}.$$

First, the border is a dynamic structure that changes with  $P_i$ , i.e., depends on  $\leq^e$ . Its evolution between two construction steps  $i$  and  $i + 1$  can be simply formulated: the new border always includes the new element,  $c_i$  whereas all elements of the old border that are greater than  $c_i$  are dropped out.

**Property 3.3.**

$$\perp(P_i) = \{c_i\} \cup (P_{i-1} - \uparrow_{P_{i-1}} c_i).$$

Next, the set constitutes an *anti-chain* in the entire lattice  $\mathcal{L}$ . Observe that the set  $c_i^\nabla$  is split between the border  $\perp(P_{i-1})$  and the inner part of  $P_{i-1}$ , i.e., the rest of the set. Moreover, all the elements to drop from the border upon joining  $c_i$  are, in fact, upper covers of  $c_i$ .

**Property 3.4.**

$$(P_{i-1} - \uparrow_{P_{i-1}} c_i) \subseteq c_i^\nabla.$$

As we have noticed earlier, the above set inclusion is strict in the general case. Thus, a simple test of comparability between  $c_i$  and the members of  $\perp(P_i)$  is not enough to fix  $c_i^\nabla$ : some upper covers will lay "higher" in  $P_{i-1}$ . The question is how to recognize these nodes that are "shaded" by elements in the border that are inferior to them.

### 3.3 Which are the shaded covers?

To devise a criterion for upper covers that are not border elements, we need some basic facts. First, for each shaded cover, say  $c_s$ , a border element  $c_b$  exists which is inferior to the cover,  $c_b \leq_{P_{i-1}} c_s$ . This follows trivially from the definition of the border. A less trivial fact is that the shaded cover is exactly the join of  $c_b$  and the current element  $c_i$ . The general proposition may be formulated as follows<sup>1</sup>:

**Property 3.5.**

$$\forall c_u \in c_i^\nabla, \exists c_b \in \perp(P_{i-1}), c_b \vee_{\mathcal{L}} c_i = c_u.$$

The above proposition does not yield a recognition criterion alone, but provides its basis. It says all upper covers could be found as joins of the new node and a border element. Thus, a possible search strategy would be to look for all those GLBs, an approach that would require a considerable amount of graph search. In addition, one

---

<sup>1</sup>Observe that in the degenerated case,  $c_u$  and  $c_b$  are the same.

has to check the minimality of each join since not all of them need to be upper covers (the condition is necessary but not sufficient).

Fortunately, we do not need to perform such an exhaustive search. At this point, we may apply a basic result from the Galois lattice theory that helps limit the search. Recall that the intent of a *join* of two concepts is the intersection of both intents.

**Property 3.6.** *Given two concepts  $(X_1, Y_1)$  and  $(X_2, Y_2)$  in  $\mathcal{L}$ , their LUB in  $\mathcal{L}$  is*

$$(X_1, Y_1) \vee_{\mathcal{L}} (X_2, Y_2) = ((Y_1 \cap Y_2)', Y_1 \cap Y_2).$$

Thus, a possible strategy for the discovery of upper covers could be to generate all the intents of those concepts and to use the intents to localize the nodes themselves within  $P_{i-1}$ . The generation is a simple intersection between the intents. Observe that the minimality constraint holds also on intents, therefore only intents that are “minimal” with respect to the lattice order  $\leq_{\mathcal{L}}$  are sought. This means, as  $\leq_{\mathcal{L}}$  follows the inverse of the inclusion between intents,  $\supseteq$ , we are looking for intents that are maximal for  $\subseteq$ . The total amount of knowledge about the  $P_{i-1}$  structure that is necessary for the correct integration of  $c_i$  is summarized in the following lemma.

**Lemma 3.1.** *For an arbitrary  $\leq_e$ , a linear extension of  $\leq_{\mathcal{L}}$ , and  $i \in [1..l]$ , let  $c_i = (X_i, Y_i)$  be the element of the  $i$ -th rank in  $\mathcal{C}$  and  $P_{i-1}$  the partial order induced by the first  $i - 1$  elements of  $\mathcal{C}$ . Let also the set  $D_r$  be the set of all intents of upper covers of  $c_i$  in  $\mathcal{L}$ ,  $D_r = \{Y_j | (X_j, Y_j) \in c_i^{\nabla}\}$ . Finally, let  $D_c$  be the set of all maxima of the set of intersections between  $Y_i$  and an intent of a border concept,  $D_c = \{Y_b | Y_b = Y_i \cap Y_j, (X_j, Y_j) \in \perp(P_{i-1})\}$ . Then the two sets are identical,  $D_r = D_c$ .*

Once the set of all upper cover intents has been extracted from the set of all generated intents, the task of finding the actual nodes in  $P_{i-1}$  could be done by a simple look-up. It is noteworthy that the real gain of using the intents instead of looking for nodes is that the comparability between concepts (nodes) can be established in a far faster way.

## 4 A straightforward implementation of the diagram building

From the remarks of the previous paragraph, a quite simple algorithm could be drawn.

### 4.1 The algorithm

The procedure has three steps: the first one generates the intents of all LUBs, the second finds all maximal sets among the intents and the third one first finds all nodes in  $P_i$  that correspond to maximal intents and then connects the current node to them. Of course, an additional effort is necessary to maintain the list of border elements which is independent from the upper cover computation.

The above algorithm follows strictly the reasoning we have presented previously. It first sorts the lattice nodes in  $\mathcal{C}$  in order to obtain a linear extension of  $\leq_{\mathcal{L}}$ . Then, the nodes are processed in a decreasing order, each time integrating the current node into the partial structure  $\mathcal{L}$  that finally contains the entire lattice. At each step, the intersections between the current concept intent and the intents of all border concepts are computed and stored in the *Intents* set. Then, the maximal elements of *Intents* are

```

1: procedure HASSE
2:
3: Input :  $\mathcal{C} = \{c_1, c_2, \dots, c_l\}$ 
4: Output :  $\mathcal{L} = \langle \mathcal{C}, \leq_{\mathcal{L}} \rangle$ 
5:
6: SORT( $\mathcal{C}$ )
7:  $\mathcal{L} \leftarrow \{c_1\}$ 
8:  $Border \leftarrow \{c_1\}$ 
9: for  $i$  from 2 to  $l$  do
10:    $Intents \leftarrow \emptyset$ 
11:   for all  $\bar{c} \in Border$  do
12:      $Intents \leftarrow Intents \cup (Int(\bar{c}) \cap Int(c_i))$ 
13:   Cover-Intents  $\leftarrow \text{MAXIMA}(Intents)$ 
14:   for all  $Y \in Cover-Intents$  do
15:      $\hat{c} \leftarrow \text{FIND-CONCEPT}(Y)$ 
16:      $\text{MAKE-LINK}(c_i, \hat{c})$ 
17:    $Border \leftarrow (Border - \text{upper-covers}(c_i)) \cup \{c_i\}$ 
18:    $\mathcal{L} \leftarrow \mathcal{L} \cup \{c_i\}$ 

```

**Algorithm 1:** Building the Hasse diagram of a lattice from its ground set

selected and put in *Cover-Intents* (the MAXIMA primitive). These intents correspond to the most specific concepts that are greater than  $c_i$ , i.e., its upper covers. Next, for each element of *Cover-Intents*, the corresponding concept is localized within the partial structure  $P_i$  (FIND-CONCEPT) and the respective link is created. Finally, the upper covers of a concept are excluded from the border of the partial structure and the current element is added.

## 4.2 An example

The following table gives a partial trace of the algorithm by showing some of the iterations on the concepts (previously sorted according to the size of their intent). For example, the processing of concept  $c_6$  generates as many intents as there are nodes in the current border. The corresponding upper covers are therefore  $\{a\}$  and  $\{c\}$ , which give rise to two links: one between  $c_6$  and  $c_2$  (the node having  $\{a\}$  as an intent), and another one between  $c_6$  and  $c_3$ . The border is then updated by inserting the current node  $c_6$  and deleting the parents of  $c_6$  (i.e.,  $c_2$  and  $c_3$ ).

## 4.3 Complexity issues

The complexity of this first algorithm may be roughly evaluated as follows. The sorting procedure may be done in  $O(l \log(l))$  where  $l = \|\mathcal{C}\|$ . The global loop (lines 9 to 18) on  $\mathcal{C}$  takes  $l$  steps. The intent generation loop (lines 11-12) is executed once for each border element. As the border is made up of mutually incomparable elements, its size is bound by  $\omega(\mathcal{L})$ , the size of the maximal anti-chain of  $\mathcal{L}$ , also called the width of the lattice. Moreover, a straightforward intersection (line 12) algorithm for sets takes time that is linear in the set size. As the intersection is between intents, the cost is bound by  $O(m)$  where  $m = \|A\|$ . The complete time complexity of the generation step is therefore  $O(\omega(\mathcal{L})m)$  algorithm. The maxima computation, when performed in a direct manner, may require up to  $\omega(\mathcal{L})^2$  comparisons of intents, thus reaching a total complexity of

$c_i$	Intents	Cover-Intents	Links	Border
$c_2$	$\{\emptyset, \emptyset\}$	$\{\emptyset\}$	$(c_1, c_2)$	$\{c_2\}$
$c_3$	$\{\emptyset, \emptyset\}$	$\{\emptyset\}$	$(c_1, c_3)$	$\{c_2, c_3\}$
$c_4$	$\{\emptyset, \emptyset\}$	$\{\emptyset\}$	$(c_1, c_4)$	$\{c_2, c_3, c_4\}$
$c_5$	$\{\emptyset, \emptyset\}$	$\{\emptyset\}$	$(c_1, c_5)$	$\{c_2, c_3, c_4, c_5\}$
$c_6$	$\{\{a\}, \{c\}, \emptyset, \emptyset\}$	$\{\{a\}, \{c\}\}$	$(c_2, c_6); (c_3, c_6)$	$\{c_4, c_5, c_6\}$
$c_7$	$\{\{a\}, \{g\}, \emptyset\}$	$\{\{a\}, \{g\}\}$	$(c_2, c_7); (c_5, c_7)$	$\{c_4, c_6, c_7\}$
$c_8$	$\{\{c\}, \emptyset, \emptyset\}$	$\{\{c\}\}$	$(c_3, c_8)$	$\{c_4, c_6, c_7, c_8\}$
$c_9$	$\{\{b\}, \{g\}, \emptyset, \emptyset\}$	$\{\{b\}, \{g\}\}$	$(c_4, c_9); (c_5, c_9)$	$\{c_6, c_7, c_8, c_9\}$
$c_{12}$	$\{\{a\}, \{g\}, \{a, g, i\}\}$	$\{\{a, g, i\}\}$	$(c_7, c_{12})$	$\{c_9, c_{10}, c_{11}, c_{12}\}$
$c_{13}$	$\{\{b\}, \{c, f, h\}, \{c\}, \emptyset\}$	$\{\{b\}, \{c, f, h\}\}$	$(c_4, c_{13}); (c_8, c_{13})$	$\{c_9, c_{10}, c_{11}, c_{12}, c_{13}\}$

Table 1: The trace of the execution of Algorithm 1 with the concepts of the context in Figure 1.

the minima computation of  $O(\omega(\mathcal{L})^2 m)$ . Next, the loop over all maximal intents (lines 14-16) is run once for each upper cover. The number of the covers is bound by another lattice characteristics,  $d(\mathcal{L})$ , the maximal degree (input or output) of a lattice node. Concept look-up could be supported by a search structure that ensures linear worst-case time, i.e.,  $O(l)$ . Thus, the cover localization takes  $O(d(\mathcal{L})lm)$  in time. Finally, the border update may take at worst  $O(d(\mathcal{L})\omega(\mathcal{L})m)$  in time since each combination of upper cover and border element may have to be tested.

In sum, the total complexity amounts to  $O(l(\log(l) + \omega(\mathcal{L})m(d(\mathcal{L}) + \omega(\mathcal{L})))$ . The most time-consuming task is the maxima computation as the degree  $d(\mathcal{L})$  is always bound by the lattice width  $\omega(\mathcal{L})$ . With the remark that  $\log(l)$  is for sure less than  $\omega(\mathcal{L})m$ , the above formula simplifies to  $O(l\omega(\mathcal{L})^2 m)$ .

## 5 An improved algorithm

In the previous algorithm, we made no substantial use of some *ad-hoc* information that could essentially speed-up the computations. For example, one may observe that the first sorting operation, SORT, may be performed in linear time. In a similar way, the operation of finding the maximal intersections, MAXIMA, could take the advantage of a preliminary sorting so that a large number of unnecessary comparisons could be dropped off. Finally, instead of utilizing an additional search-and-insert structure to store the already processed part of the concept set, one may simply use the partial lattice structure. For this purpose, it is enough to define search criteria that efficiently explore the available links between concepts.

In the following, we describe in more detail the above improvements and then provide the new version of the algorithm.

### 5.1 Sort of concept set

In fact, a simple criteria that yields a linear extension of the lattice order is the size of the intents. In fact, if a concept is considered greater than another one whenever its intent contains more attributes, the obtained order clearly includes the lattice order  $\leq \mathcal{L}$ . However, the concept intents are of bounded size: the larger one contains all the attributes, so the sizes are at most  $m$ . Therefore, the set of possible values to sort upon, or keys, is limited to  $\{0, 1, 2, \dots, m\}$ .

In general, the sort of an item set  $\mathcal{I}$  with a restricted key domain  $D_K$  may be performed efficiently by means of an array which is indexed by the key set (see below). Here, the objective is simply to split the set of concepts into bunches of equal intent sizes and then to concatenate the bunches to obtain the desired order. Actually, the order within a bunch is ignored, therefore, the first task will be to assign to each entry (e.g., 2) all the concepts that score to that value (e.g.,  $(\{1, 2\}, \{a, c\})$ ,  $(\{1, 4\}, \{c, f, h\})$ , etc.). To sort the whole item set, it will then be enough to sort the keys and then concatenate the bunches according to the resulting order. However, integer keys are naturally sorted, so no additional effort is to be provided. As splitting into bunches is linear in time, the entire sort operation is linear. The data structure required to implement the efficient sort is a simple array of sets.

```

1: SORT( $\mathcal{C} = \{c_1, c_2, \dots, c_l\}_i$ )
2:   Bunches : array [0..m] of sets
3:   Order : list of concepts
4:
5:   for  $i$  from 1 to m do
6:     Bunches[ $i$ ]  $\leftarrow \emptyset$ 
7:     Order  $\leftarrow \emptyset$ 
8:     for  $i$  from 1 to l do
9:        $k \leftarrow \|Int(c_i)\|$ 
10:      Bunches[ $k$ ]  $\leftarrow Bunches[k] \cup \{c_i\}$ 
11:      for  $i$  from 1 to m do
12:        for all  $c \in Bunches[i]$  do
13:          Order  $\leftarrow c \& Order$ 
```

**Algorithm 2:** Linear-time sorting of the concept set

The above algorithm produces a linear extension of the lattice order that is stored in the *Order* list. The algorithm is linear in time but also in space with respect to the number of concepts.

## 5.2 Order the intersections for comparison

The same linear-time sorting may be used to speed-up the search for maximal intents generated by intersections. The basic idea is that such a sorting will spare some redundant inclusion tests. The algorithm works as follows: first the intents are sorted, then, they are checked for minimalness. The main data structure used for inclusion checking is a list *Maxi* of the previously discovered minima.

Intuitively, a list of intents that are candidates for minima may be maintained easily. Provided the list is properly initialized, one may expect that a simple comparison of every intent  $Y$  with the elements of the list could be enough to validate or invalidate the  $Y$  minima status. Concerning the entry/exit discipline for the list, one may consider the following simple strategies. An intent  $Y$  enters *Maxi* whenever the list does not contain another intent  $Y'$  which is a super-set of  $Y$ . Conversely, each time a new intent  $Y$  is found that is a super-set of an intent  $Y'$  already in *Maxi*,  $Y'$  is dropped off the list.

As a matter of fact, the resulting simple algorithm does not improve the overall complexity of the **Maxima** task, as exposed in the previous section, since a square number of comparisons would be necessary in the worst case. However, when the intents are

processed in a decreasing order (with respect to a linear extension that may easily be established upon their cardinalities as described in the previous section), the complexity improves substantially. To see the point, observe that with the decreasing order, no exit operation is to be performed (as for any given concept, all its super-concepts have already been processed). This means that the list can not decrease during the checking of a particular element  $Y$ , it may only remain the same ( $Y$  is not a minimum) or grow ( $Y$  is an effective minimum). Therefore, the maximal size of the list equals the number of effective upper covers of the concept  $c_i$  (see the Algorithm 1 above) which is bounded by the number  $d$ .

```

1: MAXIMA( $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_r\}$ )
2:    $Maxi : \text{list of intents}$ 
3:
4: SORT( $\mathcal{Y}$ )
5:  $Maxi \leftarrow \emptyset$ 
6: for  $i$  from 1 to  $r$  do
7:    $is-min \leftarrow \text{true}$ 
8:   for all  $\hat{Y} \in Maxi$  do
9:      $is-min \leftarrow is-min \text{ and } (Y_i \not\subseteq \hat{Y})$ 
10:  if  $is-min$  then
11:     $Maxi \leftarrow Maxi \& Y_i$ 
12: return  $Maxi$ 
```

**Algorithm 3:** Efficient detection of the minima

Algorithm 3 has a time complexity which is quasi linear in the number of intents in the initial set  $\mathcal{Y}$ . In fact, the external loop (rows 6-11) will be executed exactly  $||\mathcal{Y}||$  number of times. For each intent, the inner loop will perform at most  $||Maxi||$  steps, a number which is bounded by  $d(\mathcal{L})$ . Finally, each inclusion test takes  $O(m)$  in time where  $m = ||A||$ .

### 5.3 Concept lookup through the partial lattice

Another time consuming task is the lookup of concepts based on their intents, FIND-CONCEPT of Algorithm 1. Usually, operations that require lookup of concepts upon their intents or extents are powered by sophisticated search structures. However, the operations of search and maintenance of such a structure (entry/exit) often represent efforts of different magnitudes. Here, we use the lattice structure itself to find concepts as it contains all the inter-concept links necessary for an efficient search. The basic idea is to start the search of a concept  $c$  of intent  $Y$  by a lower bound of that concept, say  $c'$ , and to follow an up-going path until  $c$  is found. For that purpose, with each intent  $Y$  in the list *Intents* of Algorithm 1 we keep a pointer that (physically) indicates the concept  $c'$  which helped in generating  $Y$  (by intersection with the intent of  $c_i$ ). In the Algorithm 4 below, the primitive *generator()* yields the concept that generated an intent. Actually, if the intent is taken as a pure set, then there are several such concepts in the general case. However, as we do not check uniqueness in the intent generation step, all the copies of the same intent are considered different and as such they will point to different concepts. This does not disturb our lookup as any of the pointed concepts lays in an up-going path that leads to the target concept. As an example, at the step 13 of the main loop in Algorithm 1 (see Table 4.2), the *Cover-Intents* set contains the sets

$\{b\}$  and  $\{c, f, h\}$ . The intent  $\{b\}$  has been generated by the intersection of the intents in  $c_{13}$  and  $c_9$ . However, it is a strict subset of the intent of  $c_9$ ,  $\{b, e, g\}$  whereas the concept whose intent is equal to  $\{b\}$ ,  $c_4$ , lays above  $c_9$ .

The lookup algorithm is simple, it moves upwards, one sub-concept link at a time. At any step, it checks, in a one-step look-ahead manner the inclusion of the intent  $Y$  in the intents of the concepts met. The auxiliary primitive *next* hereafter helps enumerate a collection of items by successfully providing each of its members in an arbitrary order.

```

1: FIND-CONCEPT( $Y$ )
2:
3:  $c \leftarrow \text{generator}(Y)$ 
4: while  $\text{Int}(c) \neq Y$  do
5:    $\hat{c} \leftarrow \text{next}(\text{upper-covers}(c))$ 
6:   while  $Y \not\subseteq \text{Int}(\hat{c})$  do
7:      $\hat{c} \leftarrow \text{next}(\text{upper-covers}(c))$ 
8:    $c \leftarrow \hat{c}$ 
9: return  $c$ 
```

**Algorithm 4:** Efficient lookup by intent.

The algorithm termination with the right concept can be proved trivially based on two facts. First, the intents of the concepts along the path that is walked through by the algorithm represent a strictly decreasing sequence of sets. Next, the intent  $Y$  is always a subset of the current intent.

The algorithm has a modest time complexity. Actually, a straightforward evaluation may lead to a result that is linear in the size of the lattice: the up-going path is bounded by the height of the lattice  $h$ , i.e., the size of a maximal chain. This value, usually much less than the lattice size, could nevertheless grow up to that size in the worst case (totally ordered lattice). Although, the number of concepts examined could be estimated by the number of elements in the set difference between the intent of  $\text{generator}(Y)$  and  $Y$ . Theoretically, this difference is bounded by  $m$ , yielding a total time complexity of  $O(m^2 d(\mathcal{L}))$  (as the inner loop executes at most  $d$  steps). However, in a realistic context the cardinality of the difference, and therefore the number of steps of the outer loop, would not exceed 4 or 5, since the chances for  $Y$  both being a maximum in *Intents* and having much smaller size than the intent of  $\text{generator}(Y)$  are close to zero.

## 5.4 Complexity issues

The efficient procedures presented above help improve the overall complexity of the diagram building algorithm. The following table summarizes the complexity of the different stages of the Algorithm 1 both in the straightforward and in the improved version.

Step	Straightforward algorithm	Improved algorithm
SORT	$O(l \log(l))$	$O(l)$
MAXIMA	$O(\omega(\mathcal{L})^2 m)$	$O(d(\mathcal{L})\omega(\mathcal{L})m)$
FIND-CONCEPT	$O(lm)$	$O(m^2 d(\mathcal{L}))$
1st inner <b>for</b>	$O(\omega(\mathcal{L})m)$	$O(\omega(\mathcal{L})m)$
2nd inner <b>for</b>	$O(d(\mathcal{L})lm)$	$O(md(\mathcal{L})^2)$
outer <b>for</b>	$O(lw(\mathcal{L})^2 m)$	$O(lmw(\mathcal{L})d(\mathcal{L}))$
HASSE	$O(lw(\mathcal{L})^2 m)$	$O(lmw(\mathcal{L})d(\mathcal{L}))$

As the above table suggests, the overall time complexity of the algorithm is  $O(lm\omega(\mathcal{L})d(\mathcal{L}))$ . Moreover, the factor  $m$  could be further dropped off the formula under some circumstances. In fact, whenever the number of the attributes remains reasonable (at most several hundreds), the comparisons of concepts could be substituted, through an appropriate encoding, by binary vector operations. These operations take constant time, so that the time complexity reduces to  $O(l\omega(\mathcal{L})d(\mathcal{L}))$ . Concerning space, the algorithm has linear complexity in the number of concepts. In fact, the number of data items relevant to a concept is bounded by the number of the auxiliary structures used, which is constant.

Our complexity results relate to the work reported by Ganter et al. [8] and Lindig [14]. The algorithm suggested by Ganter *et al.* has a time complexity of  $O(lg^2m)$  where  $g$  is the cardinality of the object set  $O$ ,  $g = \|O\|$ . Lindig in turn proposed a complete algorithm for the lattice building task of the same asymptotic time complexity. It is difficult to compare the algorithms with respect to their analytical complexity as the respective functions depend on different lattice parameters. However, one may hypothesize that our algorithm will perform better in the case of a sparse context, i.e., tables with few attributes per object. In fact, with such contexts, the width of the Galois lattice is a linear function of the number of objects/attributes, so the asymptotic complexity will reduce to  $O(lm(m + g)d(\mathcal{L}))$  which is essentially linear in lattice size and quadratic in the context size ( $mg$ ).

## 6 Conclusions and further research

We presented an algorithm that computes the Hasse diagram of a Galois (concept) lattice given the set of the closed sets (concepts). It relies upon a strategy of gradual insertion of the concepts into the partially built diagram, in a top-down manner. At each step, the portion of the diagram already constructed is maximally reused to speed-up the current insertion.

The algorithm represents a complement of the well-known algorithms of Noris, Chein and Ganter that compute the set of closed sets of a binary relation. In that sense, our algorithm, when it is jointly applied with a concept computing algorithm, is an alternative to the batch procedures of Bordat and Lindig as well as to the incremental procedure of Godin, which constructs both the ground set and the Hasse diagram.

Our algorithm shows asymptotic time complexity that is linear in the number of the concepts, the number of the attributes and the width of the lattice. In case of a sparse context, this result is significantly better than the results of the already existing techniques.

Much of the work on the algorithm is still to come. On the one hand, detailed experiments will be necessary in order to evaluate the practical performances of the method. For the time being, it is implemented in C++ and an initial package of tests are carried out. The testbed, made up of randomly generated data contexts, is aimed at evaluating the practical complexity with respect to various lattice measures (width, height, number of objects and/or attributes, number of couples in the incidence relation, etc.). The initial results suggest very good performances which are to be explained by the efficient detection of the upper-covers by a simple intent intersection (instead of closure computation as suggested by previous work). On the other hand, it will be interesting to find some estimators of the algorithm's key complexity factor, the lattice width, which is not measurable directly from the context. We believe some combinatorial results for digraphs may be successfully adapted to the special case of concept lattices.

Alternatively, the appropriate use of some advanced data structures may help improve the effective complexity score of the algorithm. For example, the use of Ordered Binary-Decision Diagrams [4] for the concept lookup seems to be a promising research track.

## References

- [1] M. Barbut and B. Monjardet. *Ordre et Classification: Algèbre et combinatoire*. Hachette, 1970.
- [2] B. Birkhoff. *Lattice Theory*, volume 25. American Mathematical Society Colloquium Publ., Providence, revised edition, 1973.
- [3] J.-P. Bordat. Calcul pratique du treillis de Galois d'une correspondance. *Mathématiques et Sciences Humaines*, 96:31–47, 1986.
- [4] R. Bryant. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.
- [5] M. Chein. Algorithme de recherche des sous-matrices premières d'une matrice. *Bull. Math. de la soc. Sci. de la R.S. de Roumanie*, 13, 1969.
- [6] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, 1992.
- [7] B. Ganter. Two basic algorithms in concept analysis (preprint). Technical Report 831, Technische Hochschule, Darmstadt, 1984.
- [8] B. Ganter and S. Kuznetsov. Stepwise Construction of the Dedekind-McNeille Completion. In *Proceedings of the 6th ICCS*, Montpellier, 1998.
- [9] B. Ganter and R. Wille. *Formal Concept Analysis, Mathematical Foundations*. Springer-Verlag, 1999.
- [10] R. Godin, G. W. Mineau, and R. Missaoui. Incremental structuring of knowledge bases. In *Proceedings of the first International Symposium on Knowledge Retrieval, Use and Storage for Efficiency, Santa Cruz, CA, USA*, pages 179–193, 1995.
- [11] R. Godin and R. Missaoui. An Incremental Concept Formation Approach for Learning from Databases. *Theoretical Computer Science*, 133:378–419, 1994.
- [12] A. Guénoche. Construction du treillis de galois d'une relation binaire. *Mathématiques et Sciences Humaines*, 109:41–53, 1990.
- [13] R. Kent. Rough concept analysis: A synthesis of rough sets and formal concept analysis. *Fundamenta Informaticae*, 27:169–181, 1996.
- [14] C. Lindig. Fast Concept Analysis. In *Proceedings of the 8th ICCS*, Darmstadt, 2000.
- [15] E. M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2):243–250, 1978.
- [16] G. Snelting and F. Tip. Reengineering class hierarchies using concept analysis. In *Proceedings of ACM SIGPLAN/SIGSOFT Symposium on Foundations of Software Engineering*, pages 99–110, Orlando, FL, 1998.

# Generalized ribbons and Macdonald symmetric functions

Mike Zabrocki

**ABSTRACT.** Given a partition  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ , let  $\lambda^{rc} = (\lambda_2 - 1, \lambda_3 - 1, \dots, \lambda_k - 1)$ . It is easily seen that the diagram  $\lambda/\lambda^{rc}$  is connected and has no  $2 \times 2$  subdiagrams which we shall refer to as a ribbon. To each ribbon  $R$ , we associate a symmetric function operator  $S^R$  which has a very combinatorial action on the Schur basis. We may define the major index of a ribbon  $maj(R)$  to be the major index of any permutation that fits the ribbon. We demonstrate an operator  $H_{1k}^t = \sum_R t^{maj(R)} S^R$  which adds a column to the partition indexing a Hall-Littlewood symmetric function. Next, we generalize the notion of a ribbon operator so that the thickness of the ribbon can be greater than 1 and state and demonstrate a formula for a recurrence that adds a column to the partition indexing a Macdonald symmetric function.

**RÉSUMÉ.** Soit  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$  un partage, et  $\lambda^{rc} = (\lambda_2 - 1, \lambda_3 - 1, \dots, \lambda_k - 1)$  son intérieur. On voit facilement que le diagramme  $\lambda/\lambda^{rc}$  est connexe et n'a pas de sous-diagramme de la forme  $2 \times 2$ . Nous appelons ces partages gauches des rubans. Pour chaque ruban, nous associons un opérateur de fonctions symétriques  $S^R$  qui a une action combinatoire sur la base de Schur. On définit l'index majeur d'un ruban  $maj(R)$  comme l'index majeur d'une permutation qui s'adapte à la forme du ruban. Nous définissons un opérateur  $H_{1k}^t = \sum_R t^{maj(R)} S^R$  qui ajoute une colonne au partage qui étiquette une fonction symétrique de Hall-Littlewood. Ensuite, nous généralisons la notion d'un opérateur de ruban tel que l'épaisseur d'un ruban peut être plus grande que 1. Nous présentons une formule pour une récurrence qui ajoute une colonne au partage qui étiquette une fonction symétrique de Macdonald.

## 1. RIBBONS AND RIBBON OPERATORS

The Schur functions indexed by a sequence of integers  $(p_1, p_2, \dots, p_k)$  can be defined by the Jacobi-Trudi identity  $s_{(p_1, p_2, \dots, p_k)} = \det |h_{p_i+i-j}|_{1 \leq i, j \leq k}$ . It is well known and easy to show that we have the relation

$$s_{(p_1, \dots, p_i, p_{i+1}, \dots, p_k)} = -s_{(p_1, \dots, p_{i+1}-1, p_i+1, \dots, p_k)}. \quad (1.1)$$

Let us recall that for a given symmetric function  $f$  it is customary to denote by  $f^\perp$  the operator that is dual to multiplication by  $f$  with respect to the Hall inner product. We shall make crucial use here of the Bernstein [7] operator

$$S_m = \sum_{k \geq 0} (-1)^k h_{m+k} e_k^\perp \quad (1.2)$$

Its action on the Schur basis may be easily computed with the formula  $S_m s_{(p_1, p_2, \dots, p_k)} = s_{(m, p_1, p_2, \dots, p_k)}$  and relation (1.1).

For a partition  $\lambda = (\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m > 0)$  we set  $\ell(\lambda) = m$ . Partitions here are drawn by the French convention with the smallest part on the top. With this operator we have a “Rodriguez” formula for Schur functions indexed by a partition  $\lambda$ ,

$$S_{\lambda_1} S_{\lambda_2} \cdots S_{\lambda_k} 1 = s_\lambda \quad (1.3)$$

We shall also use a modification of this operator  $\tilde{S}_m = \omega S_m \omega$ , which by virtue of the identity  $\omega s_\lambda = s_{\lambda'}$  adds a column on the Schur symmetric function.

A ribbon is a connected skew partition that contains no  $2 \times 2$  blocks. If  $\lambda$  is a partition of length  $k$ , we set  $\lambda^{rc} = (\lambda_2 - 1, \lambda_3 - 1, \dots, \lambda_k - 1)$  (the partition with the first row and column removed). It is easy to see that every ribbon partition will be  $\lambda/\lambda^{rc}$  for some partition  $\lambda$ .

Label the cells in a ribbon diagram with the numbers  $\{1, 2, \dots, k-1\}$  from left to right, top to bottom. This done, we let

$$D(R) = \{i \in [1, k-1] : i + 1^{\text{st}} \text{ cell of } R \text{ lies below the } i^{\text{th}} \text{ cell}\} \quad (1.4)$$

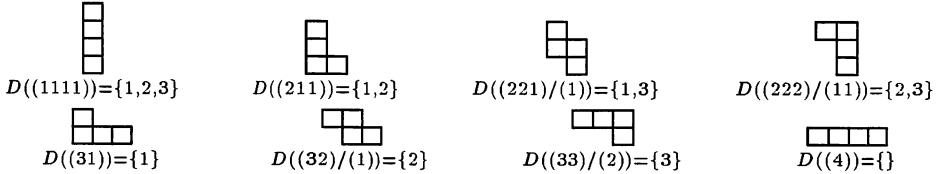
and refer to it as the descent set of  $R$ . The ribbons are therefore in one to one correspondence with the subsets of  $\{1, 2, \dots, k-1\}$ . If  $R$  is a ribbon of size  $k$ , we use  $\overline{R}$  to denote the ribbon whose descent set is  $\{1, 2, \dots, k-1\} - D(R)$ .

We also let the ‘major’ index of  $R$  be defined as

$$\text{maj}(R) = \sum_{i \in D(R)} i \quad (1.5)$$

and we set  $\text{comaj}(R) = \binom{k}{2} - \text{maj}(R)$  to be the complementary statistic. Clearly we have the relation  $\text{maj}(\overline{R}) = \text{comaj}(R)$ .

Below we have listed all of the ribbon partitions of size 4 with the corresponding descent set which is a subset of  $\{1, 2, 3\}$ .



We will use the symbol  $R$  to represent an arbitrary ribbon and the notation  $R \models k$  to indicate that  $R$  is a ribbon of size  $k$ .

For each ribbon  $R$  we create an operator that raises the degree of the symmetric function by the size of  $R$ . If  $R = \lambda/\lambda^{rc}$  then set

$$S^R = s_{\lambda^{rc}}^\perp \tilde{S}_{\lambda'_1} \tilde{S}_{\lambda'_2} \cdots \tilde{S}_{\lambda'_{\lambda_1}}. \quad (1.6)$$

where  $\lambda'_i$  is the length of the  $i^{\text{th}}$  column in the partition  $\lambda$ .

The action of this operator is very combinatorial in nature, we attach a ribbon on the left of the Schur function and reduce using the commutation relations  $\tilde{S}_a \tilde{S}_b = -\tilde{S}_{b-1} \tilde{S}_{a+1}$  and  $\tilde{S}_a \tilde{S}_{a+1} = 0$ , followed by the Littlewood-Richardson rule. We present one large example below.

**Example 1.1.**

$$S^{(432)/(21)}(s_{33221}) = -s_{665} - s_{764} - s_{755}$$

## 2. A HALL-LITTLEWOOD COLUMN ADDER

Define the symmetric functions  $H_\lambda[X; t] = \sum_\mu K_{\lambda\mu}(t) s_\mu[X]$  where  $K_{\lambda\mu}(t)$  is the Kostka-Folkes polynomial. There is an operator  $H_m^t$  that adds a row to this symmetric function when  $m$  is larger than  $\lambda_1$  that is due to Jing [3]. In particular, it yields the ‘Rodriguez’ formula

$$H_{\lambda_1}^t H_{\lambda_2}^t \cdots H_{\lambda_k}^t 1 = H_\lambda[X; t]. \quad (2.1)$$

One main result is the construction of an operator  $H_{1^k}^t$  which adds a *column* to the partition indexing  $H_\lambda[X; t]$ . More precisely, we demonstrate here the operator

$$H_{1^k}^t = \sum_{R \models k} t^{\text{comaj}(R)} S^R \quad (2.2)$$

has the following remarkable properties.

**Theorem 2.1.** *For all  $k \geq 0$ ,*

$$H_{m+1}^t H_{1^k}^t = H_{1^{k+1}}^t H_m^t. \quad (2.3)$$

*As a result, we have for  $\ell = \ell(\lambda)$*

$$H_{1^{\lambda_1}}^t H_{1^{\lambda_2}}^t \cdots H_{1^{\lambda_\ell}}^t 1 = H_{\lambda'}[X; t] \quad (2.4)$$

*where  $\lambda'$  is the conjugate partition to  $\lambda$ .*

The property that  $H_{1^k}^t$  adds a column to the Hall-Littlewood symmetric functions is a consequence of the commutation relation, since

$$H_{1^k}^t H_\lambda[X; t] = H_{\lambda_1+1}^t H_{\lambda_2+1}^t \cdots H_{\lambda_k+1}^t H_{1^0}^t(1) = H_{(\lambda_1+1, \lambda_2+1, \dots, \lambda_k+1)}[X; t]. \quad (2.5)$$

This result is the end product of a number of very interesting identities satisfied by ribbon operators. Our basic tool in establishing them is a truly remarkable new involution in the theory of symmetric functions.

We should mention that (2.2) is a rather surprising extension to the general case of the classical identity

$$H_{1^k}[X; t] = \sum_{\sigma \in S_k} t^{comaj(\sigma)} s_{\lambda(\sigma)}[X] \quad (2.6)$$

where the sum is over the symmetric group  $S_k$  and  $\lambda(\sigma)$  denotes the shape of the standard tableaux corresponding to  $\sigma$  under Robinson-Schensted correspondence. In fact, by grouping terms according to descent sets we derive from (2.6) that

$$H_{1^k}[X; t] = \sum_{R \models k} t^{comaj(R)} s_R[X]. \quad (2.7)$$

We present below an example of this operator and demonstrate the combinatorial nature of the operation of adding a column on the Hall-Littlewood symmetric functions.

**Example 2.2.** We see that for  $k = 3$ , formula (2.2) reduces to

$$H_{111}[X; t] = \begin{array}{|c|} \hline \square \\ \hline \end{array} + t \begin{array}{|cc|} \hline \square & \square \\ \hline \end{array} + t^2 \begin{array}{|ccc|} \hline \square & \square & \square \\ \hline \end{array} + t^3 \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} \quad (2.8)$$

To compute the Hall-Littlewood symmetric function  $H_{222}[X; t]$  we act on the symmetric function  $H_{111}[X; t] = s_{111} + (t + t^2)s_{21} + t^3s_3$  with each ribbon of size 3.

$$\begin{array}{|c|} \hline \square \\ \hline \end{array} H_{111}[X; t] = \begin{array}{|ccc|} \hline \square & \square & \square \\ \hline \end{array} + (t + t^2) \begin{array}{|ccccc|} \hline \square & \square & \square & \square & \square \\ \hline \end{array} + t^3 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \quad (2.9)$$

$$\begin{aligned} t \begin{array}{|cc|} \hline \square & \square \\ \hline \end{array} H_{111}[X; t] &= t \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} + (t^2 + t^3) \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} + t^4 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \\ &= (t^2 + t^3) \left( \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} + \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} \right) + t^4 \left( \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} + \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \right) \end{aligned} \quad (2.10)$$

$$\begin{aligned} t^2 \begin{array}{|c|} \hline \square \\ \hline \end{array} H_{111}[X; t] &= t^2 \begin{array}{|ccccc|} \hline \square & \square & \square & \square & \square \\ \hline \end{array} + (t^3 + t^4) \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} + t^5 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \\ &= -t^2 \begin{array}{|cccc|} \hline \square & \square & \square & \square \\ \hline \end{array} + t^5 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \end{aligned} \quad (2.11)$$

$$\begin{aligned} t^3 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} H_{111}[X; t] &= t^3 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} + (t^4 + t^5) \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} + t^6 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \\ &= t^6 \begin{array}{|cccccc|} \hline \square & \square & \square & \square & \square & \square \\ \hline \end{array} \end{aligned} \quad (2.12)$$

We have therefore computed that  $H_{222}[X; t] = s_{222} + (t + t^2)s_{321} + t^3s_{33} + t^3s_{411} + (t^2 + t^3 + t^4)s_{42} + (t^4 + t^5)s_{51} + t^6s_6$ .

### 3. A GENERALIZATION OF RIBBON OPERATORS

Define a notion of a generalized ribbon operator that starts with a ribbon  $R \models m$  with  $R = \lambda/\lambda^{rc}$  and associate with this a sequence  $v = (v_1, v_2, \dots, v_m)$  with  $v_i \geq 0$ . We generalize the notion of a ribbon by setting the ‘thickness’ of the  $i^{\text{th}}$  cell of the ribbon to be  $v_i + 1$  so that when the sequence consists of  $m$  zeros we have a standard ribbon.

Say that  $D(R) = \{i_1 < i_2 < \dots < i_{\ell(\lambda)-1}\}$  and  $\{1, \dots, m\} - D(R) = \{j_1 < j_2 < \dots < j_{\ell(\lambda)}\}$ . Let  $\alpha = (\lambda_2 - v_{i_1+1} - 1, \lambda_3 - v_{i_2+1} - 1, \dots, \lambda_{\ell(\lambda)} - v_{i_{\ell(\lambda)-1}+1} - 1)$  and  $\beta' = (\lambda'_1 + v_1, \lambda'_2 + v_{j_1+1}, \lambda'_3 + v_{j_2+1}, \dots, \lambda'_{\ell(\lambda)} + v_{j_{\ell(\lambda)-1}+1})$ . Then set  $S^{(R,v)} = (-1)^{|\lambda^{rc}| - |\alpha|} s_{\alpha}^{\perp} S_{\beta'_1} S_{\beta'_2} \cdots S_{\beta'_{\ell(\lambda)}}$ . We will call this a generalized ribbon operator.

The formulation of these operators leads to a simple construction with a picture, draw the original ribbon and place  $v_i$  cells either to the left or above the  $i^{\text{th}}$  cell in the ribbon depending on if  $i - 1$  is a descent of the ribbon.  $\alpha$  is the sequence representing the space underneath the diagram and  $\beta'$  is the sequence representing the heights of the columns of the diagram. We present a couple of examples to give a better picture of these truly combinatorial constructions.

**Example 3.1.** Consider the ribbon  $R = \begin{array}{c} \square \\ \square \end{array}$  of size 4 with  $D(R) = \{2, 3\}$ . If  $v = (0, 0, 0, 0)$  of course we have the original ribbon operator. If  $v$  is one of  $(1, 0, 0, 0)$ ,  $(0, 1, 0, 0)$ ,  $(0, 0, 1, 0)$ ,  $(0, 0, 0, 1)$  then  $S^{(R,v)}$  is equivalent to the following ribbon operators (respectively)

$$\begin{array}{cccc} \begin{array}{c} \square \\ \square \\ \square \end{array} = s_{11}^{\perp} \tilde{S}_4 \tilde{S}_3 & \begin{array}{c} \square \\ \square \\ \square \end{array} = s_{11}^{\perp} \tilde{S}_3 \tilde{S}_4 & - \begin{array}{c} \square \\ \square \\ \square \end{array} = s_1^{\perp} \tilde{S}_3 \tilde{S}_3 & - \begin{array}{c} \square \\ \square \\ \square \end{array} = s_{01}^{\perp} \tilde{S}_3 \tilde{S}_3 \end{array}$$

The second and fourth generalized ribbons are 0. The second because it contains the operation of adding a row of size  $d - 1$  on a row of size  $d$ , and the fourth for the same reason, but in this case a row of size 0 is added on a row of size 1 in the skew part of the operator.



**Example 3.2.** Let  $R = \begin{array}{cc} \square & \square \\ \square & \square \end{array}$ . Now let  $v = (1, 1, 0, 1, 0, 2, 0, 1, 0)$ . We represent this by the following picture where we have placed a dot in each of the cells representing the original ribbon and there are  $v_i$  cells either to the left or above the  $i^{\text{th}}$  depending on if  $i - 1 \in D(R)$  (and the  $v_1$  cells always go above the first cell in the ribbon).

$$\begin{array}{c} \begin{array}{ccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{array} \\ = s_{4321}^{\perp} \tilde{S}_6 \tilde{S}_6 \tilde{S}_5 \tilde{S}_5 \tilde{S}_3 \end{array}$$

Representing these operators with a diagram of this sort works fine if  $i - 1$  is a descent and  $v_i$  is so large that it creates a negative index in  $\alpha$ . We just interpret this to mean that skewing by a Schur function with a negative index kills the term and the result is 0.

Note also that some ‘straightening’ using relation (1.1) may be necessary.

For example  $R$  is as above, but  $v = (1, 1, 0, 1, 0, 4, 0, 1, 5)$ . Then  $R$  can be represented by the image as one would expect

$$\begin{array}{c} \text{Diagram of } R \\ \text{A 9x9 grid with shaded cells at positions (1,1), (1,2), (2,1), (2,2), (3,1), (3,2), (4,1), (4,2), (5,1), (5,2), (6,1), (6,2), (7,1), (7,2), (8,1), (8,2), (9,1), (9,2).} \end{array} = -s_{-1,3,2,1}^\perp \tilde{S}_6 \tilde{S}_6 \tilde{S}_5 \tilde{S}_7 \tilde{S}_3 = s_{-1,3,2,1}^\perp \tilde{S}_6 \tilde{S}_6 \tilde{S}_6 \tilde{S}_6 \tilde{S}_3 = s_{2,1,1,1}^\perp \tilde{S}_6 \tilde{S}_6 \tilde{S}_5 \tilde{S}_5 \tilde{S}_3 = \begin{array}{c} \text{Diagram of the result} \\ \text{A 9x9 grid with shaded cells at positions (1,1), (1,2), (2,1), (2,2), (3,1), (3,2), (4,1), (4,2), (5,1), (5,2), (6,1), (6,2), (7,1), (7,2), (8,1), (8,2), (9,1), (9,2).} \end{array}$$

The final image comes from first straightening the columns of the generalized ribbon and then straightening Schur function that one skews by with appropriate sign changes.

#### 4. A MACDONALD COLUMN ADDER

The Macdonald symmetric functions are a generalization of the Hall-Littlewood symmetric functions. The basis  $P_\mu[X; q, t]$  is defined by the following two conditions.

1.  $P_\mu[X; q, t] = m_\mu[X] + \sum_{\lambda < \mu} c_{\lambda\mu}(q, t) m_\lambda[X]$  for suitable coefficients  $c_{\lambda\mu}(q, t)$
2. The family  $\{P_\mu[X; q, t]\}_\mu$  is orthogonal with respect to the scalar product defined on the power symmetric functions by  $\langle p_\lambda, p_\lambda \rangle_{q,t} = \prod_{i=1}^{\ell(\lambda)} \frac{1-q^{\lambda_i}}{1-t^{\lambda_i}}$  and  $\langle p_\lambda, p_\mu \rangle_{q,t} = 0$  if  $\lambda \neq \mu$ .

Let  $a_\mu(s)$  represent the arm of cell  $s$  in the partition  $\mu$  and  $l_\mu(s)$  its leg, so that if  $s$  lies in the  $i^{th}$  row and the  $j^{th}$  column, then  $a_\mu(s) = \mu_i - j$  and  $l_\mu(s) = \mu'_j - i$ . The  $q, t$ -Kostka coefficients  $K_{\lambda\mu}(q, t)$  are then defined by the expansion of the following symmetric function in the dual-Schur basis  $S_\lambda[X; t]$ .

$$\prod_{s \in \mu} (1 - q^{a_\mu(s)} t^{l_\mu(s)+1}) P_\mu[X; q, t] = \sum_{\lambda \vdash |\mu|} K_{\lambda\mu}(q, t) S_\lambda[X; t] \quad (4.1)$$

We then set  $H_\mu[X; q, t] = \sum_{\lambda \vdash |\mu|} K_{\lambda\mu}(q, t) s_\lambda[X]$  as the family of symmetric functions that will interest us and we refer to this basis as the Macdonald symmetric functions. This family specializes to the Hall-Littlewood symmetric functions (discussed in the first two sections) by setting  $q = 0$  and they have the properties  $H_\mu[X; t, q] = \omega H_{\mu'}[X; q, t]$  and  $H_\mu[X; q, t] = t^{n(\mu)} q^{n(\mu')} \omega H_\mu[X; \frac{1}{q}, \frac{1}{t}]$  where  $n(\mu) = \sum_i (i-1)\mu_i$ .

A simple and elegant observation about the Macdonald polynomials allows us to recover a column adding operator given that we know what the operator is when  $q = 0$ . In particular this observation provides us with the following formula.

**Theorem 4.1.** *An operator that adds a column to the partition indexing the symmetric function  $H_\lambda[X; q, t]$  when  $\ell(\lambda) \leq m$  is given by*

$$H_{1^m}^{qt} = \sum_{R \models m} \sum_v t^{\text{comaj}(R)} q^{|v|} S^{(R, v)} e_v^\perp \quad (4.2)$$

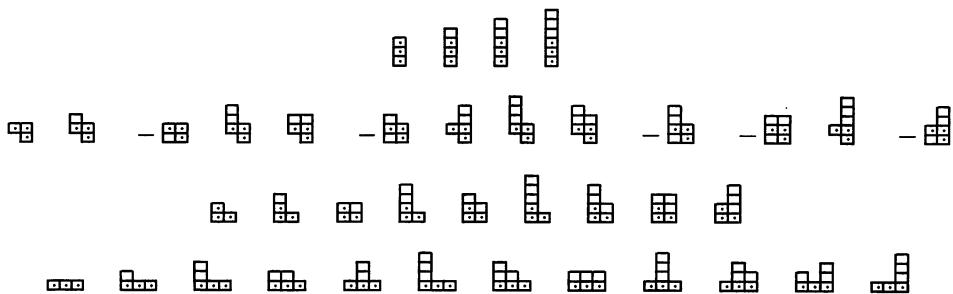
where the sum over  $v$  is over all sequence  $(v_1, v_2, \dots, v_m)$  such that  $0 \leq v_i$  and  $e_v$  is the elementary symmetric function indexed by the sequence  $v$  and  $|v| = \sum_i v_i$ .

**Example 4.2.** We will use formula (4.2) to compute  $H_{222}[X; q, t]$ . This is a long and involved example, but it demonstrates the power of this recurrence since with a reasonable amount of work one can calculate a Macdonald polynomial of size 6 by hand.

We start with the formula for  $H_{111}[X; q, t] = \boxed{\square} + (t + t^2)\boxed{\square\square} + t^3\boxed{\square\square\square}$ .

The sum in equation (4.2) over  $v$  is finite because we need only compute the terms such that  $|v|$  is less than or equal to the degree of the symmetric function that is being acted on, but it is much larger than the sum needed to compute the Hall-Littlewood symmetric function of the same size.

We list all of the relevant operators (those which are non-zero) and we place a dot in the cells that consist of the core of the operator so that it is easy to read the sequence  $v$  from the picture.



To complete this computation we need  $e_\lambda^\perp$  on the symmetric function  $H_{111}[X; t]$  for  $|\lambda| \leq 3$ . This is given by the following list

$$e_1^\perp H_{111}[X; t] = (1 + t + t^2)(ts_2 + s_{1,1})$$

$$e_2^\perp H_{111}[X; t] = (1 + t + t^2)s_1$$

$$e_{11}^\perp H_{111}[X; t] = (1 + 2t + 2t^2 + t^3)s_1$$

$$e_3^\perp H_{111}[X; t] = 1$$

$$e_{21}^\perp H_{111}[X; t] = 1 + t + t^2$$

$$e_{111}^\perp H_{111}[X; t] = 1 + 2t + 2t^2 + t^3$$

The computation proceeds as follows. The coefficient of  $q^0$  is just the Hall-Littlewood symmetric function  $H_{222}[X; t]$  as it was computed in Example 2.2. So the coefficient of  $q^0$  is

$$s_{222} + (t + t^2)s_{321} + t^3s_{33} + t^3s_{411} + (t^2 + t^3 + t^4)s_{42} + (t^4 + t^5)s_{51} + t^6s_6$$

The coefficient of  $q^1$  is the operator  $\boxed{\square} + t\boxed{\square\square} - t\boxed{\square\square\square} + t^2\boxed{\square\square\square\square} + t^2\boxed{\square\square\square\square\square} + t^3\boxed{\square\square\square\square\square\square}$  when it acts on the symmetric function  $(1 + t + t^2)(ts_2 + s_{1,1})$ .

$$(1 + t + t^2)(t^4s_{51} + (t^2 + t^3)s_{411} + t^3s_{42} + (t + t^2)s_{321} + t^2s_{33} + ts_{3111} + s_{2211})$$

The coefficient of  $q^2$  comes from two components,  $\begin{smallmatrix} \square \\ \square \end{smallmatrix} + t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix}$  when it acts on the symmetric function  $e_2^\perp H_{111}[X; q, t]$  and  $t \begin{smallmatrix} \square \\ \square \end{smallmatrix} - t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix}$  when it acts on the symmetric function  $e_{11}^\perp H_{111}[X; q, t]$ . The first part is

$$(t^2 + t + 1) (-t^3 s_{42} + t^3 s_{411} + (t + t^2) s_{3111} - ts_{222} + s_{2211}t + s_{21111})$$

and the second is

$$(1 + 2t + 2t^2 + t^3) (t^3 s_{42} + t^2 s_{321} + ts_{222})$$

The sum of these two quantities is

$$(t^2 + t + 1) (t^4 s_{42} + t^3 s_{411} + (t^2 + t^3) s_{321} + (t + t^2) s_{3111} + t^2 s_{222} + ts_{2211} + s_{21111})$$

The coefficient of  $q^3$  comes from three different operators acting each on a

different constant. The first operator is  $\begin{smallmatrix} \square \\ \square \end{smallmatrix} + t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix}$  when it acts on 1, the second is  $t \begin{smallmatrix} \square \\ \square \end{smallmatrix} - t \begin{smallmatrix} \square \\ \square \end{smallmatrix} - t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^2 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix}$  when it acts on  $1 + t + t^2$ , and the third is  $-t \begin{smallmatrix} \square \\ \square \end{smallmatrix} + t^3 \begin{smallmatrix} \square \\ \square \end{smallmatrix}$  when it acts on the symmetric function  $2t + 2t^2 + t^3 + 1$ . These three parts are

$$t^3 s_{33} - t^3 s_{321} + t^3 s_{3111} - (t + t^2) s_{222} + (t + t^2) s_{21111} + s_{111111}$$

$$(1 + t + t^2) (-2t^3 s_{33} + t^3 s_{321} + (t^2 + 2t) s_{222} + t^2 s_{2211})$$

$$(2t + 2t^2 + t^3 + 1) (-ts_{222} + t^3 s_{33})$$

The sum of these three quantities is

$$t^6 s_{33} + (t^4 + t^5) s_{321} + t^3 s_{3111} + t^3 s_{222} + (t^2 + t + 1) t^2 s_{2211} + (t + t^2) s_{21111} + s_{111111}$$

which is the coefficient of  $q^3$  in  $H_{222}[X; q, t]$ .

## 5. REFERENCES

1. A. M. Garsia, Orthogonality of Milne's polynomials and raising operators, *Discrete Math.* **99** (1992), 247–264.
2. A. M. Garsia and M. Zabrocki, Polynomaility of the  $q, t$ -Kostka revisited, *Proceedings for conference in honor of Gian-Carlo Rota*, to appear.
3. N. Jing, Vertex operators and Hall-Littlewood symmetric functions, *Adv. Math.* **87** (1991), 226–248.

4. I. G. Macdonald, "Symmetric Functions and Hall Polynomials, Oxford Mathematical Monographs," second edition, Oxford Univ. Press, 1995.
5. M. Zabrocki, Ribbon Operators and Hall-Littlewood Symmetric Functions, *Adv. Math.*, to appear.
6. M. Zabrocki,  $q$ -Analogs of symmetric function operators, (preprint).
7. A. V. Zelevinsky, "Representations of finite classical groups: a Hopf algebra approach," Springer Lecture Notes, 869, 1981.

Mike Zabrocki e-mail: [zabrocki@math.uqam.ca](mailto:zabrocki@math.uqam.ca)  
LACIM, UNIVERISTÉ DU QUÉBEC À MONTRÉAL