HABILITATION À DIRIGER DES RECHERCHES

PRÉSENTÉE À

L'UNIVERSITÉ DE MARNE-LA-VALLÉE

INSTITUT GASPARD MONGE

Par Florent Hivert

SPÉCIALITÉ : INFORMATIQUE FONDAMENTALE

SUR LE SUJET

Combinatoire et calcul symbolique dans les algèbres de Hopf

4 octobre 2004

Résumé

Les travaux de combinatoire algébrique présentés dans ce mémoire sont centrés sur les fonctions symétriques et la combinatoire des tableaux. Nous explorons les généralisations de cette situation très riche et, en particulier, de la structure d'algèbre de Hopf. Dans un premier temps, nous décrivons un ensemble de méthodes s'appuyant sur les polynômes non commutatifs et les généralisations du monoïde plaxique, pour construire des algèbres de Hopf sur un grand nombre d'objets combinatoires tels que les permutations, les matrices d'entiers, les arbres et les graphes.

Dans un deuxième temps, nous nous intéressons aux interprétations de ces algèbres de Hopf, en particulier dans la théorie des représentations. Nos résultats montrent que ces structures encodent combinatoirement les représentations de tours d'algèbres, de la même manière que les fonctions symétriques encodent les représentations des groupes symétriques.

Enfin, ce travail de recherche s'appuie continuellement sur l'expérimentation informatique. Je présente la bibliothèque `MuPAD-Combinat` dont je coordonne le développement depuis 2001. Cette bibliothèque permet l'énumération et la manipulation des objets combinatoires, ainsi que la construction de nouvelles structures algébriques.

Table des matières

Introduction

Nous nous intéressons à la combinatoire algébrique, un domaine à l'interface de l'algorithmique, des mathématiques et de la physique. Le but est de décrire des méthodes théoriques et pratiques pour le calcul dans les structures algébriques.

L'un des objets centraux de la combinatoire algébrique actuelle est la structure de tableau de Young en relation avec les fonctions symétriques. En fait, une grande partie de leur propriétés intéressantes se retrouve dans la structure du monoïde plaxique de A. Lascoux et M.-P. Schützenberger, *via* la correspondance de Robinson-Schensted. L'un des premiers problèmes classiques dans ce domaine est le calcul des coefficients de Littlewood-Richardson qui codent la table de multiplication des fonctions symétriques de Schur.

Mon travail de recherche est essentiellement centré sur les possibilités de généraliser cette situation extrêmement riche. Mes premiers résultats ont concerné une généralisation des fonctions symétriques, due à Gessel [Ge], appelées fonctions quasi-symétriques. En imitant l'une des constructions classiques des fonctions de Hall-Littlewood, j'ai pu reproduire les liens qui existent entre le groupe symétrique et les fonctions quasi-symétriques et ainsi, *via* la combinatoire des rubans, construire de nouvelles bases des fonctions quasi-symétriques. Avec J.-Y. Thibon et A. Lascoux, nous avons poussé ce travail plus loin avec des analogues multi-paramétriques «à la Macdonald». Rappelons que les fonctions de Macdonald motivent une très grande part de l'effort de recherche sur ce sujet dans ces dernières années (voir les travaux de Cherednik [C1, C2], Knutson-Tao [KnuTao], Garsia [GH], Haiman [Hai] pour n'en citer que quelques uns).

D'autres résultats portent non plus sur l'étude d'algèbres déjà connues généralisant les fonctions symétriques, mais sur la construction de nouvelles situations. Le premier pas a été fait en collaboration avec G. Duchamp et J.-Y. Thibon. Nous avons construit plusieurs généralisations des fonctions symétriques : les fonctions symétriques libres, les fonctions quasi-symétriques libres et les fonctions quasi-symétriques matricielles. Grâce aux deux premières, nous donnons une nouvelle preuve particulièrement simple et éclairante de la règle de Littlewood-Richardson. La troisième est liée à la généralisation par Knuth de la correspondance de Robinson-Schensted.

Dans toutes ces constructions, l'ingrédient fondamental caché est le monoïde plaxique ou une modification nommée monoïde hypoplaxique. La question de l'existence d'autres monoïdes jouant le même rôle avait en effet été posée par Schützenberger ; J.-Y. Thibon, J.-C. Novelli et moi-même avons montré que le monoïde plaxique n'est pas un cas isolé. Nous avons défini un autre exemple intéressant à partir de l'algèbre des arbres binaires de Loday et Ronco [LR1, LR2]. Nous avons ainsi défini un nouveau monoïde appelé *monoïde sylvestre*, qui permet de construire l'algèbre des arbres binaires en suivant la même procédure que pour le cas classique du monoïde plaxique. L'algorithme de Robinson-Schensted est remplacé ici par un algorithme bien connu, puisque qu'il s'agit de l'insertion d'une lettre dans un arbre binaire de recherche. Nous obtenons donc un parallèle étroit entre la combinatoire des tableaux et celle des arbres binaires. Notons enfin que nous avons trouvé beaucoup d'autres exemples de cette situation, sans pouvoir, pour le moment, en donner une définition précise ou une description exhaustive.

Une autre piste de recherche féconde est l'interprétation de ces algèbres en théorie des représentations. Depuis Frobenius et Schur, on sait que l'algèbre des fonctions symétriques encode combinatoirement la théorie des représentations des groupes symétriques et linéaires. Je me suis donc intéressé aux liens que pouvaient avoir des algèbres de Hopf évoquées précédement avec la théorie des représentations. Dans un premier temps, j'ai donné des analogues des formules des caractères de Weyl et de Demazure pour un groupe quantique dégénéré, ce qui explique la combinatoire des fonctions quasi-symétriques. Plus récemment, j'ai interprété d'une part l'algèbre des pics de Stembridge [St] comme anneau de Grothendieck des algèbres de Hecke-Clifford de Olshanski [O], et d'autre part les algèbres de descente des produits en couronne de Poirier-Mantaci-Reutenauer [ManR, P] comme anneau de Grothendieck des algèbres de Hecke cyclotomique d'Ariki-Koike-Shoji [AK, SakS]. C'est un sujet de recherche très nouveau et je travaille actuellement sur les représentations d'une algèbre qui réunit les groupes symétriques et leurs algèbres de Hecke.

Ce travail de recherche est indissociable de l'expérimentation informatique. C'est pourquoi, en collaboration avec N. Thiéry, je coordonne le développement de la bibliothèque `MuPAD-Combinat` pour le système de calcul formel MuPAD. Il s'agit d'une plate-forme d'expérimentation dont le but est de fournir des algorithmes pour engendrer et manipuler les objets combinatoires, ainsi que pour définir naturellement des structures algébriques dont les bases sont indexées par ces objets combinatoires. Nous encadrons ainsi une équipe d'une dizaine de développeurs réguliers auxquels s'ajoutent des contributeurs occasionnels. Tout ce travail est disponible sous licence LGPL et le noyau stable est intégré dans le logiciel MuPAD lui-même. Nous avons d'ores et déjà des applications importantes en dehors de la combinatoire algébrique, en particulier dans la vérification automatique de programmes.

Ce document est structuré de la manière suivante. Après un chapitre de préliminaires, le deuxième chapitre est dédié à la construction de nouvelles algèbres de Hopf, vues comme fonctions symétriques généralisées. Le chapitre suivant traite de l'interprétation de ces algèbres en théorie des représentations. Le dernier chapitre présente le système d'expérimentation `MuPAD-Combinat` et quelques exemples d'utilisation.

Les références [H1-21] renvoient aux articles de ma liste de publications (ci-dessous).

Publications

[H1] *Analogues non-commutatifs et quasi-symétriques des fonctions de Hall-Littlewood, et modules de Demazure d'une algèbre enveloppante quantique dégénérée*, C. R. Acad. Sci. Paris, **362** (1998), Série I, 1-6.

[H2] *Quasi-symmetric functions and Hecke algebra actions*, dans *Actes du 10e congrès «Séries Formelles et Combinatoire algébrique»*, Fields intitute, Toronto, juin 1998 (N. Bergeron et F. Sottile editeurs), p. 325–339.

[H3] *Combinatoire des fonctions quasi-symétriques*, Thèse de doctorat, Université de Marne-La-Vallée (1999).

[H4] avec G. Duchamp et J.-Y. Thibon, *Some common generalizations of quasi-symmetric functions and noncommutative symmetric functions*, C. R. Acad. Sci. Paris, **328** (1999), Série I, 1113–1116.

[H5] *Hecke Algebras, Differences Operators, and Quasi-Symmetric Functions*, Adv. Math., **155** (2000), 181-238.

[H6] avec G. Duchamp et J.-Y. Thibon, *Some Generalizations of Quasi-symmetric Functions and Noncommutative Symmetric Functions*, Actes du 12e congrès «Séries Formelles et Combinatoire algébrique», Université d'État de Moscou, juin 2000 (D. Krob, A.-A. Mikhalev et A.-V. Mikhalev éditeurs), 170–178, Springer, Berlin, 2000.

[H7] avec J. Cassaigne, M. Espie, D. Krob et J.-C. Novelli, *The Chinese monoid,* Internat. J. Algebra Comput., **11(3)** (2001), 301–334.

[H8] avec F. Boulier, D. Krob et J.-C. Novelli, *Pseudo-permutations. II. Geometry and representation theory,* dans *Discrete models : combinatorics, computation, and geometry* (Paris, 2001), Discrete Math. Theor. Comput. Sci. Proc., AA (2001), 123–132 (électronique).

[H9] avec N. Thiéry, *Deformation of symmetric functions and the rational Steenrod algebra,* dans *Proceedings of the Classical Invariant Theory Workshop*, Kingston, April 8-19, 2002, CRM Proceedings series of the AMS (2002).

[H10] avec G. Duchamp et J.-Y. Thibon, *Noncommutative symmetric functions VI : Free quasi-symmetric functions and related algebras,* Internat. J. Alg. Comp., **2** (2002), 671–717.

[H11] avec J.-C. Novelli et J.-Y. Thibon, *Un analogue du monoïde plaxique pour les arbres binaires de recherche,* C. R. Acad. Sci. Paris, **335** (2002), 1–4.

[H12] avec J.-C. Novelli et J.-Y. Thibon, *Sur quelques propriétés de l'algèbre des arbres binaires,* C. R. Math. Acad. Sci. Paris, **337(9)** (2003), 565–568.

[H13] avec J.-C. Novelli et J.-Y. Thibon, *An analogue of the plactic monoïd for binary search trees*, dans *Proceedings of the 4th Conference on Combinatorics on Words*, Turku, 10-13 septembre 2003), Turku Centre for Comp. Sc. **27** (T. Harju et J. Karhumäki éds.).

[H14] avec N. Thiéry, *Mupad-combinat, an open-source package for research in algebraic combinatorics,* Séminaire Lotharingien de Combinatoire, **51** (2003), 70 p. électronique.

[H15] avec N. Bergeron et J.-Y. Thibon, *The peak algebra and the Hecke-clifford algebras at $q = 0$,* J. Combinatorial Theory A, **117** (2004), 1–19.

[H16] avec J.-C. Novelli et J.-Y. Thibon, *The algebra of binary search trees*, Theoretical Computer Science, à paraître, `math.CO/0401089`.

[H17] avec A. Lascoux et J.-Y. Thibon, *Noncommutative symmetric functions and quasi-symmetric functions with two parameters*, prépublications du Isaac Newton Institute de Cambridge, préprint `math.CO/0106191`.

[H18] avec J.-C. Novelli et J.-Y. Thibon, *Representation theory of the 0-Ariki-Koike-Shoji algebras*, préprint `math.CO/0407218`.

[H19] *On some generalization of quasi-symmetric functions*, en préparation.

[H20] avec N. Thiéry, *The Hecke-Symmetric algebra*, en préparation.

[H21] avec G. Duchamp, J.-C. Novelli et J.-Y. Thibon, *Noncommutative symmetric functions VII : Free quasi-symmetric functions revisited*, en préparation.

Chapitre 1

Préliminaires

1.1 Des séries génératrices aux algèbres de Hopf

Pour compter des objets combinatoires, on utilise habituellement une méthode simple mais très puissante : les séries génératrices. Ainsi, si A est un ensemble d'objets, chacun de ces objets ayant une taille, on définit la *série génératrice* de A comme l'expression

$$f_A(t) := \sum_{a \in A} t^{\mathrm{Taille}(a)} . \tag{1.1}$$

Par exemple, l'ensemble $\{1,2\}$ admet quatre sous-ensembles $\{\}$, $\{1\}$, $\{2\}$ et $\{1,2\}$ dont un est de taille 0, deux de taille 1 et un de taille 2. Ainsi, la série génératrice des sous-ensembles de $\{1,2\}$ est donnée par $1+2t+t^2$. Le coefficient de t^k dans une telle série donne le nombre d'objets de taille k.

La remarque fondamentale est alors que :

- d'une part, si l'ensemble A est la *réunion* de B et C, alors la série génératrice de A est la *somme* des séries génératrices de B et C ;
- d'autre part, si A est *l'ensemble des couples* d'éléments de B et C, la taille d'un couple étant la somme des tailles de ses composantes, alors la série génératrice de A est le *produit* des séries génératrices de B et de C.

Par exemple, un arbre binaire ordonné est soit vide (de taille 0) soit composé d'un noeud (de taille 1) auquel on a greffé deux sous-arbres. Ainsi, un arbre non vide peut être identifié à un triplet (noeud, arbre, arbre) et l'on trouve que la série génératrice de l'ensemble des arbres vérifie l'équation

$$f(t) = 1 + tf(t)f(t) . \tag{1.2}$$

On a ainsi transféré la structure récursive des arbres dans la structure algébrique de l'anneau des séries formelles. Plus précisément, si l'on connaît la manière dont les arbres se *composent*, on peut utiliser les lois de *composition* des séries formelles (somme et produit) pour en calculer le nombre.

Cette méthode est très puissante et très efficace, mais elle perd une grande partie de la richesse des objets combinatoires, car ils ne se contentent pas de se composer, ils se *décomposent* également. On peut ainsi très naturellement construire d'autres opérations que la somme et le produit, que l'on peut aussi utiliser pour faire du comptage. Certaine de ces structures enrichies sont appelées *algèbres de Hopf*. Ce seront les personnages principaux de ce mémoire.

Dans les premières pages de cette section, je vais tenter de décrire de manière informelle et, je l'espère, simple, quelques apparitions naturelles de ces structures algébrico-combinatoires très riches.

1.1.1 Décompositions des mots et produits tensoriels

Soient $a, b, c, \dots$ des lettres. Nous nous intéressons aux suites de lettres que l'on appellera mots, par exemple *abca*. Il existe sur les mots une loi de composition naturelle : la concaténation, c'est-à-dire le collage de deux mots l'un derrière l'autre. Ainsi, si l'on concatène les mots *abca* et *bca*, on obtient le mot *abcabca*. Il existe un mot vide (que l'on note ϵ) qui ne change rien quand on le colle à un autre. Nous avons ici un exemple très classique de composition d'objets combinatoires, c'est l'algèbre des séries formelles non commutatives.

Il existe une autre opération : étant donné un mot, on peut chercher toutes les manières dont on peut le décomposer comme deux sous-mots disjoints. Par exemple, du mot *abcadbac* on peut extraire le sous-mot *badc* en ne gardant que les lettres grasses de $a\mathbf{bcad}ba\mathbf{c}$; d'autre part, si l'on efface les mêmes lettres grasses, il reste le mot *acba*. Ainsi, on peut dire que le mot *abcadbac* se décompose en le couple de mots *badc* et *acba*. Évidement, il y a beaucoup d'autres décompositions. Voici la liste de toutes les décompositions du mot *badc* :

$$(\epsilon, badc), (b, adc), (a, bdc), (d, bac), (c, bad), (ba, dc), (bd, ac), (bc, ad),$$
$$(ad, bc), (ac, bd), (dc, ba), (bad, c), (bac, d), (bdc, a), (adc, b), (badc, \epsilon)$$

J'ai choisi un mot très particulier de manière à ce que toutes les décompositions soient différentes. En partant du mot *aab*, on peut obtenir deux fois la décomposition (a, ab) : soit par $\mathbf{a}ab$ soit par $a\mathbf{a}b$. Bien sûr, on a envie de tenir compte des multiplicités et de dire que *aab* se décompose en

$$(\epsilon, aab) + 2(a, ab) + (b, aa) + (aa, b) + 2(ab, a) + (aab, \epsilon)$$

On se retrouve donc à devoir compter avec multiplicité des couples d'objets, étant entendu que, avec 2 objets X et 3 objets Y, il y a 6 manières de former une paire (X, Y). On trouve ainsi la formule

$$(2X, 3Y) = 6(X, Y)\,.$$

La formalisation mathématique de cette construction porte le nom de *produit tensoriel* et se note $\otimes$. La formule précédente s'écrit donc

$$2X \otimes 3Y = 6X \otimes Y\,.$$

L'opération de décomposition des mots décrite plus haut est donc une opération qui va des mots dans les produits tensoriels de mots :

$$\Delta(aab) = \epsilon \otimes aab + 2a \otimes ab + b \otimes aa + aa \otimes b + 2ab \otimes a + aab \otimes \epsilon\,.$$

De plus, cette opération de décomposition est compatible avec la multiplication des mots : c'est un morphisme. Un tel morphisme de $\mathbb{K}\langle A\rangle$ vers $\mathbb{K}\langle A\rangle \otimes \mathbb{K}\langle A\rangle$ est, en un certain sens, dual d'une opération de produit ; on l'appelle par conséquent un *coproduit*. C'est l'opération principale que l'on ajoute à une structure d'algèbre pour faire une algèbre de Hopf.

Je vais maintenant décrire un exemple montrant comment ce genre d'opération apparaît naturellement dans des questions de combinatoire.

1.1.2 Les ensembles

Étant donnée une liste d'objets combinatoires, par exemple a, b, c, d, nous appelons e_i l'opération qui consiste à prendre les ensembles de taille i d'objets de la liste :

$$e_3(a, b, c, d) := \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}\,,$$

que je préfère noter

$$e_3(a + b + c + d) := abc + abd + acd + bcd\,.$$

Si maintenant nous décidons que les deux premiers objets de la liste sont indistinguables, tout en restant différents, on obtient

$$e_3(a + a + c + d) = e_3(2a + c + d) = aac + aad + 2acd = a^2c + a^2d + 2acd\,.$$

La notation a^2c exprime que l'on trouve un ensemble contenant deux a et un c.

Une manière de rendre des objets indistinguables est de ne retenir que leur taille. Par exemple, si l'on suppose que a et b sont de taille 1, que c est de taille 2 et que d est de taille 4, on est conduit à écrire la formule

$$e_3(2t + t^2 + t^4) := ttt^2 + ttt^4 + 2tt^2t^4 = t^4 + t^6 + 2t^7$$

qui exprime que dans un ensemble formé de deux objets de taille 1, un de taille 2 et un de taille 4, on peux faire un sous-ensemble de taille 4, un de taille 6 et 2 de taille 7. L'opération précédente peut être rendue rigoureuse et cohérente. On peut ainsi voir e_i comme une opération qui, étant donnée la série génératrice d'un ensemble d'objets, donne la série génératrice des sous-ensembles de taille i.

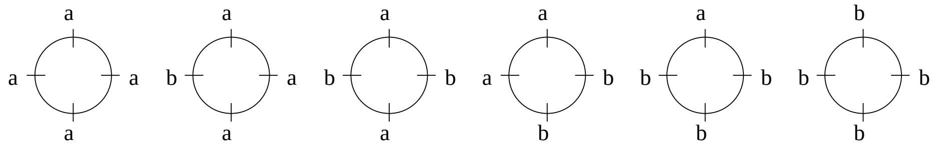
Notons que dans la première expression

$$e_3(a + b + c + d) := abc + abd + acd + bcd\,,$$

les lettres a, b, c, d jouent des rôles symétriques. On a ainsi décrit une expression de type polynomial, symétrique en a, b, c, d. L'objet e_i que je viens de décrire s'appelle une fonction symétrique élémentaire.

1.1.3 Les colliers

Je présente maintenant un deuxième exemple des fonctions symétriques agissant comme opérateurs sur les objets et les séries génératrices. Repartons de deux objets a, b. On s'intéresse maintenant aux colliers (orientés) de longueur 4 que l'on peut fabriquer avec ces objets, c'est-à-dire aux manières de placer des objets a, b sur un cercle à rotation près. Voici les différentes solutions :



Ainsi, si l'on ne retient que le nombre d'apparitions de chaque lettre, on trouve

$$c_2(a+b) = a^4 + a^3b + 2a^2b^2 + ab^3 + b^4 \,.$$

Il est possible de regrouper les termes de cette expression : par exemple, si l'on considère les colliers ne contenant que des objets identiques, ici 4 a ou bien 4 b, il n'y a toujours qu'un seul collier. De même, si l'on a trois fois le même objet et une fois un autre objet, il y a également une seule solution. En revanche, avec deux objets a et deux objets b, il y a deux solutions. En regroupant ainsi les termes par la manière dont on a choisi les objets à placer sur le cercle, on peut alors écrire

$$c_2(a+b) = m_4(a+b) + m_{3,1}(a+b) + 2m_{2,2}(a+b)$$

où le coefficient de $m_{i,j,k,\ldots}$ décrit le nombre de manières de construire un collier en utilisant i objets d'un type et j objets d'un autre type, etc. L'avantage de cette écriture est qu'elle est indépendante du nombre de type d'objets considérés. Ainsi, on a toujours

$$c_2 = m_4 + m_{3,1} + 2\,m_{2,2} + 3\,m_{2,1,1} + 6\,m_{1,1,1,1} \,.$$

Si l'on écrit chacun des m_λ sur les variables, on obtient à nouveau une expression symétrique : c'est en effet la somme de tous les monômes dont les exposants sont donnés par la suite d'entiers λ. On appelle une telle somme une fonction symétrique monomiale.

Il existe une très belle formule due à Polyà qui permet, entre autres, d'écrire n'importe quel opérateur c_i sur les m_i. Par exemple,

$$\begin{aligned} c_6 = 10\,m_{3,2,1} + 60\,m_{2,1,1,1,1} + 20\,m_{3,1,1,1} + 30\,m_{2,2,1,1} + 5\,m_{4,1,1} + \\ m_{5,1} + 3\,m_{4,2} + 16\,m_{2,2,2} + 4\,m_{3,3} + m_6 + 120\,m_{1,1,1,1,1,1} \,. \end{aligned}$$

Ainsi, il existe 16 colliers de longueur 6 avec deux a deux b et deux c.

1.1.4 Structures algébriques

Nous avons mis en parallèle des constructions sur les objets combinatoires et des constructions sur les séries génératrices. En fait, on peut voir chacun des opérateurs définis précédemment (ensemble e_i, collier c_i et monôme m_i) comme agissant soit sur les objets eux-mêmes, soit sur les séries génératrices. L'une des formalisations de cette remarque est la théorie des espèces de Joyal. Notons qu'il existe un grand nombre d'autres opérateurs et qu'il est possible de définir l'action d'un opérateur sur des séries plus générales que les polynômes, typiquement les fractions rationnelles.

Le pléthysme

Nous avons déjà vu deux manières de combiner ces opérateurs (addition, multiplication), mais il en existe beaucoup d'autres. Par exemple, on peut vouloir compter les paires de taille 2 de colliers d'objets de longueur 3. On (*i.e.* mon ordinateur) trouve alors

$$\begin{aligned} e_2 \circ c_3 = 4\,m_{3,2,1} + 20\,m_{2,1,1,1,1} + 8\,m_{3,1,1,1} + 10\,m_{2,2,1,1} + \\ 3\,m_{4,1,1} + m_{5,1} + m_{4,2} + 4\,m_{2,2,2} + 2\,m_{3,3} + 40\,m_{1,1,1,1,1,1}\,. \end{aligned}$$

Par exemple, en utilisant 3 fois l'objet a et 3 fois l'objet b, il y a deux paires de colliers $\{[aaa],[bbb]\}$ et $\{[aab],[abb]\}$ et avec $2a$, $2b$ et $2c$, il y a 4 paires de colliers : $\{[aac],[bbc]\}$, $\{[abb],[acc]\}$, $\{[aab],[bcc]\}$ et $\{[abc],[acb]\}$.

Il apparaît ainsi naturellement sur ces opérateurs une opération de composition que l'on appelle habituellement pléthysme, car elle produit, quand on la développe, une pléthore de termes. Algorithmiquement, elle est difficile à calculer efficacement.

Le coproduit

Il existe encore une autre opération importante. Soient deux ensembles d'objets B et C dont l'union est A. On veut décrire les sous-ensembles de A en fonction des sous-ensembles de B et C. Il est clair par exemple qu'un sous-ensemble de A de taille 3 est

- soit inclus dans B ;
- soit la réunion d'un sous-ensemble de B de taille 2 et d'un sous-ensemble de C de taille 1 ;
- soit la réunion d'un sous-ensemble de B de taille 1 et d'un sous-ensemble de C de taille 2 ;
- soit inclus dans C.

Ceci s'écrit naturellement par la formule

$$e_3(A) = e_3(B+C) = e_3(B) + e_2(B)e_1(C) + e_1(B)e_2(C) + e_3(C)\,, \qquad (1.3)$$

ou de manière plus uniforme en utilisant les ensembles vides :

$$e_3(B+C) = e_3(B)e_0(C) + e_2(B)e_1(C) + e_1(B)e_2(C) + e_0(B)e_3(C)\,. \qquad (1.4)$$

Plus généralement, on trouve

$$e_n(B+C) = \sum_{i+j=n} e_i(B)e_j(C)\,. \tag{1.5}$$

Cette dernière formule décompose l'opérateur e_n en plusieurs opérateurs. Elle est exactement du même type que la décomposition des mots vue précédemment et elle définit un coproduit sur l'algèbre des e_i :

$$\Delta(e_n) = \sum_{i+j=n} e_i \otimes e_j\,. \tag{1.6}$$

On peut bien sûr donner des formules analogues pour décomposer les opérateurs «manières» et «colliers» évoqués plus haut.

Ainsi nous savons combiner deux opérateurs d'au moins quatre façons différentes : l'union qui correspond à la somme, les couples qui correspondent au produit, la composition qui correspond au pléthysme et la décomposition qui correspond au coproduit. Non seulement il en existe d'autres, mais on peut également jouer au jeu qui consiste à utiliser plusieurs de ces lois de composition et décomposition des opérateurs dans la même formule. On a ainsi une structure algébrique extrêmement riche avec un grand nombre d'opérations.

La structure des opérateurs que l'on vient d'évoquer est la structure des fonctions symétriques, qui est le prototype des structures algébriques étudiées dans ce mémoire. Dans la suite de ces préliminaires, je vais formaliser ce que nous avons évoqué dans cette introduction.

1.1.5 Définition formelle

Pour fixer le cadre et les notations, je rappelle très brièvement les définitions formelles des notions d'algèbres de Hopf. Sur ce point, on peut notamment consulter [A].

Produits tensoriels et algèbres de Hopf

On rappelle qu'une *algèbre* sur un corps $\mathbb{K}$ est un triplet $(A, 1, \cdot)$ où A est un espace vectoriel sur $\mathbb{K}$, et où $\cdot$ désigne une application bilinéaire associative de $A \times A$ dans A pour laquelle 1 est l'élément neutre. Dans tout ce mémoire, les algèbres auront pour corps de base le corps $\mathbb{C}$ des nombres complexes.

Si U et V sont deux espaces vectoriels sur $\mathbb{C}$, le *produit tensoriel* $U \otimes V$ est un espace vectoriel avec une application bilinéaire $f\ :\ U \times V \mapsto U \otimes V$ tels que pour toute application bilinéaire de $g\ :\ U \times V \mapsto W$, il existe une application linéaire de $h\ :\ U \otimes V \mapsto W$ vérifiant $g = h \circ f$. Si B_U est une base de U et B_V une base de V, alors le produit tensoriel a pour base les symboles $u \otimes v$ où u décrit B_U et v décrit B_V. Le produit tensoriel de deux algèbres A et B est naturellement lui-même une algèbre pour la loi de produit

$$(f_1 \otimes g_1)(f_2 \otimes g_2) := f_1 f_2 \otimes g_1 g_2\,, \tag{1.7}$$

l'unité étant donnée par $1_A \otimes 1_B$.

La définition du produit tensoriel montre que le produit d'une algèbre peut être vu comme une application μ linéaire de $A \otimes A \mapsto A$. Un *coproduit* Δ est le dual d'un produit, c'est donc une application linéaire de V dans $V \otimes V$. Le dual de la notion d'unité est la notion de *co-unité* : c'est une application linéaire c de V dans $\mathbb{K}$ qui vérifie

$$(\mathrm{Id} \otimes c) \circ \Delta = (c \otimes \mathrm{Id}) \circ \Delta = \mathrm{Id} \tag{1.8}$$

dans l'identification naturelle de V à $\mathbb{K} \otimes V$ et à $V \otimes \mathbb{K}$. Une *cogèbre* est un espace vectoriel avec un coproduit et une co-unité.

Définition 1.1 *Une* bigèbre *ou* algèbre de Hopf *est un espace vectoriel qui est à la fois une algèbre et une cogèbre tel que le coproduit soit un morphisme d'algèbre de A dans $A \otimes A$.*

La définition usuelle d'une algèbre de Hopf requiert, en plus d'être une bigèbre telle que je l'ai définie ici, l'existence d'une application appelée antipode. Dans le paragraphe qui va suivre, je vais demander que toutes les algèbres de Hopf soient graduées et connexes, ce qui garantit automatiquement l'existence de l'antipode ; je me permets donc ce raccourci.

Graduations

Un *espace vectoriel gradué* est un espace vectoriel V qui se décompose en une somme directe

$$V = \bigoplus_{n \in \mathbb{N}} V_n \,. \tag{1.9}$$

On appelle les V_n les *composantes homogènes* de V, et dans ce mémoire, on demande de plus qu'elles soient toutes de dimension finie. Si U et V sont deux espaces gradués, alors $U \otimes V$ est naturellement gradué par

$$(U \otimes V)_n = \bigoplus_{i+j=n} U_i \otimes V_j \quad \textit{i.e.} \qquad \deg(u \otimes v) = \deg(u) + \deg(v)\,. \tag{1.10}$$

Une algèbre, une cogèbre ou une algèbre de Hopf est dite *graduée* si l'espace vectoriel sous-jacent est gradué et si les opérations de produit et coproduit conservent le degré. On dit de plus qu'une algèbre de Hopf est *connexe* si la composante de degré zéro est de dimension 1, c'est-à-dire si c'est la droite engendrée par l'unité. On peut alors naturellement identifier cette droite avec le corps de base. **Toutes les algèbres de Hopf considérées dans ce mémoire seront graduées et connexes**. Comme, je l'ai dit plus haut, ces deux conditions garantissent l'existence de l'antipode. Je me permets donc d'oublier son existence.

Dualité

On considère ici la notion de dualité dans les algèbres de Hopf graduées connexes où toutes les composantes homogènes sont de dimension finie. Dans ce cadre, Le *dual gradué* d'un espace vectoriel est la somme des duaux des composantes homogènes. Le coproduit dual d'un produit est défini par

$$\langle \Delta(a) | b \otimes c \rangle = \langle a | bc \rangle \ , \tag{1.11}$$

où $\langle x|y \rangle$ est le crochet de dualité. Deux algèbres de Hopf sont dites *duales* l'une de l'autre, si les espaces vectoriels gradués sous-jacent sont duaux et si le produit de chacune des algèbres est le dual du coproduit de l'autre.

Sur la base de ces définitions techniques, je vais maintenant présenter les principaux exemples intéressants.

1.2 Fonctions symétriques et généralisations

1.2.1 Fonctions symétriques

Le groupe symétrique $\mathfrak{S}_n$ agit naturellement sur les polynômes en $X = \{x_1, x_2, \ldots, x_n\}$ par permutation des variables

$$(\sigma\, f)(x_1, \ldots, x_n) = f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}). \tag{1.12}$$

Un polynôme f est dit *symétrique* s'il est invariant par échange des variables, c'est-à-dire si

$$\sigma\, f = f \quad \text{pour tout } \sigma \in \mathfrak{S}_n. \tag{1.13}$$

Il est souvent pratique de considérer un alphabet X infini. Ceci est possible car les morphismes de restriction d'alphabet

$$\rho_{m,n} \ : \ f(x_1, \ldots, x_m) \longmapsto f(x_1, \ldots, x_n, 0, \ldots, 0) \tag{1.14}$$

sont compatibles (c'est-à-dire $\rho_{m,p} = \rho_{n,p} \circ \rho_{m,n}$ pour $m > n > p$). On parle alors de *fonctions symétriques* plutôt que de polynômes symétriques. Les fonctions symétriques sont donc des sommes formelles de monômes, de degré fini, invariantes par échange des variables. On note Sym l'algèbre des fonctions symétriques et $\mathrm{Sym}(X)$ l'algèbre des polynômes symétriques sur l'alphabet X.

1.2.1.1 Fonctions symétriques élémentaires

On appelle *r-ième fonction symétrique élémentaire* la fonction définie par

$$e_r = \sum_{i_1 < i_2 < \cdots < i_r} x_{i_1} x_{i_2} \ldots x_{i_r}. \tag{1.15}$$

Il est souvent plus pratique de manipuler les fonctions élémentaires par leur série génératrice : on la note $E(t)$, elle est donnée par

$$E(t) = \sum_{r \geq 0} e_r t^r = \prod_{i \geq 1} (1 + x_i t). \tag{1.16}$$

La structure de l'algèbre des polynômes symétriques est décrite par le théorème suivant que l'on appelle théorème fondamental des polynômes symétriques.

Théorème 1.2 *L'algèbre des polynômes symétriques en n variables est librement engendrée par les polynômes symétriques élémentaires* $(e_i)_{1\leq i\leq n}$.

L'algèbre des fonctions symétriques est librement engendrée par les fonctions symétriques élémentaires $(e_i)_{i\geq 0}$.

On peut donc voir les fonctions symétriques comme les polynômes en les variables $(e_i)_{i\geq 1}$ avec la graduation particulière $\deg(e_i) = i$. On en déduit que la famille des $e_\lambda = e_{\lambda_1} e_{\lambda_2} \dots e_{\lambda_p}$, où $\lambda = (\lambda_1 \geq \dots \geq \lambda_p)$ décrit l'ensemble des partitions, est une base de Sym.

1.2.1.2 Fonctions symétriques monomiales

Une autre base naturelle de Sym est donnée par les fonctions symétriques monomiales. Soit λ une partition. La *fonction symétrique monomiale* d'indice λ est la fonction définie par

$$m_\lambda = \sum x_1^{\lambda_1} \cdots x_r^{\lambda_r}, \tag{1.17}$$

somme de toutes les permutations distinctes du monôme $x_1^{\lambda_1} \cdots x_r^{\lambda_r}$, c'est-à-dire somme de tous les monômes dont la partition obtenue en réordonnant les exposants est égale à λ.

1.2.1.3 Fonctions complètes

On définit la *r-ieme fonction homogène complète* comme la somme de tous les monômes de degré r :

$$h_r = \sum_{\lambda \vdash r} m_\lambda \,. \tag{1.18}$$

Leur série génératrice est reliée à la série des fonctions élémentaires de la manière suivante :

$$H(t) = \sum_{r\geq 0} h_r t^r = E(-t)^{-1} = \prod_{i\geq 1} (1 - x_i t)^{-1}. \tag{1.19}$$

Comme dans le cas des fonctions élémentaires, la famille $h_\lambda = h_{\lambda_1} h_{\lambda_2} \dots h_{\lambda_p}$, où $\lambda = (\lambda_1 \dots \lambda_p)$ décrit l'ensemble des partitions, est une base de Sym.

1.2.1.4 Dualité

On peut munir l'algèbre des fonctions symétriques d'une structure de Hopf naturelle pour laquelle elle est autoduale : on définit sur Sym un produit scalaire par la formule

$$\langle m_\lambda | h_\mu \rangle = \delta_{\lambda,\mu} \,. \tag{1.20}$$

La formule suivante, qui matérialise l'autodualité de Sym, est appelée formule de Cauchy. Soient u_λ et v_μ deux bases duales de Sym ; alors

$$\langle u_\lambda | v_\mu \rangle = \delta_{\lambda,\mu} \quad \Longleftrightarrow \quad \prod_{i,j} (1 - x_i y_j)^{-1} = \sum_\lambda u_\lambda(X) v_\mu(Y) \,. \tag{1.21}$$

Ceci nous conduit à définir sur Sym le coproduit suivant, dual du produit usuel :

$$\Delta(h_r) = \sum_{i+j=r} h_i \otimes h_j \,. \tag{1.22}$$

On montre alors que Sym, munie du produit usuel et du coproduit Δ, est une algèbre de Hopf graduée autoduale.

1.2.2 Fonctions de Schur

Les fonctions de Schur, bien qu'ayant été définies par Jacobi [J], portent le nom de Schur car c'est lui qui a mis à jour le rôle fondamental joué par ces dernières dans la théorie des représentations [Schur]. Nous commençons par rappeler la définition originale en termes d'alternants.

On appelle *alternant* le déterminant de Vandermonde :

$$a_\delta = \prod_{1\le i<j\le n} (x_i - x_j) = \mathrm{Det}(x_i^{n-j}) \,. \tag{1.23}$$

Soit $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ un monôme. On considère le polynôme obtenu par antisymétrisation de x^α :

$$a_\alpha = \sum_{\sigma\in\mathfrak{S}_n} \mathrm{Sgn}(\sigma)\, \sigma(x^\alpha) \,. \tag{1.24}$$

L'alternant est donc l'antisymétrisé du monôme x^δ, appelé *monôme escalier*, défini par $x^\delta = x_1^{n-1} x_2^{n-2} \cdots x_{n-2}^2 x_{n-1}$. Toute fonction antisymétrique est divisible par l'alternant. Si λ est une partition, on pose $x^{\lambda+\delta} = x_1^{\lambda_1+n-1} \cdots x^{\lambda_n}$. On appelle *fonction de Schur* la fonction s_λ définie par

$$s_\lambda = a_{\lambda+\delta}/a_\delta = \frac{\sum_{\sigma\in\mathfrak{S}_n} \mathrm{Sgn}(\sigma)\, \sigma(x^{\lambda+\delta})}{\sum_{\sigma\in\mathfrak{S}_n} \mathrm{Sgn}(\sigma)\, \sigma(x^\delta)} \,. \tag{1.25}$$

On montre que les fonctions de Schur sont des fonctions symétriques à coefficients entiers, et que la famille $\{s_\lambda\}$ est une base de Sym.

Il existe une définition combinatoire des fonctions de Schur en termes de tableaux :

$$s_\lambda = \sum_{T\in\mathrm{Tab}(\lambda)} x^T \,. \tag{1.26}$$

où $\mathrm{Tab}(\lambda)$ désigne l'ensemble des tableaux de forme λ, et x^T l'image commutative de la lecture du tableau.

Les fonctions de Schur forment une base autoduale. Autrement dit, elles sont orthonormées pour le produit scalaire de (1.20) :

$$\langle s_\lambda | s_\mu \rangle = \delta_{\lambda\mu} \,. \tag{1.27}$$

C'est une conséquence de l'identité de Cauchy :

$$\sum_\lambda s_\lambda(X) s_\lambda(Y) = \prod_{i,j} (1 - x_i y_j)^{-1} \,. \tag{1.28}$$

1.2.3 Fonctions symétriques non commutatives

Le théorème fondamental de la théorie des fonctions symétriques (théorème 1.2) affirme que l'algèbre des fonctions symétriques n'est autre que l'algèbre des polynômes commutatifs sur les indéterminées indépendantes e_k, avec la graduation $\deg(e_k) = k$.

En oubliant l'alphabet sous-jacent, on définit l'algèbre des *fonctions symétriques non commutatives* comme l'algèbre associative libre sur une suite infinie Λ_k d'indéterminées non commutatives (appelées fonctions élémentaires non commutatives) graduée par $\deg(\Lambda_k) = k$ (*cf.* [GKLLRT]).

On définit la série génératrice comme dans le cas commutatif :

$$\lambda(t) = \sum_{n\geq 0} \Lambda_n t^n \,. \tag{1.29}$$

Les *fonctions homogènes complètes non commutatives* S_n sont définies par leur série génératrice :

$$\sigma(t) = \lambda(-t)^{-1} = \sum_{n\geq 0} S_n t^n \,. \tag{1.30}$$

Il existe une réalisation concrète, notée $\mathbf{Sym}(A)$, des fonctions symétriques non commutatives dans l'algèbre libre. Soit $A = \{a_n | n \geq 1\}$ une suite infinie d'indéterminées non commutatives de degré 1. On pose

$$\sum_{n\geq 0} \Lambda_n(A) t^n = \overleftarrow{\prod_{i\geq 1}} (1 + t a_i) = \cdots (1 + t a_3)(1 + t a_2)(1 + t a_1)\,. \tag{1.31}$$

Ainsi $\Lambda_n(A)$ est identifié avec la somme de tous les mots strictement décroissants de longueur n et $S_n(A)$ avec la somme des mots de longueur n croissants au sens large.

Le morphisme d'algèbres

$$\begin{aligned} \mathrm{Eval}\ : \mathbf{Sym}(A) &\to \mathrm{Sym}(X) \\ a_i &\mapsto x_i, \end{aligned} \tag{1.32}$$

qui envoie un mot sur son image commutative, envoie Λ_n sur e_n. Ainsi **Sym** est un relèvement non commutatif de l'algèbre des fonctions symétriques classiques.

Il est à noter que les fonctions symétriques non commutatives ainsi réalisées ne sont pas symétriques pour l'action classique du groupe symétrique, mais pour une action différente due à Lascoux et Schützenberger [LS1].

Les mêmes formules que dans le cas commutatif peuvent être utilisées pour munir **Sym** d'une structure d'algèbre de Hopf :

$$\Delta\Lambda_n = \sum_{k=0}^{n} \Lambda_k \otimes \Lambda_{n-k}\,, \qquad \Delta S_n = \sum_{k=0}^{n} S_k \otimes S_{n-k}\,. \tag{1.33}$$

Enfin, l'involution $\tilde{\omega}(\Lambda_n) = (-1)^n S_n$ vue comme antipode fait de **Sym** une algèbre de Hopf.

Le rôle des fonctions de Schur est ici joué par les fonctions de Schur rubans. On les définit par la somme alternée

$$R_I = \sum_{I \succeq J} (-1)^{\ell(I)-\ell(J)} \, S^J \quad \text{ou encore} \quad S^I = \sum_{I \succeq J} R_J \,, \tag{1.34}$$

où $\succeq$ désigne l'ordre de raffinement des compositions. L'image commutative de R_I est la fonction de Schur ruban ordinaire introduite par Mac-Mahon [MacM]. Ces fonctions de Schur rubans ont une règle de multiplication particulièrement simple [Mal] :

$$R_I \, R_J = R_{I \triangleright J} + R_{I \cdot J} \,. \tag{1.35}$$

Cette règle était connue de Mac-Mahon dans le cas commutatif.

1.2.4 Fonctions quasi-symétriques

Il n'existe pas d'analogue non commutatif des fonctions monomiales. En effet, dans le cas commutatif, elles forment la base duale des fonctions complètes. Or, le coproduit de **Sym** est co-commutatif. Il faut donc chercher l'analogue des fonctions monomiales dans une algèbre commutative duale de **Sym**. C'est l'algèbre des fonctions quasi-symétriques.

Un polynôme $f \in \mathbb{C}[X]$ est dit *quasi-symétrique* si, pour toute composition $I = (i_1, \ldots, i_r)$, le coefficient de

$$\{x_{j_1}, x_{j_2}, \ldots, x_{j_r}\}^I = x_{j_1}^{i_1} x_{j_2}^{i_2} \cdots x_{j_r}^{i_r} \tag{1.36}$$

est indépendant de $j_1 < j_2 < \cdots < j_r$. Remarquons que l'ordre des variables est ici fondamental. Comme pour les fonctions symétriques, on fait tendre le nombre de variables vers l'infini, en prenant la limite projective dans la catégorie des algèbres graduées, et on obtient une algèbre appelée algèbre des *fonctions quasi-symétriques* notée QSym [Ge]. On peut voir de telles «fonctions» comme des polynômes sur un alphabet infini $X = \{x_1 < x_2 < \cdots < x_n < \cdots\}$.

Il est clair que les fonctions dites *quasi-monomiales*

$$M_I = \sum_{j_1 < \cdots < j_r} x_{j_1}^{i_1} \cdots x_{j_r}^{i_r} \tag{1.37}$$

indexées par les compositions $I = (i_1 \ldots, i_r)$, forment une base de QSym.

Une deuxième base importante de QSym est formée par les *fonctions quasi-rubans*. Dans [Ge], Gessel les définit comme les fonctions caractéristiques des permutations qui ont un ensemble de descentes donné. Elles apparaissent comme les analogues des fonctions de Schur [KT1, KT3]. On peut les définir par

$$F_I = \sum_{I \preceq J} M_J \,. \tag{1.38}$$

La dualité entre **Sym** et QSym est alors donnée par le crochet

$$\left\langle M_I | S^J \right\rangle = \delta_{IJ} \quad \text{où encore} \quad \langle F_I | R_J \rangle = \delta_{IJ} \,. \tag{1.39}$$

On vérifie alors que **Sym** est le dual gradué de QSym, c'est-à-dire que :

$$\langle f \otimes g | \Delta(P) \rangle = \langle fg | P \rangle \,, \tag{1.40}$$
$$\langle \gamma(f) | P \otimes Q \rangle = \langle f | PQ \rangle \,, \tag{1.41}$$

où le coproduit de QSym est défini comme suit : soit Y un deuxième alphabet et $X \sqcup Y$ la somme ordinale de X et de Y, c'est-à-dire l'union de X et de Y où les variables de X sont plus petites que les variables de Y. Alors, toute fonction $f \in \mathrm{QSym}(X)$ s'étend naturellement à une fonction quasi-symétrique $f(X \sqcup Y)$ que l'on peut voir comme une fonction

$$\gamma(f) \in \mathrm{QSym}(X) \otimes \mathrm{QSym}(Y) \,. \tag{1.42}$$

1.2.5 Anneaux de Grothendieck d'une tour d'algèbres

Nous décrivons ici une construction très féconde en algèbres de Hopf combinatoire. Cette construction est issue de la théorie des représentations. Je commence donc par quelques rappels succincts.

On considère les *modules* de dimension finie sur une algèbre A, c'est-à-dire les espaces vectoriels M avec une action linéaire «·» de A sur M :

$$a \cdot (b \cdot m) = (ab) \cdot m \quad \text{pour } a, b \in A \text{ et } m \in M \,. \tag{1.43}$$

Un module est dit *simple* ou *irréductible* s'il n'admet pas d'autre sous-module que lui-même et le module trivial réduit à 0. Une algèbre de dimension finie n'admet qu'un nombre fini de modules simples deux à deux non isomorphes.

Pour tout module M, le théorème de Jordan-Hölder permet de définir une notion de multiplicité des modules simples dans un module M qui est parfaitement adaptée à une étude combinatoire. Ainsi, si $(S_i)_{i \in I}$ est un système de représentant des modules simples à isomorphisme près, on définira pour tout module M, le symbole $[M]$ par la somme formelle

$$[M] = \sum_{i \in I} m_i S_i \,, \tag{1.44}$$

où m_i est la multiplicité du module S_i dans M. L'ensemble de ces combinaisons linéaires est appelé *groupe de Grothendieck* et est noté $G_0(A)$. L'intérêt de cette définition réside dans le fait qu'il est compatible avec les notions de somme directe et de quotient

$$[M \oplus N] = [M] + [N] \qquad \text{et} \qquad [M/N] = [M] - [N] \,. \tag{1.45}$$

C'est une sorte de série génératrice qui encode, dans une certaine mesure, la structure du module M.

L'ingrédient important pour notre étude est donné par la définition suivante :

Définition 1.3 *On appelle* tour inductive et associative *d'algèbres toute suite A_n d'algèbres*

$$\mathbb{C} = A_0 \subset A_1 \subset \cdots \subset A_n \subset \cdots, \tag{1.46}$$

munie de morphismes d'inclusion de $\rho_{m,n} : A_m \otimes A_n \hookrightarrow A_{m+n}$ pour tout m, n qui satisfont la condition d'associativité suivante

$$\rho_{i,j+k} \circ (\mathrm{Id}_i \otimes \rho_{j,k}) = \rho_{i+j,k} \circ (\rho_{i,j} \otimes \mathrm{Id}_k).$$

La plupart du temps, on dira simplement tour d'algèbres. De plus, les A_n seront de dimension finie sur $\mathbb{C}$.

Étant donnée une injection d'algèbres $\phi : A \mapsto B$, il existe deux constructions duales dues à Frobenius qui permettent de passer des modules de A aux modules de B et inversement. Si l'on a un B-module M, on peut faire agir A dessus par $a.m = \phi(a).m$. On obtient un A-module. On dit alors que l'on a *restreint* le module de B à A ; on note le module obtenu $\mathrm{Res}_A^B(M)$. Il est possible de faire une construction duale, c'est *l'induction* définie par

$$\mathrm{Ind}_A^B(M) = B \otimes_A M. \tag{1.47}$$

Ainsi, étant donnée une tour d'algèbres, on peut définir le *produit d'induction* de deux modules M et N sur A_m et A_n par l'induction

$$\mathrm{Ind}_{A_m \otimes A_n}^{A_{m+n}} (M \otimes N). \tag{1.48}$$

Si au lieu de travailler vraiment avec les modules, on se contente de compter les multiplicités de Jordan-Hölder, on a une opération de

$$\mathrm{Ind}_{A_m \otimes A_n}^{A_{m+n}} \; : \; G_0(A_m) \times G_0(A_n) \mapsto G_0(A_{n+m}), \tag{1.49}$$

qui définit un produit associatif sur la somme directe

$$\mathcal{G} := \bigoplus_{n \in \mathbb{N}} G_0(A_n). \tag{1.50}$$

La restriction fournit, pour sa part, un coproduit naturel sur cette somme directe, faisant de $\mathcal{G}$ une algèbre de Hopf graduée.

Le premier exemple historique de cette situation est la tour des algèbres des groupes symétriques. L'anneau de Grothendieck $\mathcal{G}$ peut en effet s'identifier naturellement à l'algèbre des fonctions symétriques en envoyant le module simple V_λ sur la fonction de Schur s_λ. Ainsi, le coefficient de Littlewood-Richardson $c^\nu_{\lambda,\mu}$ qui exprime la décomposition du produit de deux fonctions de Schur

$$s_\lambda s_\mu := \sum_\nu c^\nu_{\lambda,\mu} s_\nu \tag{1.51}$$

s'interprète comme la multiplicité du module V_ν dans le produit d'induction de V_λ par V_μ.

Si l'on remplace le groupe symétrique par l'algèbre de Hecke dégénérée $H_n(0)$, on trouve l'algèbre des fonctions quasi-symétriques, la base des modules simples étant envoyée sur la base des fonctions quasi-rubans F_I.

L'algèbre de Hopf duale de $\mathcal{G}$ que l'on notera

$$\mathcal{K} := \bigoplus_{n\in\mathbb{N}} K_0(A_n) \tag{1.52}$$

est obtenue par une construction similaire. On se restreint à la catégorie des modules dits *projectifs* qui apparaissent comme facteurs directs des modules libres $P \oplus Q = A_n^{\oplus k}$. On compte alors la multiplicité des modules projectifs indécomposables dans un module projectif quelconque. L'induction et la restriction d'un module projectif est encore projectif, on peut donc définir un produit et un coproduit comme précédemment.

Le quotient par le radical $P_i \to S_i := P_i/\operatorname{Rad}(P_i)$ définit enfin une bijection naturelle entre les projectifs indécomposables et les modules simples, qui est encodée par la dualité :

$$\langle P_i | S_j \rangle = \delta_{i,j} \,. \tag{1.53}$$

Pour finir, si l'on compte la multiplicité des modules simples dans les projectifs indécomposables, on trouve naturellement un morphisme de Hopf de $\mathcal{K}$ dans $\mathcal{G}$ que l'on appelle *application de Cartan.* Les coefficients $m_{i,j}$ de cette application sont appelés *invariants de Cartan* de la tour A_n. Ce sont les coefficients du développement :

$$[P_i] = \sum_j m_{i,j} S_j \,. \tag{1.54}$$

Bien évidement, tous les coefficients définis dans cette section sont des multiplicités et donc des *entiers positifs* ! C'est une mine d'or pour le chercheur en combinatoire, qui va s'efforcer de trouver des formules permettant de les calculer en énumérant des objets combinatoires simples et concrets plutôt que des objets algébriques abstraits !

Chapitre 2

Constructions d'algèbres de Hopf : fonctions symétriques généralisées

2.1 Fonctions quasi-symétriques et généralisations

2.1.1 Actions quasi-symétrisantes

L'un des thèmes récurrents de mon travail est l'algèbre de Hopf des fonctions quasi-symétriques. Le premier résultat que j'ai obtenu sur ce sujet est en lui-même très simple mais fécond en conséquences : il s'agit d'une réécriture de la définition des fonctions quasi-symétriques en termes d'une action du groupe symétrique :

Théorème 2.1 *[H5]*

Il existe une action du groupe symétrique sur les polynômes, et même de son algèbre de Hecke, telle que les fonctions quasi-symétriques sont exactement les polynômes invariants pour cette action. Cette action est dite quasi-symétrisante.

On a ainsi un parallèle entre la définition des fonctions quasi-symétriques et celle des fonctions symétriques usuelles.

Outre les interprétations détaillées dans le chapitre 3, cette construction permet une compréhension plus fine de la structure des algèbres des fonctions quasi-symétriques et des fonctions symétriques non commutatives. En particulier, elle permet de construire des bases de ces algèbres :

Théorème 2.2 *[H5]*

Soient $\boldsymbol{\pi}_\omega$ et $\square_\omega$ les opérateurs de symétrisation maximaux de $H_n(0)$ et $H_n(q)$ dans l'action quasi-symétrisante. Alors les éléments $\boldsymbol{\pi}_\omega(X^I)$ et $\boldsymbol{\pi}_\omega(X^I)$ où I parcourt l'ensemble des compositions forment une base de l'algèbre des fonctions quasi-symétriques.

Par dualité, on obtient donc des bases des fonctions symétriques non commutatives. La première base était déjà connue par Gessel [Ge] sous le nom de base fondamentale ou base des quasi-rubans. Rappelons que cette base est l'analogue quasi-symétrique des fonctions de Schur.

La deuxième base, qui interpole entre la base des fonctions monomiales et la base fondamentale, est donc un analogue de la base des fonctions de Hall-Littlewood qui réalise l'algèbre de Hall dans les fonctions symétriques (*cf.* [Mcd]). L'existence d'un analogue de cette base dans les fonctions symétriques non commutatives semble être un indice qu'il existe une algèbre de Hall non commutative. Pour le moment, l'existence d'un tel objet est encore conjecturale : nous n'avons pas réussi par des moyens algébriques ou géométriques à construire une algèbre qui pourrait se réaliser dans les fonctions symétriques non commutatives au moyen de cette base.

Nous avons également défini des analogues à deux paramètres qui semblent jouer le rôle des fonctions de Macdonald (*cf.* [H17]). Signalons que de tels analogues ont aussi été définis par Bergeron et Zabrocki [BZ] en utilisant une autre approche. Bien que les bases qu'ils ont obtenues ne soient pas les mêmes que les nôtres, toutes deux sont très proches et vérifient une même propriété de factorisation quand q est spécialisé à une racine de l'unité, propriété connue dans le cas classique [LLT1, LLT2] (voir aussi [Mcd, BP]) :

Théorème 2.3 *[H5]*

*Soient k un entier et ζ une racine primitive k-*ieme* de l'unité. Soient $J_1, \ldots, J_c$ des compositions de longueur k et J_{c+1} une composition de longueur $r < k$; soit I la concaténation des compositions $J_1 \cdots J_{c+1}$. Alors la fonction de Hall-Littlewood non commutative $H_I(A;\zeta)$ se factorise de la manière suivante :*

$$H_I(A;\zeta) = H_{J_1}(A;\zeta) H_{J_2}(A;\zeta) \cdots H_{J_{c+1}}(A;\zeta)\,. \tag{2.1}$$

Il existe d'autres coïncidences étonnantes. Par exemple, les actions quasi-symétrisantes décrites ci-dessus sont en fait des actions d'un quotient de l'algèbre de Hecke nommé algèbre de Temperley-Lieb [H5]. Cette dernière a pour dimension le nombre de Catalan C_n qui est aussi, d'après [AB, ABB], la dimension du quotient de l'anneau des polynômes par les fonctions quasi-symétriques. Pourtant, on ne peut pas déduire une propriété de l'autre, car les actions quasi-symétrisantes ne sont pas compatibles avec la structure de produit de l'anneau des polynômes.

Pour mieux comprendre ce phénomène, j'ai essayé de voir si l'on pouvait trouver d'autres situations analogues. J'ai ainsi construit une généralisation de ces actions qui permet de montrer le théorème suivant :

Théorème 2.4 *[H19]*

Il existe une suite infinie d'algèbres de Hopf graduées emboîtées qui interpole entre les fonctions symétriques et les fonctions quasi-symétriques.

$$\mathrm{QSym} = \mathrm{QSym}^1 \supset \cdots \supset \mathrm{QSym}^r \supset \cdots \supset \mathrm{QSym}^\infty = \mathit{Sym}\,. \tag{2.2}$$

D'après un résultat de Garcia et Wallach [GW], les fonctions quasi-symétriques forment un module libre sur les fonctions symétriques. Je conjecture que la même propriété reste vraie pour les fonctions QSym^r ci-dessus. En revanche, le quotient de l'anneau des polynômes par QSym^r ne semble pas avoir de structure intéressante.

2.1.2 Fonctions quasi-symétriques généralisées

L'un des intérêts de la construction précédente est qu'elle peut se généraliser à beaucoup d'autres objets combinatoires. On obtient ainsi très simplement plusieurs algèbres de Hopf combinatoires :

Théorème 2.5 *[H4, H10]*
L'ensemble des polynômes non commutatifs invariants par l'action quasi-symétrisante forme une algèbre de Hopf **WQSym***.*
L'ensemble des produits tensoriels de monômes non nuls et qui sont invariants par l'action quasi-symétrisante forment une algèbre de Hopf **MQSym** *dont les bases sont indexées par les matrices d'entiers sans lignes ni colonnes nulles.*

Ces matrices d'entiers sont en bijection naturelle avec les doubles classes du groupe symétrique par des sous-groupes de Young. On peut ainsi les voir comme des permutations généralisées.

Ce dernier point de vue permet de démontrer un théorème analogue de la dualité de Schur-Weyl pour le groupe symétrique. Si $\mathcal{H}$ est une cogèbre connexe et $\mathcal{H}^+$ l'idéal d'augmentation, il existe une structure d'algèbre de Hopf naturelle sur l'algèbre tensorielle $T(\mathcal{H}^+)$. On peut alors montrer que :

Théorème 2.6 *[H10]*
Pour toute cogèbre $\mathcal{H}$, il existe un morphisme canonique de **MQSym** *dans l'algèbre de convolution de $T(\mathcal{H}^+)$.*

Dans le cas de la cogèbre $S(U)$, ce résultat peut être raffiné :

Théorème 2.7 *[H10]*
Soit U un espace vectoriel de dimension finie N. Soit ρ le morphisme canonique de **MQSym** *dans* $\mathrm{Convol}(T(S^+(U)))$*. Pour tout entier n, l'image par ρ de la composante homogène de degré n de* **MQSym** *est exactement le commutant de $GL(U)$ dans la composante homogène de degré n de* $\mathrm{End}(T(S^+(U)))$*. De plus, ρ_U est injectif sur les composantes* $\mathrm{End}^n(T(S^+(U))$ *de degré n inférieur à N.*

Ainsi, les éléments de **MQSym** peuvent s'interpréter comme des endomorphismes de $T(S^+(U))$ et donc se composer. L'algèbre de Hopf **MQSym** est donc munie d'une autre structure de produit dite interne, à l'intérieur des composantes homogènes.

2.2 Formule de Cauchy libre et monoïdes plaxiques généralisés

Le but de cette section est de présenter une méthode générale pour construire des algèbres de Hopf combinatoires. Cette méthode repose sur les réalisations en terme de polynôme non commutatif.

2.2.1 Fonctions quasi-symétriques libres et formule de Cauchy libre

L'ingrédient fondamental est la définition suivante :

Définition 2.8 *[H10] Le* quasi-ruban libre $\mathbf{F}_\sigma$ *indexé par la permutation* $\sigma \in \mathfrak{S}_n$ *est le polynôme non commutatif*

$$\mathbf{F}_\sigma = \sum_{\mathrm{Std}(w)=\sigma^{-1}} w \quad \in \mathbb{Z}\langle A\rangle\,, \tag{2.3}$$

où $\mathrm{Std}(w)$ *désigne le standardisé du mot* w.

On a alors le théorème suivant :

Théorème 2.9 *[H10]*
Les $\mathbf{F}_\sigma$ *forment la base d'une sous-algèbre* **FQSym** *de l'algèbre libre que l'on peut munir naturellement d'une structure d'algèbre de Hopf graduée connexe. Cette algèbre est isomorphe à son dual pour le produit scalaire*

$$\langle \mathbf{F}_\sigma | \mathbf{F}_\tau \rangle = \delta_{\sigma,\tau^{-1}}\,. \tag{2.4}$$

Cette algèbre de Hopf avait déjà été construite sans réalisation par Malvenuto et Reutenauer [MalR]. La dualité dans les fonctions symétriques usuelles peut se traduire par la formule de Cauchy qui est en fait une factorisation de l'identité de Sym dans $\mathrm{Sym}^* \otimes \mathrm{Sym}$. Dans notre cas, on a également un analogue de la formule de Cauchy :

Théorème 2.10 *[H21]*
Il existe un morphisme naturel f *de* $\mathbb{K}[x_{i,j}]$ *dans* $\mathbb{K}\langle X\rangle \otimes \mathbb{K}\langle X\rangle$ *tel que*

$$f\left(\prod_{i,j} \frac{1}{1-x_{ij}}\right) = \sum_{\mathrm{Std}(v)=\mathrm{Std}(u)^{-1}} u \otimes v = \sum_\sigma \mathbf{F}_\sigma \otimes \mathbf{F}_{\sigma^{-1}} = \sum_\sigma \mathbf{P}_\sigma \otimes \mathbf{Q}_\sigma \tag{2.5}$$

pour toute paire de bases duales $\mathbf{P}_\sigma, \mathbf{Q}_\sigma$ *de* **MQSym***.*

En particulier, si l'on décide de plonger cette formule dans $\mathbb{K}[X] \otimes \mathbb{K}\langle X\rangle$, on obtient une formule de Cauchy qui décrit la dualité entre QSym et **Sym**. C'est l'idée qui permet de construire une famille générale de quotients et de sous-algèbres de **FQSym** quand l'on se place dans $\mathbb{K}[M]$ où M est un monoïde approprié.

2.2.2 Analogue du monoïde plaxique

Depuis son étude en 1981 par Lascoux et Schützenberger [LS1], le monoïde plaxique semblait être un objet unique. Plus récemment Krob et Thibon [KT2, KT3] ont défini un quotient du monoïde plaxique nommé pour cette raison hypoplaxique. Nous donnons ici une classe générale de monoïdes qui partage de très nombreuses propriétés combinatoires avec le monoïde plaxique.

On considère les mot w sur un alphabet A totalement ordonnée. On note $|w|$ l'image commutative d'un mot w et $\mathrm{Std}(w)$ son standardisé (*cf.* [H10]).

Définition 2.11 *Un quotient M du monoïde libre est dit compatible avec la standardisation si et seulement si $w \equiv w'$ est équivalent à $|w| = |w'|$ et $\mathrm{Std}(w) = \mathrm{Std}(w')$.*

Un quotient M du monoïde libre est dit compatible avec la restriction aux intervalles si et seulement si, pour tout intervalle de l'alphabet I, on a $w \equiv w'$ implique $w|_I \equiv w'|_I$, où $w|_I$ désigne le mot w où l'on a effacé toutes les lettres qui ne sont pas dans I.

On appellera bon monoïde un monoïde qui vérifie ces deux conditions.

Il existe une très grande variété de monoïdes qui vérifient ces deux conditions et nous n'avons pas pu en donner une classification. Cependant cette définition présente un intérêt certain, comme le montrent les résultats qui suivent. Le premier théorème, dont nous donnons une preuve dans [H10] dans le cas particulier du monoïde plaxique, construit automatiquement plusieurs algèbres de Hopf déjà connues.

Théorème 2.12 *[H10]*

Soit M un bon monoïde. Alors le quotient $\mathbf{FQSym}(M)$ est un quotient de Hopf de $\mathbf{FQSym}$. Les polynômes $\sum_{\sigma\in C}\mathbf{F}_\sigma$ où C parcourt l'ensemble des classes standards du monoïde engendre la sous-algèbre de Hopf de $\mathbf{FQSym}$ duale.

On obtient ainsi une réalisation d'algèbres de Hopf en tant qu'algèbres de polynômes non commutatifs. Le produit des algèbres, dont l'expression peut être très compliquée, est ainsi obtenu comme le produit de concaténation de certain polynômes non commutatifs bien choisis. Cette réalisation est très efficace pour calculer dans ces algèbres de Hopf.

Nous avons exploré deux exemples intéressants de cette construction. Dans le cas du monoïde plaxique nous construisons ainsi l'algèbre de Hopf des tableaux de Poirier et Reutenauer [P]. Ceci fournit une preuve extrêmement simple de la règle de Littlewood-Richardson pour le produit des fonctions de Schur. Le deuxième exemple est l'algèbre de Loday et Ronco et sera développé dans la prochaine section.

Notons enfin qu'une autre construction générale de sous-algèbres de **FQSym** a été proposée plus récement par Reading [Rea]. Il apparaît que cette construction est un cas particulier de la nôtre, mais les algèbres de Hopf ainsi obtenues ont une propriété supplémentaire : le produit peut se décrire au moyen d'un ordre partiel. Nous pensons cependant que la construction de Reading est trop particulière. En effet, l'algèbre des tableaux de Poirier et Reutenauer dont on sait, par ailleurs, que le produit s'exprime également au moyen d'un ordre, ne rentre pas dans le cadre de Reading.

2.2.3 Algèbre des arbres binaires de Loday et Ronco

Un autre exemple très intéressant est l'algèbre des arbres binaires [LR1, LR2]. On l'obtient à l'aide de la construction précédente à partir du monoïde dit sylvestre défini comme suit :

Définition 2.13 *[H11, H16]*
On dit que les mots w_1 et w_2 sont adjacents pour la congruence sylvestre s'il existe trois mots u, v, w et trois lettres $A \leq B < C$ tels que

$$w_1 = u\, AC\, v\, B\, w \qquad \text{et} \qquad w_2 = u\, CA\, v\, B\, w\,. \tag{2.6}$$

La relation d'équivalence sylvestre est la clôture transitive de cette relation. C'est une congruence (elle est compatible avec la concaténation), et le monoïde sylvestre est le quotient du monoïde libre par cette congruence.

Pour ce monoïde, il existe un analogue de la correspondance de Robinson-Schensted :

Théorème 2.14 *Il existe une bijection entre l'ensemble des mots et l'ensemble des couples (BST, DT) où BST est un arbre binaire de recherche et DT est un arbre décroissant.*

Cette bijection est obtenue au moyen de l'algorithme classique d'insertion dans un arbre binaire de recherche. Elle est illustrée par la figure 2.1.

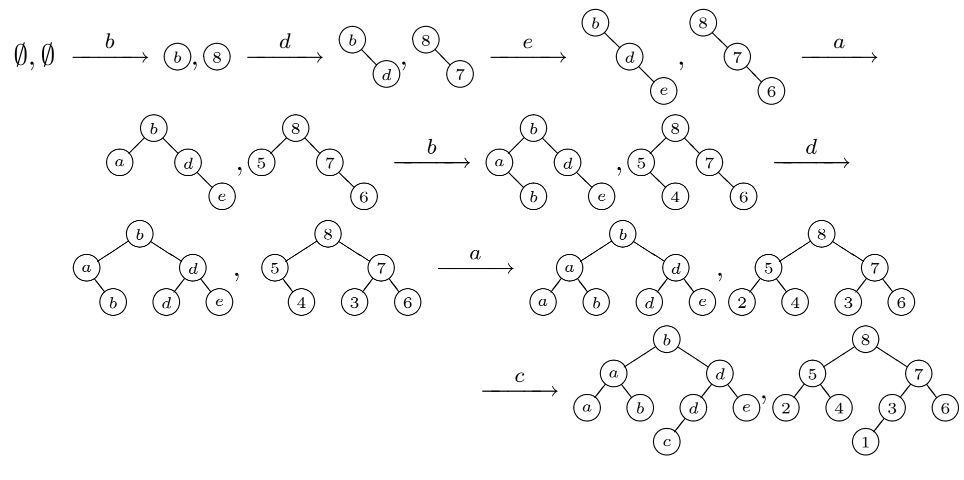


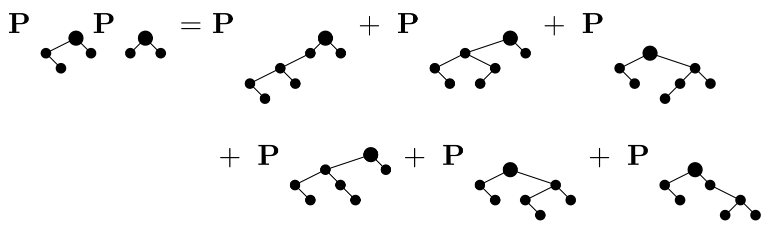
FIG. 2.1 : Insertion sylvestre du mot *cadbaedb*

On voit facilement que le monoïde sylvestre est un bon monoïde. On obtient alors le théorème suivant :

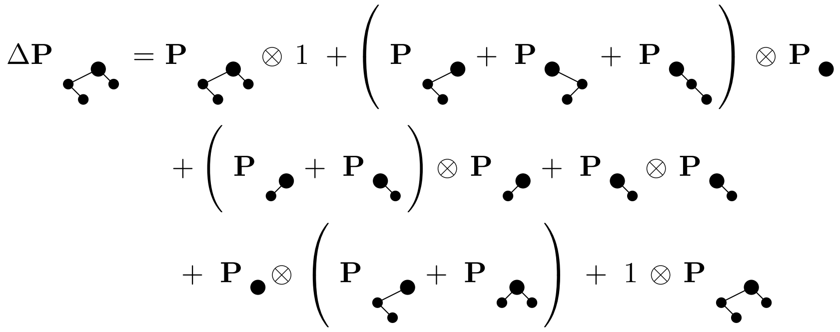
Théorème 2.15 *[H10]*
La sous-algèbre de **FQSym** *associée au monoïde sylvestre est une algèbre de Hopf isomorphe à l'algèbre des arbres binaires de Loday et Ronco.*

La réalisation en termes de polynômes non commutatifs donnée précédemment permet de calculer relativement aisément dans cette algèbre. Rappelons que les produits et coproduits décrits par Loday et Ronco [LR1] sont définis par une double récurrence et peuvent se calculer au moyen d'un ordre sur les arbres [LR2]. Voici un exemple de produit et de coproduit avec la représentation des mots canoniques sylvestres et des arbres :

$$\mathbf{P}_{4213}\mathbf{P}_{312} = \mathbf{P}_{7421356} + \mathbf{P}_{7452136} + \mathbf{P}_{7456213} + \mathbf{P}_{7542136} + \mathbf{P}_{7546213} + \mathbf{P}_{7564213}\,,$$



$$\begin{aligned}\Delta\mathbf{P}_{4213} = {}& \mathbf{P}_{4213}\otimes 1 + (\mathbf{P}_{213}+\mathbf{P}_{231}+\mathbf{P}_{321})\otimes\mathbf{P}_1 + (\mathbf{P}_{12}+\mathbf{P}_{21})\otimes\mathbf{P}_{12}\\ &+\mathbf{P}_{21}\otimes\mathbf{P}_{21}+\mathbf{P}_1\otimes(\mathbf{P}_{213}+\mathbf{P}_{312})+1\otimes\mathbf{P}_{4213}\,,\end{aligned}$$



Le fait de pouvoir calculer aisément nous a permis d'effectuer des changements de bases dans cette algèbre de Hopf. Ces bases peuvent être vues comme des analogues des bases classiques des fonctions symétriques (fonctions élémentaires, fonctions complètes, fonction monomiales). Nous avons ainsi décrit explicitement des bases multiplicatives, ce qui nous permet de montrer de manière constructive le théorème suivant, connu par ailleurs pour des raisons abstraites :

Théorème 2.16 *[H16]*
L'algèbre de Hopf des arbres binaires est une algèbre libre, et autoduale.

Nous avons enfin donné explicitement des bases des éléments primitifs.

Pour conclure cette section, à l'aide des réalisations nous obtenons de manière immédiate le diagramme commutatif de morphismes d'algèbre de Hopf suivant :

$$\begin{tikzcd}
 & \mathbf{FQSym} \arrow[dl, two heads] & \\
\mathbf{PBT}^* \arrow[rr, leftrightarrow, "\approx"] \arrow[d, two heads] & & \mathbf{PBT} \arrow[ul, hook] \\
\mathrm{QSym} & & \mathbf{Sym} \arrow[u, hook] \arrow[dl, two heads] \\
 & \mathrm{Sym} \arrow[ul, hook] &
\end{tikzcd}$$

2.3 Perspectives

Dans ce chapitre, nous donnons différentes constructions d'algèbres de Hopf, qui sont toutes de notre point de vue des généralisations de la construction des fonctions symétriques. En particulier, nous nous sommes attachés à faire apparaître chacune de ces algèbres comme des algèbres de polynômes dont les variables vérifient des relations de commutation variées. Ce point de vue a été relativement peu exploré, il semble pourtant très prometteur. Il permet, en effet, de construire très facilement de nouvelles bases de ces algèbres de Hopf, ainsi que les morphismes qui relient ces algèbres entre elles. Pour des motivations très diverses, une grande variété d'algèbres de Hopf a été récement définie : algèbres de séries génératrices pour les graphes et les ordres partiels, structures d'opérade et mathématique, algèbres de renormalisation en théorie quantique des champs, etc. Je pense que l'application de cette méthode à ces nouvelles algèbres pourrait être très fructueuse.

Une autre piste de recherche est la construction de q-analogues. J'ai montré qu'il en existe de très intéressants dans les fonctions quasi-symétriques. Je pense qu'il en existe dans la plupart de ces algèbres de Hopf. D'ailleurs, la combinatoire des arbres, entre autres, a fait apparaître un certain nombre de q-analogues, sans en faire des structures algébriques. Il me paraît donc très prometteur de chercher à intégrer ces q-analogues combinatoires dans les structures algébriques. Cela éclaircira très probablement les liens entre ces structures et les groupes quantiques.

Chapitre 3

Algèbres de Hopf et représentations

3.1 0-algèbre de Hecke et 0-groupe quantique

Dans le chapitre précédent, j'ai montré comment on pouvait définir une action du groupe symétrique et de son algèbre de Hecke sur les polynômes dont les invariants sont les polynômes quasi-symétriques. Dans le cas usuel des polynômes symétriques, la dualité de Schur-Weyl interprète ceci de deux manières différentes :

- l'algèbre des fonctions symétriques est la somme directe des anneaux de caractères des groupes symétriques ;
- les polynômes symétriques en n variables sont les anneaux de caractères des représentations polynomiales du groupe linéaire GL_n.

Dans le cas des fonctions quasi-symétriques, une théorie analogue a été développée par Krob et Thibon [KT2, KT3]. Il faut alors remplacer le groupe symétrique par l'algèbre de Hecke à $q = 0$ et le groupe linéaire par une version dégénérée de l'algèbre enveloppante quantique $\mathrm{U}_0(gl_N)$. Il est alors extrêmement tentant d'interpréter l'action quasi-symétrisante comme l'action du groupe de Weyl pour le groupe quantique dégénéré. C'est effectivement le cas, dans une certaine mesure.

La principale difficulté vient du fait que $\mathrm{U}_0(gl_N)$ n'est pas semi-simple. Nous nous sommes donc restreints à l'étude des modules simples. Rappelons que la théorie des bases cristallines de Kashiwara encode combinatoirement les représentations de l'algèbre enveloppante quantique $\mathrm{U}_q(gl_n)$. Dans notre cas, q est égal à 0, ce qui rend la construction beaucoup plus simple. Ainsi, dans chaque module simple, il existe une base sur laquelle l'action de $\mathrm{U}_0(gl_N)$ est combinatoire : par les générateurs de Chevalley un vecteur de base est envoyé soit sur un vecteur de base, soit sur zéro. La figure 3.1 illustre l'action de $\mathcal{U}_0(gl_4)$ sur un module simple.

Nos résultats ont alors été les suivants : il existe une notion combinatoire de poids sur les éléments de la base combinatoire dit quasi-cristalline des modules simples de $\mathrm{U}_0(gl_N)$. Pour cette notion de poids, le théorème suivant est l'analogue exact de la formule des caractères de Weyl.

Théorème 3.1 *[H5]*

Soit $\boldsymbol{\pi}_\omega$ l'opérateur de symétrisation maximal de l'action de l'algèbre de Hecke dégénérée $H_n(0)$. Soit K une composition. Le caractère du module irréductible $\mathbf{D}_K$ est donné par la formule

$$\chi(\mathbf{D}_K) = \boldsymbol{\pi}_\omega\, X^K = F_K. \tag{3.1}$$

D'autre part montré qu'il existe aussi une notion de modules de Demazure pour lesquels nous avons une formule des caractères.

Théorème 3.2 *[H5]*

Soit $\mathbf{D}_K$ un $\mathrm{U}_0(gl_N)$-module irréductible. Soient $\sigma \in W$ une permutation, $\xi_\sigma = \boldsymbol{\sigma}\,\xi_K$ le vecteur extrémal associé et $\boldsymbol{\pi}_\sigma$ l'opérateur de symétrisation associé dans l'action de l'algèbre de Hecke dégénérée $H_n(0)$. Alors, le caractère du module de Demazure $\mathrm{U}_0(\mathfrak{b}_+)\,\xi_\sigma$ est donné par la formule

$$\chi\left(\mathrm{U}_0(\mathfrak{b}_+)\,\xi_\sigma\right) = \boldsymbol{\pi}_\sigma\, X^K. \tag{3.2}$$

Ces deux théorèmes, ainsi que la construction d'une version combinatoire non commutative de l'algèbre de Hall décrite au chapitre précédent, prolongent de manière étonnante la liste des similarités qu'il peut y avoir entre la théorie des fonctions symétriques classiques et la théorie des fonctions quasi-symétriques.

3.2 Pics, couleurs et numérologie

Comme l'ont montré Krob et Thibon dans [KT2], le couple d'algèbres de Hopf QSym, **Sym** est le couple des anneaux de Grothendieck de la tour des algèbres de Hecke dégénérées $H_n(0)$. Nous avons tenté de trouver des analogues de cette situation. Il semble ainsi qu'un grand nombre des algèbres de Hopf combinatoires puissent être interprétées comme des anneaux de Grothendieck de tours d'algèbres de dimension finies. Nous avons ainsi obtenu des interprétations de deux algèbres de Hopf.

3.2.1 Algèbre des pics

Les fonctions quasi-symétriques ont été définies par Gessel lors de son étude des descentes des permutations. Plus récemment, l'étude des pics dans les permutations a donné naissance à deux algèbres de Hopf en dualité ; la première, Peak, due à Stembridge [St], est une sous-algèbre de QSym. Comme décrit dans [BMSW], son dual gradué, Peak^*, est naturellement identifié à un quotient de **Sym**. En fait, comme nous l'avons montré dans [H15], l'existence de Peak^* ainsi que nombre de ses propriétés étaient déjà implicites dans [KLT]. On a ainsi montré que Peak est aussi l'image de QSym par un homorphisme naturel obtenu par l'analogue quasi-symétrique des transformations d'alphabet de la théorie des λ-anneaux. Ainsi, Peak^* est aussi une sous-algèbre de **Sym**.

Nous rappelons quelques définitions combinatoires. On appelle *descente* d'une permutation σ de $\mathfrak{S}_n$ tout $i \in \{1 \ldots n-1\}$ tels que $\sigma(i) > \sigma(i+1)$. Par

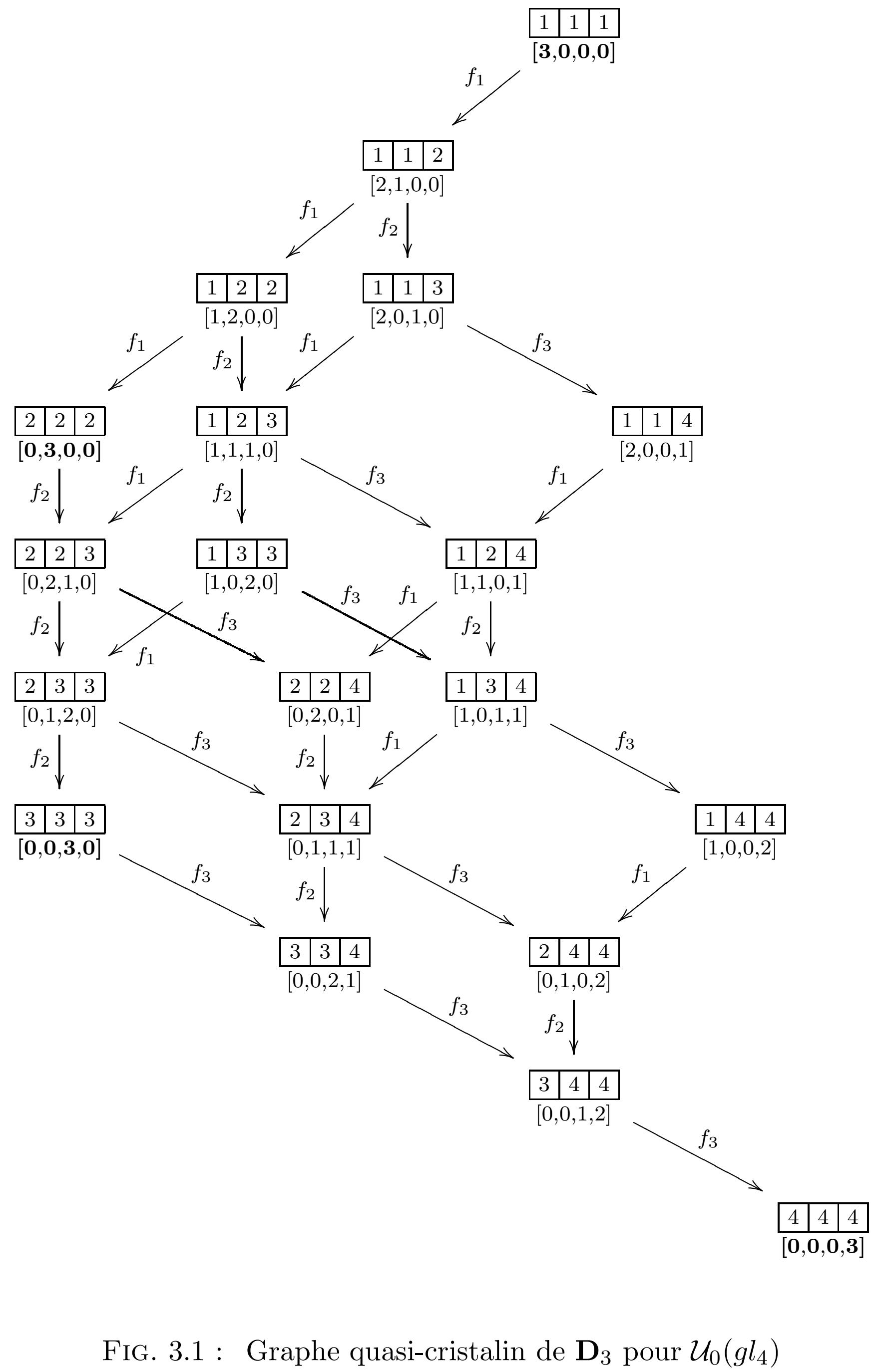


FIG. 3.1 : Graphe quasi-cristalin de $\mathbf{D}_3$ pour $\mathcal{U}_0(gl_4)$

une bijection classique, on peut représenter les ensembles de descentes par des compositions. On appelle *pics* d'une permutation σ de $\mathfrak{S}_n$ tout $i \in \{2 \dots n-1\}$ tel que $\sigma(i) > \sigma(i-1)$ et $\sigma(i) > \sigma(i+1)$. Il est clair que les permutations ayant un ensemble de descentes fixé ont le même ensemble de pics. On peut ainsi parler d'ensemble de pics d'une composition. De plus, les compositions qui codent les ensembles de pics sont appelées *compositions pics* ; ce sont les compositions dont aucune part ne vaut 1 sauf éventuellement la dernière. Peak^* est alors la sous-algèbre de **Sym** engendrée par la somme des fonctions de Schur ruban R_I pour I appartenant à une classe de pics.

Rappelons que la paire (QSym, **Sym**) décrit la théorie des représentations de $H_n(0)$ de la manière suivante : QSym et **Sym** sont respectivement isomorphes à la somme directe des groupes de Grothendieck $G_0(H_n(0))$ et $K_0(H_n(0))$, les produits et coproduits étant donnés par l'induction et la restriction des modules (*cf.* [KT2]). Nous avons trouvé une interprétation similaire pour ($\mathsf{Peak}^*, \mathsf{Peak}$). Cette interprétation est fournie par la dégénérescence à $q = 0$ de la super-algèbre de Hecke-Clifford $HCl_n(q)$ définie par Olshanski [O]. Cette algèbre est un produit tensoriel tordu de l'algèbre de Hecke avec la super-algèbre de Clifford.

Nos résultats sont essentiellement obtenus par induction et restriction depuis l'algèbre de Hecke $H_n(0)$. Ainsi, soit M_I le module induit à $HCl_n(0)$ par le module simple S_I de $H_n(0)$ (on pourra se reporter à [N, KT2] pour une description précise de ces modules). Alors l'action de $H_n(0)$ peut être décrite au moyen d'un ordre partiel sur les remplissages des rubans de forme I dont la figure 3.2 donne un exemple pour $n = 4$ et pour le module M_{211}. La $\mathbb{Z}/2\mathbb{Z}$ graduation de la super-algèbre $HCl_n(0)$ se reflète dans le fait que l'ordre est en fait la somme directe de deux ordres isomorphes, qui correspondent à la parité du nombre de croix du remplissage. Cet ordre nous permet entre autres d'obtenir une filtration du module qui nous donne une suite de composition de Jordan-Hölder.

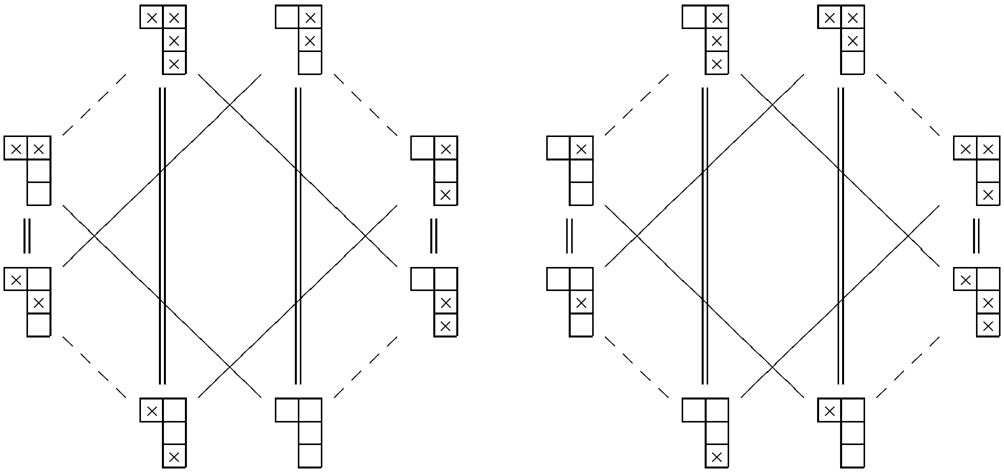


FIG. 3.2 : La filtration du module M_{211}

Nous avons alors obtenu les résultats suivants :

Théorème 3.3 *[H15]*

Les modules induits $M_I := \mathrm{Ind}_{H_n(0)}^{HCl_n(0)}(S_I)$ *et* $M_J := \mathrm{Ind}(S_J)$ *sont isomorphes si et seulement si les ensemble de pics des compositions* I *et* J *coïncident.*

Le module $M_I = \mathrm{Ind}_{H_n(0)}^{HCl_n(0)}(S_I)$ *est la somme directe de plusieurs copies d'un même module simple* $HClS_I$ *pour* $HCl_n(0)$.

L'ensemble des $HClS_I$*, où* I *parcourt l'ensemble des compositions pics, est un système complet de modules simples deux à deux non isomorphes pour* $HCl_n(0)$.

Ayant ainsi décrit la théorie des représentations de $HCl_n(0)$, nous avons une interprétation des algèbres de Hopf $\mathsf{Peak}, \mathsf{Peak}^*$:

Théorème 3.4 *[H15]*

Les algèbres de Hopf $\mathsf{Peak}, \mathsf{Peak}^*$ *sont respectivement isomorphes à la somme directe des groupes de Grothendieck* $G_0(HCl_n(0))$ *et* $K_0(HCl_n(0))$*, les produits et coproduits étant donnés par l'induction et la restriction des modules.*

Les morphismes et inclusions d'algèbre de Hopf de $\mathsf{Peak} \subset \mathrm{QSym}$ *d'une part, et de* $\mathsf{Peak}^* \subset \mathbf{Sym}$ *d'autre part s'interprètent comme la restrition et l'induction des modules entre* $H_n(0)$ *et* $HCl_n(0)$.

Rappelons que d'après Olshanski [O], les caractères de l'algèbre de Hecke-Clifford générique sont codés par l'algèbre $\mathcal{P}$ des fonctions p_λ de Schur. Cette remarque nous a permis de calculer les matrices de décomposition des modules simples de $HCl_n(q)$ quand $q = 0$. Nous obtenons ainsi le diagramme de morphismes d'algèbres de Hopf suivants où tous les parallélogrammes commutent. Les bases qui correspondent aux modules simples et projectifs indécomposables sont notées entre parenthèses.

$$\begin{tikzcd}
\mathrm{QSym}\ (F_I) \arrow[d, two heads, shift left] & & & & \mathbf{Sym}\ (R_I) \arrow[dll, two heads] \arrow[d, two heads, shift left] \\
\mathsf{Peak}\ (\Theta_I) \arrow[u, hook, shift left] & & \mathrm{Sym}\ (s_\lambda) \arrow[ull, hook] \arrow[d, two heads, shift left] & & \mathsf{Peak}^*\ (\Pi_I) \arrow[u, hook, shift left] \arrow[dll, two heads] \\
& & \mathcal{P}\ (P_\lambda) \arrow[ull, hook] \arrow[u, hook, shift left] & &
\end{tikzcd}$$

À titre d'exemple nous donnons les premières matrices de décomposition. Remarquons que pour $n = 2$ et 3 l'algèbre de Hecke-Clifford est semi-simple.

$$\begin{array}{rl} & \begin{array}{cc} 3 & 21 \end{array} \\ \begin{array}{r} 3 \\ 21 \end{array} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{rl} & \begin{array}{ccc} 4 & 31 & 22 \end{array} \\ \begin{array}{r} 4 \\ 31 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{rl} & \begin{array}{ccccc} 3 & 41 & 32 & 23 & 221 \end{array} \\ \begin{array}{r} 5 \\ 41 \\ 32 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

$$
\begin{array}{rl}
 & \begin{array}{cccccccc} \rotatebox{90}{6} & \rotatebox{90}{51} & \rotatebox{90}{42} & \rotatebox{90}{33} & \rotatebox{90}{321} & \rotatebox{90}{24} & \rotatebox{90}{231} & \rotatebox{90}{222} \end{array} \\
\begin{array}{r} 6 \\ 51 \\ 42 \\ 321 \end{array} &
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$

$$
\begin{array}{rl}
 & \begin{array}{ccccccccccccc} \rotatebox{90}{7} & \rotatebox{90}{61} & \rotatebox{90}{52} & \rotatebox{90}{43} & \rotatebox{90}{421} & \rotatebox{90}{34} & \rotatebox{90}{331} & \rotatebox{90}{322} & \rotatebox{90}{25} & \rotatebox{90}{241} & \rotatebox{90}{232} & \rotatebox{90}{223} & \rotatebox{90}{2221} \end{array} \\
\begin{array}{r} 7 \\ 61 \\ 52 \\ 43 \\ 421 \end{array} &
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 2 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 1 & 1 & 2
\end{bmatrix}
\end{array}
$$

3.2.2 Algèbres de Mantaci-Reutenauer-Poirier

Nous avons obtenu des résultats très similaires aux précédents pour une autre paire d'algèbres de Hopf en dualité : l'algèbre des descentes $\mathbf{MR}^r$ de Mantaci et Reutenauer [ManR] associée au produit en couronne de $\mathfrak{S}_n$ par un groupe cyclique, et sa duale QMR^r introduite par Poirier [P]. Ces algèbres ont plus récemment été étudiées par Baumann et Hohlweg [BH].

Comme on pouvait le deviner, l'interprétation est fournie par une dégénérescence à $q = 0$ des algèbres de Hecke des produits en couronne définies par Ariki et Koike [Ari, AK]. La difficulté ici a été de trouver la bonne manière de dégénérer q à 0. En effet, la présentation originelle d'Ariki et Koike ainsi que d'autre présentations [KLLO] ne fournissent pas le résultat désiré. Il faut utiliser une autre présentation due à Shoji [SakS, SawS, Sho]. Pour ceci nous avons fait appel à des calculs expérimentaux qui seront évoqués dans le prochain chapitre. Le problème était le suivant : étant supposé connus

- le nombre et la dimension des modules simples ;
- le nombre et la dimension des modules projectifs indécomposables ;
- la multiplicité des modules simples dans les projectifs indécomposables ;

trouver la bonne présentation. Précisons que pour discriminer entre les différentes possibilités, nous avons dû calculer expérimentalement toutes ces données pour $\mathcal{H}_{3,3}(0)$ qui est de dimension $3^3 3! = 162$ ainsi que $\mathcal{H}_{4,2}(0)$ qui est de dimension $2^4 4! = 384$. Une fois la bonne algèbre identifiée, la même stratégie que celle de la section précédente fonctionne.

Les principaux résultats sont les suivants :

Théorème 3.5 *[H15]*

Les modules simples de $\mathcal{H}_{n,r}(0)$ sont indexés par les compositions colorées de n, c'est-à-dire les paires (I, w), où I est une composition de n et c un mot de même longueur que I sur l'alphabet $\{1, \ldots, r\}$.

Théorème 3.6 *[H15]*

Les algèbres de Hopf QMR^r *de Poirier et* $\mathbf{MR}^r$ *de Mantaci-Reutenauer sont respectivement isomorphes à la somme directe des groupe de Grothendieck $G_0(\mathcal{H}_{n,r}(0))$ et $K_0(\mathcal{H}_{n,r}(0))$, les produits et coproduits étant donnés par l'induction et la restriction des modules.*

Les morphismes et inclusions d'algèbres de Hopf de $\mathrm{QSym} \subset \mathrm{QMR}^r$ *d'une part, et de* $\mathbf{Sym} \subset \mathbf{MR}^r$ *d'autre part, s'interprètent comme la restriction et l'induction des modules entre $H_n(0)$ et $\mathcal{H}_{n,r}(0)$.*

On obtient ainsi le diagramme de morphismes d'algèbres de Hopf suivant où tous les parallélogrammes commutent :

$$\begin{tikzcd}
\mathrm{QMR}^{(l)}\ (F_{I,c}) \arrow[d, shift left, ->>] & & \mathbf{MR}^{(l)}\ (R_{I,c}) \arrow[dl, ->>] \arrow[d, shift left, ->>] \\
\mathrm{QSym}\ (F_I) \arrow[u, shift left, hook] & \mathrm{Sym}^{\otimes l}\ (\otimes s_\lambda) \arrow[ul, hook] \arrow[d, shift left, ->>] & \mathbf{Sym}\ (R_I) \arrow[u, shift left, hook] \arrow[dl, ->>] \\
& \mathrm{Sym}\ (s_\lambda) \arrow[ul, hook] \arrow[u, shift left, hook] &
\end{tikzcd}$$

3.3 Perspectives

Comme nous l'avons vu précédemment, la théorie des représentations des tours d'algèbres est un moyen très fécond pour construire et interpréter les algèbres de Hopf combinatoires. Je propose ici deux pistes de recherche.

Dans [H12], nous conjecturons l'existence d'une tour d'algèbres dont la théorie des représentations serait encodée par l'algèbre des arbres binaires de Loday et Ronco. Des calculs explicites nous ont permis de construire de telles algèbres jusqu'à $n = 6$ mais sans pouvoir trouver de règle générale. Il serait très intéressant d'expliciter une telle tour d'algèbres.

D'autre part, le groupe symétrique $\mathfrak{S}_n$ et son algèbre de Hecke $H_n(0)$ peuvent tous deux être vus comme des algèbres d'opérateurs agissant sur l'espace vectoriel V_n des permutations. Soit A_n la sous-algèbre de $\mathrm{End}(V_n)$ engendrée par ces deux algèbres. A_n est par conséquent une sur-algèbre de $\mathbb{K}[\mathfrak{S}_n]$ et de l'algèbre de Hecke $H_n(0)$. Il est très facile de voir que les A_n forment aussi une tour d'algèbres associative et inductive. Les anneaux de Grothendieck de ces deux tours d'algèbres sont nécessairement liés aux fonctions symétriques commutatives et non commutatives. Dans un travail en cours avec N. Thiéry, nous étudions ces tours d'algèbres. Nous avons déjà montré le résultat suivant :

Théorème 3.7 *[H20]*
La théorie des représentations de A_n est la même que la théorie des représentations de l'algèbre d'incidence $K[Bool_{n-1}]$ du treillis booléen des sous-ensembles de $\{1, \dots n-1\}$. Plus précisément, ces deux algèbres sont Morita-équivalentes.

On rappelle que l'algèbre d'incidence d'un ordre partiel est l'algèbre de la catégorie de cet ordre partiel. D'autre part, il est apparu que les modules simples sur cette algèbre sont aussi reliés à une certaine catégorie de modules sur l'algèbre de Hecke affine. Pour le moment nous n'avons pas encore de résultats précis sur ce point.

Enfin, en s'appuyant sur les travaux de Vershik et Kerov, l'école russe a étudié dans les cas semi-simples les représentations des limites de ces tours d'algèbres et, en particulier, le cas du groupe symétrique. La même théorie doit être faisable dans le cas non semi-simple. Une telle théorie est clairement reliée à l'étude asymptotique des objets combinatoires sous-jacents et est, par conséquent, très intéressante tant du point de vue combinatoire que du point de vue algébrique.

Chapitre 4

Calcul symbolique et combinatoire algébrique : une approche expérimentale

Depuis le printemps 2001, je coordonne en collaboration avec N. Thiéry le développement de `MuPAD-Combinat`. Il s'agit d'une plate-forme d'expérimentation pour la recherche en combinatoire algébrique. Nous avons présenté ce logiciel ainsi que les choix de conception dans l'article [H14].

`MuPAD-Combinat` est une bibliothèque libre (sous licence LGPL) pour le système de calcul formel MuPAD. Il contient actuellement des fonctions pour manipuler les classes d'objets combinatoires usuels (partitions, tableaux, arbres, graphes, classes décomposables définies par des grammaires, etc), les polynômes de Schubert, les groupes symétriques et sous-groupes ainsi que les automates à poids. Nous donnons aussi des outils pour construire de nouveaux objets algébriques et comme application nous donnons, entre autres, des implantations des fonctions symétriques et de leurs généralisations ainsi que l'algèbre des arbres binaires. La plupart de mes travaux récents ont utilisé intensivement ce logiciel. Notons que les applications ne se limitent pas à la combinatoire algébrique mais s'étendent également à la complexité (*cf.* [MarMo1, MarMo2]) et à la vérification automatique de programmes [DGG, Gou].

Le code représente actuellement environ 80 000 lignes de code MuPAD, 3 000 lignes de code C++. Il inclut 50 000 lignes de code C/C++ externe ainsi que quelques 450 pages de documentation. Outre les trois développeurs principaux N. Thiéry, T. Gomez-Diaz et moi-même, nous avons une vingtaine de contributeurs dont les principaux sont É. Laugerotte (maître de conférences, LIFAR, Université de Rouen), J.-C. Novelli (chargé de recherches CNRS, IGM, Université de Marne-la-Vallée), F. Descouens (doctorant, IGM, Université de Marne-La-Vallée), X. Molinero (professeur, Universitat Politécnica de Catalunya, Espagne) et H. Abbad (doctorante, Djillali Liabes, Université de Sidi Bel Abbes, Algérie). Le cœur de `MuPAD-Combinat` est inclus dans la bibliothèque officielle de MuPAD depuis la version 2.5.0.

La bibliothèque est librement accessible sur

`http ://mupad-combinat.sourceforge.net/`

où l'on peut également trouver toute la documentation en ligne. Une introduction rapide se trouve d'autre part dans [H14]. Précisons enfin que du 28 juin au 3 juillet 2004 ont eu lieu à Rouen les premières rencontres des développeurs de `MuPAD-Combinat` qui ont réuni une vingtaine de personnes.

4.1 Génération automatique d'objets et manipulations combinatoires

4.1.1 Énumération d'objets simples

Dans cette section, nous montrons brièvement quelques manipulations d'objets combinatoires ainsi que les possibilités d'énumération offertes par `MuPAD-Combinat`.

Dans la suite, on appelle *classe combinatoire* une collection d'objets ainsi que la bibliothèque de fonctions qui permettent de les manipuler en MuPAD. Les différentes classes combinatoires ont une interface commune qui comporte des fonctions d'énumération, de comptage et de génération. On calcule la liste des permutations de $\mathfrak{S}_3$ par la commande

```
>> permutations::list(3)

 [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2],

    [3, 2, 1]]
```

De manière similaire, on peut calculer la liste des tableaux de forme (221) :

```
>> tableaux::list([2, 2, 1])

   -- +---+      +---+      +---+      +---+      +---+      --
   |  | 5 |      | 5 |      | 4 |      | 4 |      | 3 |       |
   |  +---+---+  +---+---+  +---+---+  +---+---+  +---+---+  |
   |  | 3 | 4 |  | 2 | 4 |  | 3 | 5 |  | 2 | 5 |  | 2 | 5 |  |
   |  +---+---+  +---+---+  +---+---+  +---+---+  +---+---+  |
   |  | 1 | 2 |, | 1 | 3 |, | 1 | 2 |, | 1 | 3 |, | 1 | 4 |  |
   -- +---+---+  +---+---+  +---+---+  +---+---+  +---+---+ --
```

ou des arbres binaires de taille 4 :

```
>> binaryTrees::list(4)

 -- o   , o  , o  , o , o ,  o  ,  o ,  o ,   o ,  o,  o,   o,
 |   \     \    \    \   \  / \   / \  / \   / \  /   /    /
 |    \     \   /\   /   /     \    /  \    /     \   \   /\
 --    \    /        \  /                          \  /

      o,    o --
     /     /   |
    /     /    |
    \    /    --
```

Dans la plupart des cas, nous fournissons des fonctions de génération : on appelle générateur une fonction qui retourne un à un, lors des appels consécutifs, les éléments d'une classe combinatoire. Ce concept très proche de la notion d'itérateur du C++ permet de parcourir les listes d'objets de grande classe combinatoire sans stocker cette liste en mémoire. Par exemple, nous construisons un générateur pour les 204 226 partitions de 50 :

```
>>   g := partitions::generator(50)

   proc fromNextGenerator() ... end
```

Voici les 4 premières :

```
>>   g(), g(), g(), g()
```

```
[50]

[49, 1]

[48, 2]

[48, 1, 1]
```

Ce procédé est particulièrement utile dans le cas d'objets récursifs ou de gros produit cartésien. Il permet non seulement d'économiser la mémoire, mais il évite le brassage de cette dernière lors de l'application répétée de la fonction `map` sur des listes. Pour cette raison, et malgré le surcoût engendré par les calculs que l'on doit refaire plusieurs fois, nous parvenons à gagner également en vitesse, y compris pour le calcul de la liste des éléments d'une classe. C'est le cas, par exemple, pour les matrices d'entiers dont on fixe la somme des lignes et des colonnes.

4.1.2 Énumération d'objets récursifs

Pendant la phase d'expérimentation en recherche, on se heurte souvent au problème d'avoir à utiliser une classe d'objets combinatoires qui n'est pas déjà connue du système. Pour résoudre ce problème, nous avons intégré et étendu la bibliothèque `combstruct` développée par S. Corteel, A. Denise, I. Dutour, F. Sarron et P. Zimmermann. Cette bibliothèque permet l'énumération d'une classe d'objets décrite par une grammaire récursive déterministe. À l'aide de ce procédé, nous énumérons une très grande variété d'objets tels que les partitions d'ensembles, les arbres et, grâce à un ajout récent, une grande variété de classes de permutations. Le lecteur se reportera à la page de manuel correspondante pour des explications détaillées de la syntaxe.

Nous considérons les mots dits de Fibonacci sur les lettres `A` et `B` qui ne contiennent pas deux `B` consécutifs. Un tel mot est

- soit vide ;
- soit le mot `B` ;
- soit un mot se terminant par `A` ;

– soit un mot se terminant par AB.

Dans les deux derniers cas, le mot restant est encore un mot de Fibonacci. Cette remarque peut être utilisée pour construire une grammaire non ambiguë. En MuPAD :

```
>> fiWords := decomposableObjects(
       [FiWords = Union(Epsilon,
                        Atom(B),
                        Prod(FiWords, Atom(A)),
                        Prod(FiWords, Atom(A), Atom(B)))
       ]):
```

Note : un `Epsilon` est un objet de taille 0 et `Atom` est un objet de taille 1. On peut alors calculer les objets de taille 4 décrits par cette grammaire.

```
>> fiWords::list(4)

 [Prod(Prod(Prod(B, A), A), A), Prod(

    Prod(Prod(Prod(Epsilon, A), A), A), A),

    Prod(Prod(Prod(Epsilon, A, B), A), A),

    Prod(Prod(B, A, B), A), Prod(Prod(Prod(Epsilon, A), A, B), A

    ), Prod(Prod(B, A), A, B), Prod(Prod(Prod(Epsilon, A), A),

    A, B), Prod(Prod(Epsilon, A, B), A, B)]
```

Ce n'est pas très lisible car on a la trace de l'arbre de génération des objets. Dans ce cas, `MuPAD-Combinat` offre la possibilité de donner dans la grammaire des règles de réécriture. D'autre part, en réordonnant la grammaire, on peut s'assurer que les mots sont engendrés dans l'ordre lexicographique :

```
>> fiWords := combinat::decomposableObjects(
       [FiWords    = Alias(FiWordsRec, DOM_LIST),
        FiWordsRec = Union(Epsilon(),
            Alias(Prod(Atom(A), FiWordsRec), op),
            Atom(B),
            Alias(Prod(Atom(B), Atom(A), FiWordsRec), op)
        )
       ]):
   fiWords::list(4);

 [[A, A, A, A], [A, A, A, B], [A, A, B, A], [A, B, A, A],

    [A, B, A, B], [B, A, A, A], [B, A, A, B], [B, A, B, A]]
```

Terminons cet exemple en notant que, pendant la génération, MuPAD calcule de manière efficace le nombre de mots à l'aide d'un calcul automatique de la relation de récurrence vérifiée par le nombre de mots. Ce travail est fait par un algorithme que nous avons repris de `combstruct` ; cet algorithme ne s'applique que si la grammaire est une grammaire hors contexte :

```
>> fiWords::recurrenceRelation() = 0

                  u(n - 1) - u(n) + u(n - 2) = 0
```

On reconnaît la relation de récurrence des nombres de Fibonacci. Nous utilisons en interne ce mécanisme dans de nombreux objets combinatoires.

J'ai ajouté récemment des fonctions permettant de construire des objets étiquetés, et en particulier des constructions incluant le marquage et le démarquage d'étiquette. Cela permet une grande variété de constructions telles que les partitions d'ensembles :

```
>> partEns := combinat::decomposableObjects([Bla = Set(Set(Z))],
                                             Labelled = TRUE);

>> partEns::list(3)

 [Set(Set(Z(1)), Set(Z(2)), Set(Z(3))),

   Set(Set(Z(1)), Set(Z(2), Z(3))),

   Set(Set(Z(1), Z(2), Z(3))), Set(Set(Z(1), Z(2)), Set(Z(3))),

   Set(Set(Z(1), Z(3)), Set(Z(2)))]
```

Cette construction déploie toute sa puissance, par exemple, dans l'énumération de certaines classes récursives de permutations. On appelle permutation d'André toute permutation telle que le 1 est avant le 2 et que, lorsque l'on supprime le 1, les parties droite et gauche sont, après renumérotation, des permutations d'André. La grammaire correspondante s'écrit de la manière suivante :

```
>> AndrePermutations := combinat::decomposableObjects([
       AndrePerm = Union(Epsilon, AndPerm),
       AndPerm = Union(Z,
                       Mark(Mark(Prod(
                          Unmark(AndPerm),
                          Unmark(Z),
                          Union(Epsilon, AndPerm)
                       ))))], Labelled);
```

mais il est plus confortable de lire les résultats après réécriture :

```
>> AndrePermutationsMieux := combinat::decomposableObjects([
       AndrePerm = Alias(Union(Epsilon, AndPerm),
                         revert@DOM_LIST@op),
       AndPerm = Union(Node,
```

```
                         Alias(Mark(MultX(Mark(
                            Unmark(AndPerm),
                            Unmark(Node),
                            Union(Alias(Epsilon, ()-> null()),
                                  AndPerm)
                        ))), op)
                        ),
       Node = Alias(Z, op)], Labelled);
```

```
>> AndrePermutationsMieux::list(5)
```

```
 [[5, 3, 4, 1, 2], [3, 4, 5, 1, 2], [1, 4, 5, 2, 3],

    [1, 2, 5, 3, 4], [1, 2, 3, 4, 5], [1, 5, 2, 3, 4],

    [1, 4, 2, 3, 5], [4, 5, 1, 2, 3], [3, 5, 1, 2, 4],

    [3, 4, 1, 2, 5], [5, 1, 4, 2, 3], [5, 1, 2, 3, 4],

    [4, 1, 5, 2, 3], [4, 1, 2, 3, 5], [3, 1, 5, 2, 4],

    [3, 1, 2, 4, 5]]
```

Le nombre de ces permutations donne les coefficients de la série de Taylor de la somme des fonctions trigonométriques tangente et sécante :

```
>> AndrePermutationsMieux::count(i) $ i = 0..12
```

```
 1, 1, 1, 2, 5, 16, 61, 272, 1385, 7936, 50521, 353792, 2702765
```

```
>> AndrePermutationsMieux::count(i)/i! $ i = 0..10;
   coeff(taylor(tan(x)+sec(x), x,11));
```

```
  1, 1, 1/2, 1/3, 5/24, 2/15, 61/720, 17/315, 277/8064, 62/2835
  1, 1, 1/2, 1/3, 5/24, 2/15, 61/720, 17/315, 277/8064, 62/2835
```

Pour développer ces algorithmes, nous travaillons en collaboration étroite avec C. Martinez et X. Molinero [MarMo1, MarMo2]. Grâce à leurs travaux, nous pouvons garantir que la plupart des algorithmes utilisés ont une complexité optimale. Les quelques exceptions sont le plus souvent dues au fait que les objets MuPAD ont une sémantique de copie et que donc il est très difficile de travailler en place. Par conséquent, même si l'utilisation d'un langage interprété ralentit considérablement les calculs, nous pouvons aller relativement loin. En particulier, ce genre de calculs est utilisé pour la vérification automatique de programmes. Dans ce cadre, les grammaires décrivent les exécutions possibles du programme à vérifier. On utilise alors ce système pour engendrer, dans les petits cas, toutes les exécutions possibles. Dans les cas pratiques où cet ensemble est trop gros, on travaille à l'aide d'échantillons aléatoires [Gou, DGG].

4.2 Deux exemples de constructions d'objets algébriques

Du point de vue de ma recherche, l'application principale de l'énumération d'objets combinatoires vient du fait que ces objets indexent naturellement des bases d'objets algébriques. De plus, les structures récursives vérifiées par ces objets combinatoires sont le reflet des structures algébriques.

Pendant la conception de `MuPAD-Combinat`, notre objectif principal a été de permettre d'interfacer le plus facilement possible les constructions combinatoires avec les constructions algébriques. Nous nous sommes particulièrement attachés à résoudre le problème de la linéarité, théoriquement trivial mais qui, à notre avis, n'était pas résolu de manière satisfaisante dans les logiciels existants. Plus précisément, si C est une classe combinatoire, nous voulons manipuler le plus naturellement possible des combinaisons linéaires d'objets de cette classe, c'est-à-dire calculer dans les modules libres dont les bases sont des classes combinatoires. La plupart des structures additionnelles (produits, coproduits) sont définies par linéarité ou bilinéarité à partir de la base. Nous nous sommes efforcés de faciliter ce genre de constructions. Pour illustrer ce point, nous présentons deux exemples.

4.2.1 Application aux empilements

Le premier de ces problèmes, sur lequel j'ai travaillé en collaboration avec Serge Netchaev de l'Université Indépendante de Moscou, est un problème dont l'origine se trouve en physique statistique.

Nous considérons un empilement de dominos de largeur 2 qui sont centrés sur les points entiers de l'intervalle $[1 \dots n]$ (voir figure 4.1). Les dominos tombent aléatoirement avec une distribution uniforme. L'un des grands problèmes est de calculer la vitesse de croissance de l'empilement en supposant que les dominos tombent avec une fréquence régulière. Nous nous sommes intéressés au problème orthogonal de savoir quelle est la configuration des dominos qui sont en haut de l'empilement. Plus précisément,

Définition 4.1 *Soit E un empilement de dominos. On appelle* dominos en haut *de E tout domino d tel que E est égal a l'empilement E/d sur lequel on fait tomber un domino dans la colonne de d.*

On appelle haut de l'empilement *noté $h(E)$ l'ensemble des colonnes sur lesquelles il existe un domino en haut.*

Les configurations possibles des hauts de l'empilement sont donc les sous-ensembles A de $[1 \dots n]$ tels que i et $i+1$ n'appartiennent jamais tous deux à A. Le nombre de ces sous-ensembles est donné par le nombre de Fibonacci. Par conséquent, on appelle un tel sous-ensemble un ensemble de Fibonacci.

On se pose alors le problème suivant :

Problème 4.2 *On fait tomber sur l'empilement des dominos de manière équiprobable dans chaque colonne. À chaque instant, les sous-ensembles de Fi-*

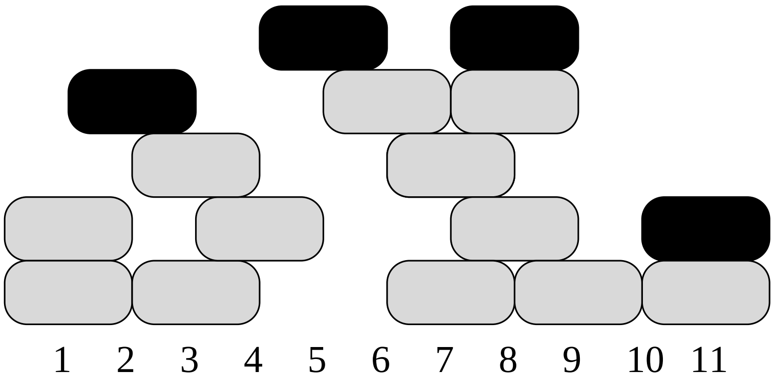


FIG. 4.1 : Le haut d'un empilement de dominos

bonacci A de $[1 \dots n]$ apparaissent avec des probabilités $P_t(A)$. On cherche à calculer la limite de cette distribution de probabilité quand le nombre de dominos du tas tend vers l'infini. On note $P(A)$ et on appelle distribution limite cette mesure de probabilité.

A priori, si l'on excepte la condition aux bords, un ensemble de Fibonacci est la même chose qu'un ensemble de pics. Il semble donc que l'algèbre des pics évoquée au chapitre 3 pourrait fournir une algèbre de séries génératrices adaptée à ce problème.

La première question est donc de calculer expérimentalement cette distribution de probabilité. `MuPAD-Combinat` permet de le faire très simplement : soit $A = h(E)$ le haut de l'empilement E. Alors, si l'on fait tomber un domino dans la colonne i le nouvel empilement a pour haut

$$h(E, i) = h(E)/\{i-1, i+1\} \cup i\,. \tag{4.1}$$

Notons D_i l'application $h(E) \mapsto h(E, i)$ définie ci-dessus. On se place maintenant dans l'espace vectoriel des sous-ensembles de Fibonacci de $\{1, \dots, n\}$. Et l'on note encore D_i l'extension par linéarité de l'application D_i précédente. Il est alors clair que, si à un instant t la mesure de probabilité des hauts est $P_t(A)$, après la chute d'un domino dans la colonne i, la nouvelle mesure de probabilité $P_{t+1}(A)$ est donnée par

$$\sum_A P_{t+1}(A)A = D_i\left(\sum_A P_t(A)A\right)\,. \tag{4.2}$$

Par conséquent, la distribution limite $P(A)$ vérifie

$$\frac{1}{n}\left(\sum_{i=1}^{n} D_i\right)\left(\sum_A P(A)A\right) = \sum_A P(A)A\,. \tag{4.3}$$

Autrement dit, $\sum_A P(A)$ est un vecteur propre pour $\sum D_i/n$ associé à la valeur propre 1. C'est exactement ce que nous codons en MuPAD :

```
>> domain setSpace(N: Type::PosInt)
       category Cat::FreeModule(Dom::ExpressionField());
       inherits Dom::FreeModule(Dom::ExpressionField(),
                    combinat::subClass(combinat::subsets,
                                       Parameters = {$1..N}));

       EiBasis := proc(st : dom::basisIndices, i : Type::PosInt)
       begin
           dom::term((st union {i}) minus {i-1, i+1})
       end_proc;
       Ei := operators::makeLinear(dom::EiBasis,
                                  Source=dom, ImageSet=dom);
   end_domain:
```

Note : pour simplifier, nous considérons l'espace de tous les sous-ensembles et non pas seulement ceux de Fibonacci.

Pour calculer la probabilité, après avoir défini une instance de ce problème pour un n fixé, nous considérons le vecteur dit générique $gen := \sum_A c_A A$ où les c_A sont des coefficients inconnus :

```
>> n := 4: space := setSpace(n):

>> gen := space::genericElement(z -> c[op(z)])

 c[1, 2, 3, 4] B({1, 2, 3, 4}) + c[2, 3, 4] B({2, 3, 4}) +

    c[1, 3, 4] B({1, 3, 4}) + c[1, 2, 4] B({1, 2, 4}) +

    c[1, 2, 3] B({1, 2, 3}) + c[3, 4] B({3, 4}) +

    c[2, 4] B({2, 4}) + c[2, 3] B({2, 3}) + c[1, 4] B({1, 4}) +

    c[1, 3] B({1, 3}) + c[1, 2] B({1, 2}) + c[4] B({4}) +

    c[3] B({3}) + c[2] B({2}) + c[1] B({1}) + c[] B({})
```

Nous définissons l'opérateur $\sum D_i/n$:

```
>> oper := z -> (1/N)*_plus(space::Ei(z, i) $ i = 1..N):
```

Et nous calculons un vecteur propre par identification des coefficients :

```
>> linsolve({coeff(oper(gen) - gen),
            _plus(op(coeff(gen))) - 1})

 [c[] = 0, c[1] = 1/24, c[2] = 1/8, c[3] = 1/8, c[4] = 1/24,

    c[1, 2] = 0, c[1, 3] = 5/24, c[1, 4] = 1/4, c[2, 3] = 0,

    c[2, 4] = 5/24, c[3, 4] = 0, c[1, 2, 3] = 0,
```

```
c[1, 2, 4] = 0, c[1, 3, 4] = 0, c[2, 3, 4] = 0,

c[1, 2, 3, 4] = 0]
```

Il semble que $24 = 4!$ soit dénominateur commun ! Ceci se vérifie facilement pour d'autres valeurs de n. Forts de cette remarque expérimentale, nous avons pu montrer le théorème suivant :

Théorème 4.3 *La probabilité limite est donnée par*

$$P(A) = \#\{\sigma \in \mathfrak{S}_n \mid \mathrm{Pic}_{\mathrm{e}}(\sigma) = A\} \tag{4.4}$$

où $\mathrm{Pic}_{\mathrm{e}}(\sigma)$ *désigne l'ensemble des pics de* σ*, étant entendu que* 1 *et* n *peuvent être des pics.*

Inversement, si l'on décide de modifier la définition du haut du tas pour qu'un domino dans les colonnes 1 et n ne soit pas considéré comme étant en haut, on montre alors le théorème suivant :

Théorème 4.4 *Dans l'algèbre des pics on a*

$$\sum_{A \in \{1 \ldots n\}} P(A)\Theta_A = \frac{1}{n!}(\Theta_{(1)})^n \tag{4.5}$$

autrement dit

$$\sum P(A)\Theta_A t^{\#(A)} = \exp(\Theta_{(1)})\,. \tag{4.6}$$

où Θ_I *est la base fondamentale définie par Stembridge dans [St].*

Par cet exemple, nous avons illustré la manière naturelle avec laquelle il est possible de poser des questions de combinatoire algébrique à `MuPAD-Combinat`. De ce point de vue, c'est un outil très efficace pour aider la recherche en combinatoire algébrique.

4.2.2 Un exemple d'algèbre combinatoire

Dans l'exemple précédent, nous avons défini un espace vectoriel et une application linéaire. `MuPAD-Combinat` offre la possibilité de définir des structures algébriques beaucoup plus riches.

Pour illustrer la possibilité de définir très facilement des algèbres combinatoires, nous donnons ici la définition de l'algèbre de Tits A_n. Cette algèbre a pour base les partitions d'ensemble ordonnées, c'est-à-dire les k-uplets $(A_1, \ldots A_k)$ de sous-ensembles de $\{1 \ldots n\}$ tels que l'ensemble $\{A_1, \ldots A_k\}$ forme une partition de l'ensemble $\{1 \ldots n\}$. La multiplication de deux k-uplets est donnée par

$$(A_1, \ldots A_k)(B_1, \ldots B_l) = V\left(((A_i \cap B_j)_{i=1\ldots k})_{j=1\ldots l}\right) \tag{4.7}$$

où V désigne l'opération qui supprime les composantes vides d'un p-uplet.

L'écriture du programme pour calculer dans cette algèbre ne prend que quelques dizaines de lignes :

```
>> alias(SetPart = combinat::setPartitionsOrdered);
   domain TitsAlgebra(N : Type::NonNegInt,
                      R = Dom::ExpressionField())

       inherits Dom::FreeModule(R,
                    combinat::subClass(SetPart, Parameters=N));
       category Cat::FiniteDimensionalAlgebraWithBasis(R);

       oneBasis := [{$1..N}];

       mult2Basis :=
       proc(part1 : dom::basisIndices,
             part2 : dom::basisIndices)
           local p1, p2;
       begin
           dom::term(select([_intersect(p1, p2)
                             $ p1 in part1 $ p2 in part2],
                            not _equal, {}));
       end_proc;

   end_domain:
```

Le code décrit les propriétés mathématiques de manière naturelle. On travaille dans le module libre (`Dom::FreeModule`) à coefficients dans `R` sur la sous-classe des partitions d'ensembles de $\{1 \ldots n\}$ (`subClass(SetPart, Parameters=N)`). On munit ce module libre d'une structure d'algèbre de dimension finie avec une base particulière (la base naturelle du module libre), en donnant l'élément de la base qui est le 1 de l'algèbre (`oneBasis`) et la loi de multiplication de deux éléments de la base (`mult2Basis`).

Nous montrons maintenant un exemple de calcul. Nous commençons par définir une instance de cette algèbre pour $n = 4$.

```
>> t4 := TitsAlgebra(4):
```

Puis nous mutliplions deux éléments de la base.

```
>> t4([{2, 4}, {1, 3}])*t4([{1, 2, 3}, {4}])

                      B([{2}, {1, 3}, {4}])
```

Note : le `B` qui apparaît ici désigne la base de l'algèbre dans laquelle elle est définie.

On peut alors par exemple très facilement calculer le centre de l'algèbre, c'est-à-dire l'espace des éléments qui commutent avec toute l'algèbre :

```
>> t3 := TitsAlgebra(3):
```

Cosidérons l'élement générique $\sum c_P P$ où les c_p sont des inconnues :

```
>> gen := t3::genericElement(z -> c[op(z)])
 c[{3}, {2}, {1}] B([{3}, {2}, {1}]) +

    c[{2}, {3}, {1}] B([{2}, {3}, {1}]) +

    c[{3}, {1}, {2}] B([{3}, {1}, {2}]) +

    c[{2}, {1}, {3}] B([{2}, {1}, {3}]) +

    c[{1}, {3}, {2}] B([{1}, {3}, {2}]) +

    c[{1}, {2}, {3}] B([{1}, {2}, {3}]) +

    c[{3}, {1, 2}] B([{3}, {1, 2}]) +

    c[{2}, {1, 3}] B([{2}, {1, 3}]) +

    c[{1}, {2, 3}] B([{1}, {2, 3}]) +

    c[{2, 3}, {1}] B([{2, 3}, {1}]) +

    c[{1, 3}, {2}] B([{1, 3}, {2}]) +

    c[{1, 2}, {3}] B([{1, 2}, {3}]) +

    c[{1, 2, 3}] B([{1, 2, 3}])
```

Pour trouver le centre, on résout le système d'équations

$$\Big((\sum c_P P)Q - Q(\sum c_P P) = 0\Big)_Q \tag{4.8}$$

où Q parcourt une base de A_n. Ce système est un système linéaire.

```
>> res := linsolve({coeff(gen*c - c*gen) $ c in t3::basis()})
 [c[{1}, {2, 3}] = 0, c[{2}, {1, 3}] = 0, c[{3}, {1, 2}] = 0,

    c[{1, 2}, {3}] = 0, c[{1, 3}, {2}] = 0, c[{2, 3}, {1}] = 0,

    c[{1}, {2}, {3}] = 0, c[{1}, {3}, {2}] = 0,

    c[{2}, {1}, {3}] = 0, c[{2}, {3}, {1}] = 0,

    c[{3}, {1}, {2}] = 0, c[{3}, {2}, {1}] = 0]
```

Si l'on substitue ce résultat dans l'élément générique on trouve

```
>> mapcoeffs(gen, subs, res)

                    c[{1, 2, 3}] B([{1, 2, 3}])
```

Le centre est donc trivial, égal à la droite engendrée par l'identité de l'algèbre. Il est donc possible de donner à MuPAD, en assez peu de temps, des questions algébriques relativement complexes. Bien sûr, si le système n'est pas un système linéaire, on se trouve rapidement limité par la lenteur de la résolution des systèmes algébriques à l'aide des bases de Groebner. Pour résoudre ce genre de problèmes nous travaillons actuellement sur les possibilités d'interfacer `MuPAD-Combinat` avec des logiciels spécialisés dans la résolution de gros systèmes linéaires ou algébriques.

4.3 Algorithmes génériques et expérimentation

Dans la section précédente, nous avons montré deux exemples de calculs à l'aide de `MuPAD-Combinat`. La méthode utilisée dans ces deux calculs est très générale et il est souhaitable de pouvoir écrire un programme générique qui permet de résoudre le problème indépendamment de la structure dans laquelle on travaille.

Les fonctionnalités orientées objet et en particulier la notion de domaine en MuPAD nous permettent d'écrire du code générique qui prend en paramètre non plus des objets et des opérations, mais une structure algébrique telle qu'une algèbre, un espace vectoriel et une application linéaire. L'un des avantages est que toutes les informations disponibles sur la structure sont accessibles et que c'est le programme plutôt que l'utilisateur qui doit savoir quelles sont les informations dont il a besoin pour travailler. D'autre part, nous pouvons ainsi faire des calculs sur les structures algébriques plutôt que sur les éléments.

4.3.1 Produit tensoriels

Je présente ici deux applications de ces principes. La première, très simple, est la possibilité de construire le produit tensoriel d'espaces vectoriels et d'algèbres.

Reprenons l'algèbre de Tits définie plus haut :

```
>> t3 := TitsAlgebra(3);
```

et définissons deux éléments

```
>> el1 := t3([{1, 3}, {2}]) + 2*t3([{1, 2}, {3}])

                B([{1, 3}, {2}]) + 2 B([{1, 2}, {3}])

>> el2 := a*t3([{1}, {2, 3}]) + 2*t3([{2}, {1}, {3}])

              2 B([{2}, {1}, {3}]) + a B([{1}, {2, 3}])
```

Nous pouvons maintenant demander de calculer le produit tensoriel de ces deux éléments. Auparavant, nous décidons de noter par le symbole `#` le produit tensoriel en MuPAD :

```
>> operators::setTensorSymbol("#")

>> elt := el1 # el2

 2 (B([{1, 3}, {2}]) # B([{2}, {1}, {3}])) +

    a (B([{1, 3}, {2}]) # B([{1}, {2, 3}])) +

    4 (B([{1, 2}, {3}]) # B([{2}, {1}, {3}])) +

    (2 a) (B([{1, 2}, {3}]) # B([{1}, {2, 3}]))
```

Le système se comporte comme prévu. Faisons toutefois quelques remarques. Si l'on demande à l'objet *elt* dans quel espace il vit, la réponse de MuPAD est

```
>> tens := domtype(elt)

 Dom::TensorProductOfFreeModules(

    [TitsAlgebra(3, Dom::ExpressionField()),

    TitsAlgebra(3, Dom::ExpressionField())])
```

On voit donc que l'objet `elt` sait qu'il appartient au produit tensoriel de l'algèbre de Tits par elle-même. De plus :

```
>> elt*elt

 (4 a + 8) (B([{1}, {3}, {2}]) # B([{2}, {1}, {3}])) +

    (4 a) (B([{1}, {3}, {2}]) # B([{1}, {2}, {3}])) +

        2
    (2 a ) (B([{1}, {3}, {2}]) # B([{1}, {2, 3}])) +

    (4 a + 8) (B([{1}, {2}, {3}]) # B([{2}, {1}, {3}])) +

    (4 a) (B([{1}, {2}, {3}]) # B([{1}, {2}, {3}])) +

        2
    (2 a ) (B([{1}, {2}, {3}]) # B([{1}, {2, 3}])) +

    (2 a + 4) (B([{1, 3}, {2}]) # B([{2}, {1}, {3}])) +

    (2 a) (B([{1, 3}, {2}]) # B([{1}, {2}, {3}])) +

     2
    a  (B([{1, 3}, {2}]) # B([{1}, {2, 3}])) +

    (8 a + 16) (B([{1, 2}, {3}]) # B([{2}, {1}, {3}])) +
```

```
  (8 a) (B([{1, 2}, {3}]) # B([{1}, {2}, {3}])) +

      2
  (4 a ) (B([{1, 2}, {3}]) # B([{1}, {2, 3}]))
```

C'est le résultat attendu, si l'on munit le carré de l'algèbre de Tits de sa structure naturelle d'algèbre donnée par

$$(f_1 \otimes g_1)(f_2 \otimes g_2) := (f_1 f_2 \otimes g_1 g_2)\,. \tag{4.9}$$

Ce qui veut dire que le domaine `tens` sait qu'il est une algèbre, comme on peut le vérifier :

```
>> tens::hasProp(Cat::AlgebraWithBasis)

                         TRUE
```

Il sait même plus : c'est une algèbre dont les éléments sont développés sur une base spécifique et donc, par exemple, les opérations d'extraction des coefficients, de conversion vers un vecteur colonne ont du sens :

```
>> coeff(elt)

                    2, a, 4, 2 a
```

Il n'y a théoriquement rien de révolutionnaire, mais c'est d'une utilité irremplaçable dans mes travaux de recherche. Il est ainsi possible d'écrire très naturellement un grand nombre de calculs.

4.3.2 Quelques détails techniques

Dans cette sous-section, je vais décrire brièvement le fonctionnement interne de ce que j'ai présenté dans la section précédente. Les objets MuPAD ont une représentation interne qui est accessible par la commande `extop`. On peut ainsi avoir la représentation de l'élément `el1`.

```
>> el1; extop(el1)

          B([{1, 3}, {2}]) + 2 B([{1, 2}, {3}])

                         3       2
                   poly(_X  + 2 _X , [_X])
```

C'est un polynôme en une variable `_X` dont les coefficients sont les coefficients de `el1`. Ainsi, à la classe combinatoire qui indice la base de `t3` est associée une paire de fonctions appelées *ranker* qui établissent une bijection entre les éléments de cette classe et l'intervalle entier $\{1 \dots r\}$. L'entier associé à une partition est appelé son *rang*. Pour les classes combinatoires infinies, cette bijection est établie à la volée, au fur et à mesure de l'apparition de nouveaux éléments.

De la sorte, le coefficient de la partition `[{1, 2}, {3}]` dont le rang est ici 2, est stocké dans le coefficient de `_X^2`. Par cette méthode, les opérations linéaires sont effectuées de manière très rapide en MuPAD. En revanche, les opérations algébriques, en particulier le produit, demandent de décoder le polynôme.

Ce mécanisme est complètement transparent pour l'utilisateur. Il a un autre avantage important qui provient de la remarque suivante. Le produit tensoriel $K[X]\otimes K[X]$ est isomorphe à $K[X_1, X_2]$. Ceci nous permet de coder les produits tensoriels par des polynômes en plusieurs variables : Ainsi l'élément

```
>> elt := el1 # el2

 2 (B([{1, 3}, {2}]) # B([{2}, {1}, {3}])) +

    a (B([{1, 3}, {2}]) # B([{1}, {2, 3}])) +

    4 (B([{1, 2}, {3}]) # B([{2}, {1}, {3}])) +

    (2 a) (B([{1, 2}, {3}]) # B([{1}, {2, 3}]))
```

est codé par le polynôme :

```
>> extop(elt)

          3     10          3     5          2     10
 poly(2 _X1   _X2   + a _X1   _X2  + 4 _X1   _X2   +

              2     5
    (2 a) _X1   _X2 , [_X1, _X2])
```

Par cette mécanique simple, nous pouvons calculer très vite les produits tensoriels. Il n'est ainsi pas rare que l'affichage d'un produit tensoriel qui demande de nombreux décodages d'un monôme vers un objet combinatoire soit beaucoup plus long que le calcul du simple produit tensoriel.

4.3.3 Matrice des invariants de Cartan

Dans le même ordre d'idées, nous pouvons construire des sous-modules, des quotients de modules, des sous-algèbres, etc. En particulier, j'ai réalisé une implantation d'un algorithme qui permet de calculer les modules simples et les modules projectifs d'une algèbre de dimension finie ainsi que les multiplicités des premiers dans les seconds, que l'on nomme les invariants de Cartan. Ce programme a été l'un des outils expérimentaux majeurs lors du travail sur l'algèbre d'Ariki-Koike (*cf.* [H18]).

L'algorithme repose sur une décomposition de l'identité de l'algèbre en idempotents orthogonaux minimaux : un élément d'une algèbre est dit *idempotent* s'il est non nul et égal à son carré. Deux idempotents e et f sont dits *orthogonaux* s'ils vérifient

$$ef = fe = 0\,. \tag{4.10}$$

Soit A une algèbre de dimension finie. On appelle *décomposition de l'identité* en idempotents orthogonaux toute famille $(e_1, \ldots, e_k)$ d'idempotents deux à deux orthogonaux dont la somme fait 1 :

$$1 = e_1 + \cdots + e_k \qquad \text{et} \qquad e_i e_j = \delta_{i,j} e_i \,. \tag{4.11}$$

De plus, un idempotent est dit *minimal* s'il n'est pas possible de l'écrire comme la somme de deux idempotents orthogonaux.

On dit que deux idempotents minimaux e et f sont *conjugués* s'ils engendrent le même idéal bilatère :

$$AeA = AfA \,. \tag{4.12}$$

On montre que les idéaux Ae et Af sont isomorphes en tant que A-modules si et seulement si les idempotents e et f sont conjugués. Dans le langage de la théorie des représentations un idéal Ae, où e est un idempotent minimal, est appelé *module projectif indécomposable*. Ainsi le calcul des modules projectifs indécomposables se ramène au calcul d'une famille d'idempotents minimaux orthogonaux et à la sélection d'un de ces idempotents dans chaque classe de conjugaison.

Le calcul des idempotents se fait essentiellement en diagonalisant simultanément tous les éléments du centre du quotient de l'algèbre A_n par son radical. Le calcul du radical lui-même se fait grâce au critère des traces de Dickson. Ainsi on peut calculer un tel système d'idempotents en calculant dans l'algèbre et en résolvant des systèmes linéaires.

La matrice des invariants de Cartan est la matrice des dimensions

$$\dim(\hom(P_i, P_j))_{i,j} \tag{4.13}$$

où i et j parcourent l'ensemble des classes d'isomorphie de modules projectifs indécomposables. Elle se calcule avec les idempotents par

$$\dim(\hom(Ae_i, Ae_i)) = \dim(e_i A e_j) \tag{4.14}$$

où les e_i sont les idempotents associés au P_i. C'est essentiellement la méthode que j'utilise. Précisons que cette méthode n'est pas nouvelle, elle est utilisée par exemple pour analyser les représentations des groupes finis.

Voici donc un exemple d'utilisation de ce programme :

```
>> t3 := TitsAlgebra(3):
```

Nous demandons à MuPAD d'afficher les progrès du calcul au fur et à mesure :

```
>> setuserinfo(experimental::algTools, 3);
```

```
>> experimental::algTools::Desosse(t3):
```

```
Representation theory of TitsAlgebra(3, Dom::ExpressionField()\
) of dimension 13
Preliminary temporary checks...done
Computing the radical...
Computing the matrices (dim = 13) .............
Computing the traces .............
Solving...Ok (nEqu=5, dim=8)
radical done
Dimension of the semi-simple quotient: 5
Computing the semi-simple center...(using the full basis).....\
done
Commutative semi-simple quotient
Decomposing the semi-simple center...(using the full basis) 5 \
3 .3 2 1 1 1 2 1 1 done
Normalizing into idempotents (and checking) .....done
Dimensions of simples:
 [1, 1, 1, 1, 1]
Lifting idempotents..Ok .Ok ..Ok ..Ok .Ok done
Dimensions of projectives:
 [6, 2, 2, 2, 1]
Cartan Matrix:
 [1, 1, 1, 1, 2], 1
 [0, 1, 0, 0, 1], 1
 [0, 0, 1, 0, 1], 1
 [0, 0, 0, 1, 1], 1
 [0, 0, 0, 0, 1], 1

Desosse finished. Elapsed time: 3848 ms
```

Comme décrit dans le chapitre 3, notre travail consiste souvent à deviner une algèbre de dimension finie à partir de la connaissance de la «numérologie» de sa théorie des représentations : dimensions des simples, dimensions des projectifs, invariants de Cartan. On a ainsi un outil extrêmement efficace pour savoir si l'on a bien deviné.

4.4 Perspectives

Nous avons montré que `MuPAD-Combinat` était une plate-forme adaptée à l'écriture des algorithmes de la combinatoire algébrique et en particulier de la combinatoire de la théorie des représentations. Dans un premier temps, en m'appuyant sur l'algorithme présenté dans la section précédente, je pense être en mesure d'écrire un algorithme pour calculer dans les anneaux de Grothendieck d'une tour d'algèbres. Comme je l'ai décrit, on pourrait ainsi calculer les premiers coefficients de structures d'une grande variété d'algèbres de Hopf combinatoires et être capable de reconnaître beaucoup plus vite à quelle tour d'algèbres correspond un couple d'algèbres de Hopf en dualité.

D'autre part, une limitation actuelle de notre système est due au fait que tous les calculs sont faits sur des bases linéaires, alors que souvent ils pourraient être plus efficaces en utilisant les générateurs algébriques. Une première ouverture à été faite en ce sens par G. Duchamp, et É. Laugerotte, en utilisant la théorie des automates. Cette méthode devrait pouvoir permettre de traiter des exemples notablement plus gros.

Bibliographie

[A] E. Abe, *Hopf algebras*, Cambridge University Press **74** (1977).

[Ari] S. Ariki, *On the decomposition numbers of the Hecke algebra of* $G(m, 1, n)$, J. Math. Kyoto Univ., **36** (1996), 789–808.

[AK] S. Ariki et K. Koike, *a Hecke Algebra of* $\mathbb{Z}/r\mathbb{Z} \wr \mathfrak{S}_n$ *and construction of its irreducible representations*, Adv. Math, **106** (1994), 216–243.

[AB] J.-C. Aval et N. Bergeron, *Catalan paths and quasi-symmetric functions*, Proc. Amer. Math. Soc., **131** (2003), no. 4, 1053–1062 (electronic).

[ABB] J.-C. Aval, F. Bergeron et N. Bergeron, *Ideals of Quasi-Symmetric Functions and Super-Covariant Polynomials for* S_n, Adv. Math. **181** (2004), no. 2, 353–367.

[BH] P. Baumann et C. Hohlweg, *A Solomon-type epimorphism for Mantaci-Reutenauer's algebra of a wreath product* $G \wr \mathfrak{S}_n$, en préparation.

[H15] N. Bergeron, F. Hivert et J.-Y. Thibon, *The peak algebra and the Hecke-clifford algebras at* $q = 0$, J. Combinatorial Theory A, **117** (2004), 1–19.

[BMSW] N. Bergeron, S. Mykytiuk, F. Sottile et S. van Willigenburg, *Non-commutative Pieri operations on posets*, J. Comb. Theory, Ser. A, **91** (2000), 84–110.

[BZ] N. Bergeron et M. Zabrocki, *q and q,t-Analogs of Non-commutative Symmetric Functions*, préprint `math.CO/0106255`.

[H8] F. Boulier, F. Hivert, D. Krob et J.-C. Novelli, *Pseudo-permutations. II. Geometry and representation theory,* dans *Discrete models : combinatorics, computation, and geometry* (Paris, 2001), Discrete Math. Theor. Comput. Sci. Proc., AA (2001), 123–132 (electronic).

[BP] P. Bouwknegt et K. Pilch, *The deformed Virasoro algebra at roots of unity*, Commun. Math. Phys. **196** (1998), 249-288.

[Car] R. W. Carter, *Representation theory of the* 0*-Hecke algebra*, J. Algebra **15** (1986), 89–103.

[H7] J. Cassaigne, M. Espie, F. Hivert, D. Krob et J.-C. Novelli, *The Chinese monoid,* Internat. J. Algebra Comput., **11(3)** (2001), 301–334.

[C1] I.Cherednik, *Double affine Hecke algebras, Knizhnik- Zamolodchikov equations, and Macdonald's operators*, IMRN (Duke M.J.), **9** (1992), 171–180.

[C2] *Double affine Hecke algebras and Macdonald's conjectures*, Annals of Mathematics **141** (1995), 191-216.

[DGG] A. Denise, M.-C. Gaudel et S.-D. Gouraud, *A Generic Method for Statistical Testing*, dans *15th IEEE International Symposium on Software Reliability Engineering*, novembre 2004 ; à paraître.

[DKLT] G. Duchamp, D. Krob, B. Leclerc et J.-Y. Thibon, *Fonctions quasi-symétriques, fonctions symétriques non commutatives, et algèbres de Hecke à* $q = 0$, C.R. Acad. Sci. Paris, **322** (1996), 107–112.

[H4] G. Duchamp, F. Hivert et J.-Y. Thibon, *Some common generalizations of quasi-symmetric functions and noncommutative symmetric functions*, C. R. Acad. Sci. Paris, **328** (1999), Série I, 1113–1116.

[H6] G. Duchamp, F. Hivert et J.-Y. Thibon, *Some Generalizations of Quasi-symmetric Functions and Noncommutative Symmetric Functions*, dans *Actes du 12e congrès «Séries Formelles et Combinatoire algébrique»*, Université d'État de Moscou, juin 2000 (D. Krob, A.-A. Mikhalev et A.-V. Mikhalev éditeurs), 170–178, Springer, Berlin (2000).

[H10] G. Duchamp, F. Hivert et J.-Y. Thibon, *Noncommutative symmetric functions VI : Free quasi-symmetric functions and related algebras,* Internat. J. Alg. Comp., **2** (2002), 671–717.

[H21] G. Duchamp, F. Hivert et J.-Y. Thibon, *Noncommutative symmetric functions VII : Free quasi-symmetric functions revisited,*, en préparation.

[GKLLRT] I.M. Gelfand, D. Krob, B. Leclerc, A. Lascoux, V.S. Retakh et J.-Y. Thibon, *Noncommutative symmetric functions*, Adv. in Math., **112** (1995), 218–348.

[Ge] I. Gessel, *Multipartite P-partitions and inner products of skew Schur functions*, dans *«Combinatorics and algebra»*, C. Greene, éd., Contemporary Mathematics **34** (1984), 289–301.

[GR] I. Gessel et C. Reutenauer, *Counting permutations with cycle and descent structure*, J. Combin. Theory Ser., A **64** (1993), no. 2, 189–215.

[GH] A.M. Garsia et M. Haiman, *A graded representation model for Macdonald's polynomials*, Proc. Nat. Acad. Sci. USA **90**, 3607–3610.

[GW] A. Garsia et N. Wallach, *Qsym over Sym is free*, J. Combin. Theory, Ser. A **104** (2003), no. 2, 217–263.

[Gou] S.-D. Gouraud, *Utilisation des Structures Combinatoires pour le Test Statistique*, Thèse de doctorat, juin 2004, Université Paris XI.

[Hai] M. Haiman *Hilbert schemes, polygraphs, and the Macdonald positivity conjecture*, J. Amer. Math. Soc., **14** (2001), 941-1006.

[H1] F. Hivert, *Analogues non-commutatifs et quasi-symétriques des fonctions de Hall-Littlewood, et modules de Demazure d'une algèbre enveloppante quantique dégénérée*, C. R. Acad. Sci. Paris, **362** (1998), Série I, 1-6.

[H2] F. Hivert, *Quasi-symmetric functions and Hecke algebra actions*, dans *Actes du 10e congrès «Séries Formelles et Combinatoire algébrique»*, Fields intitute, Toronto, juin 1998 (N. Bergeron et F. Sottile editeurs) (1998), 325–339 .

[H3] F. Hivert, *Combinatoire des fonctions quasi-symétriques*, Thèse de doctorat, Université de Marne-La-Vallée (1999).

[H5] F. Hivert, *Hecke Algebras, Differences Operators, and Quasi-Symmetric Functions*, Adv. Math., **(155)** (2000), 181-238.

[H9] F. Hivert et N. Thiéry, *Deformation of symmetric functions and the rational Steenrod algebra,* dans *Proceedings of the Classical Invariant Theory Workshop*, Kingston, April 8-19, 2002, CRM Proceedings series of the AMS (2002).

[H11] F. Hivert, J.-C. Novelli et J.-Y. Thibon, *Un analogue du monoïde plaxique pour les arbres binaires de recherche,* C. R. Acad. Sci. Paris, **335** (2002), 1–4.

[H12] F. Hivert, J.-C. Novelli et J.-Y. Thibon, *Sur quelques propriétés de l'algèbre des arbres binaires,* C. R. Math. Acad. Sci. Paris, **337(9)** (2003), 565–568.

[H13] F. Hivert, J.-C. Novelli et J.-Y. Thibon, *An analogue of the plactic monoïd for binary search trees*, dans *Proceedings of the 4th Conference on Combinatorics on Words*, Turku, 10-13 septembre 2003), Turku Centre for Comp. Sc. **27** (2003), (T. Harju et J. Karhumäki éditeurs).

[H14] F. Hivert et N. Thiéry, *Mupad-combinat, an open-source package for research in algebraic combinatorics,* Séminaire Lotharingien de Combinatoire, **51** (2003), 70 p. electronique.

[H16] F. Hivert, J.-C. Novelli et J.-Y. Thibon, *The algebra of binary search trees*, Theoretical Computer Science, à paraître, `math.CO/0401089`.

[H17] F. Hivert, A. Lascoux et J.-Y. Thibon, *Noncommutative symmetric functions and quasi-symmetric functions with two parameters*, prépublications du Isaac Newton Institute de Cambridge, préprint `math.CO/0106191`.

[H18] F. Hivert, J.-C. Novelli et J.-Y. Thibon, *Representation theory of the* 0*-Ariki-Koike-Shoji algebras*, préprint `math.CO/0407218`.

[H19] F. Hivert, *On some generalization of quasi-symmetric functions*, en préparation.

[H20] F. Hivert et N. Thiéry, *The Hecke-Symmetric algebra*, en préparation.

[J] C. G. Jacobi, *De Functionibus alternantibus*, Crelle's Journal, **22**, 360-431.

[KLLO] S.-J. Kang, I.-S. Lee, K.-H. Lee et H. Oh, *Representations of Ariki-Koike algebras and Gröbner-Shirshov bases*, Proc. London Math. Soc. (3), **89** (2004), no. 1, 54–70.

[KnuTao] A. Knutson et T. Tao, *The honeycomb model of GL(n) tensor products I : proof of the saturation conjecture*, Journal of the AMS, **12** (1999), no. 4, 1055–1090.

[KLT] D. Krob, B. Leclerc et J.-Y. Thibon, *Noncommutative symmetric functions II : Transformations of alphabets*, Internal J. Alg. Comput. **7** (1997), 181–264.

[KT1] D. Krob et J.-Y. Thibon, *A Crystalizable version of* $U_q(gl_n)$, dans *Actes du 9e congrès «Série Formelles et Combinatoire Algébrique»*, C. Krattenthaler, éditeur, Vienne (1997).

[KT2] D. Krob et J.-Y. Thibon, *Noncommutative symmetric functions IV : Quantum linear groups and Hecke algebras at* $q = 0$, J. Alg. Comb., **6** (1997), no. 4, 339–376.

[KT3] D. Krob et J.-Y. Thibon, *Noncommutative symmetric functions V : A degenerate version of* $U_q(gl_N)$, Internat. J. Algebra Comput., **9** (1999), no. 3-4, 405–430.

[L1] A. Lascoux, *Cyclic permutations on words, tableaux et harmonic polynomials*, Proc. Conf. on Algebraic Groups (Hyderabad, 1989), Manoj Prakashar, Madras (1991), 323-347

[LS1] A. Lascoux et M. P. Schützenberger, *Le Monoïde plaxique*, Quad. del. Ric. Sci., **109** (1981), 129–156.

[LS2] A. Lascoux et M. P. Schützenberger, *Symmetry and flag manifold*, Invariant Theory Springer L.N., **996** (1983), 118-144.

[LS3] A. Lascoux et M. P. Schützenberger, *Key and standard bases*, dans *Invariant Theory and Tableaux*, IMA vol. in Maths. and Appl., **19** (1990), 125–144.

[LS4] A. Lascoux et M. P. Schützenberger, *Symmetrization operators on polynomial rings*, Funct. Anal. Appl., **21** (1987), 77–78.

[LLT1] A. Lascoux, B. Leclerc et J.-Y. Thibon, *Fonctions de Hall-Littlewood et polynômes de Kostka-Foulkes aux racines de l'unité*, C.R. Acad. Sci. Paris, **316** (1993), 1–6.

[LLT2] A. Lascoux, B. Leclerc et J.-Y. Thibon, *Green polynomials and Hall-Littlewood functions at roots of unity*, Europ. J. Combinatorics, **15** (1994), 173–180.

[LLT3] A. Lascoux, B. Leclerc et J.-Y. Thibon, *Polynômes de Kostka-Foulkes et graphes cristallins des groupes quantiques de type* A_n, C.R. Acad. Sci. Paris, **320** (1995), 131–134.

[LLT4] A. Lascoux, B. Leclerc et J.-Y. Thibon, *Ribbon tableaux, Hall-Littlewood functions, quantum affine algebras, and unipotent varieties*, J. Math. Phys., **38** (1997), 1041–1068.

[LLT5] A. Lascoux, B. Leclerc et J.-Y. Thibon, *Hecke algebras at roots of unity and crystal bases of quantum affine algebras*, Commun. Math. Phys., **181** (1997), 205–263.

[LT] B. Leclerc et J.-Y. Thibon, *Canonical bases of q-deformed Fock spaces*, Int. Math. Res. Notices, **9** (1996), 447–456.

[LR1] J.-L. Loday et M.O. Ronco, *Hopf algebra of the planar binary trees*, Adv. Math., **139** (1998) n. 2, 293–309.

[LR2] J.-L. Loday et M.O. Ronco, *Order structure on the algebra of permutations and of planar binary trees*, J. Algebraic Combin., **15** (2002) n. 3, 253–270.

[Mcd] I. G. Macdonald, *Symmetric functions and Hall polynomials*, Clarendon Press (1995).

[MacM] P. A. MacMahon, *Combinatorial analysis*, Cambridge University Press (1915), Chelsea reprint 1960.

[Mal] C. Malvenuto, *Produits et coproduits des fonctions quasi-symétriques et de l'algèbre des descentes*, Thèse de doctorat, LACIM, UQAM, Montréal (1994).

[MalR] C. Malvenuto et C. Reutenauer, *Duality between quasi-symmetric functions and Solomon descent algebra*, J. Algebra, **177** (1995), 967–982.

[ManR] R. Mantaci et C. Reutenauer, *A generalization of Solomon's descent algebra for hyperoctahedral groups and wreath products*, Comm. Algebra, **23** (1995), 27–56.

[MarMo1] C. Martínez et X. Molinero, *A generic approach for the unranking of labeled combinatorial classes*, Analysis of algorithms (Krynica Morska, 2000). Random Structures Algorithms **19** (2001), no. 3-4, 472–497.

[MarMo2] C. Martínez et X. Molinero, *Unranking of labelled combinatorial structures*, Formal power series and algebraic combinatorics, (Moscow, 2000), Springer, Berlin, 288–299.

[N] P. N. Norton, 0-*Hecke algebras*, J. Austral. Math. Soc. Ser. A, **27** (1979), 337–357.

[O] G. Olshanski, *Quantized universal enveloping superalgebra of type Q and a super-extension of the Hecke algebra*, Lett. Math. Phys., **24** (1992), 93–102.

[P] S. Poirier, *Cycle type and descent set in wreath products*, Disc. Math., **180** (1998), 315–343.

[Rea] N. Reading, *Lattice congruences, fans and Hopf algebra*, préprint.

[Reu] C. Reutenauer, *Free Lie Algebras*, Oxford science publications, (1993).

[Schutz] M.-P. Schützenberger, *La correspondance de Robinson*, dans *Combinatoire et représentation du groupe symétrique*, D. Foata, éditeur, Lecture Notes in Math., Vol. 579, Springer-Verlag, New York, NY, (1977), 59–135.

[Schur] I. Schur, *Über die Darstellung der symmetrischen und der alternieren den Gruppe durch gebrochene lineare Substitutionen*, Crelle's Journal **139** (1911), 155–250 (Ges. Abhandlugen, **1**, 346–441).

[SakS] M. Sakamoto et T. Shoji, *Schur-Weyl reciprocity for Ariki-Koike algebras*, J. of Algebra, **221**, (1999), 293–314.

[SawS] N. Sawada et T. Shoji, *Modified Ariki-Koike algebras and cyclotomic q-Schur algebras*, préprint.

[Sho] T. Shoji, *A Frobenius formula for the characters of Ariki-Koike algebras*, J. of Algebra, **226**, (2000), 818–856.

[St] J. Stembridge, *Enriched P-partitions*, Trans. Amer. Math. Soc., **349** (1997), 763–788.