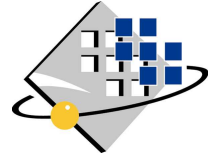




UNIVERSITÉ
PARIS-SUD 11



UNIVERSITÉ PARIS-SUD
FACULTÉ DES SCIENCES D'ORSAY

MÉMOIRE

Présenté pour obtenir

LE DIPLÔME
D'HABILITATION À DIRIGER DES RECHERCHES
DE L'UNIVERSITÉ PARIS XI

Spécialité : Mathématiques

par

Nicolas M. THIÉRY

**Algèbre combinatoire et effective :
des graphes aux algèbres de Kac
via l'exploration informatique**

Soutenu le 10 décembre 2008
devant la commission d'examen :

François Bergeron	Rapporteur
Peter Cameron	Rapporteur
Bernard Leclerc	Rapporteur

Jean-Benoît Bost	Examineur
Mireille Bousquet-Mélou	Examinatrice
Alain Lascoux	Examineur
Jean-Yves Thibon	Examineur
Leonid Vainerman	Examineur
Paul Zimmermann	Examineur

Table des matières

Table des figures	3
Liste de publications	5
Remerciements	7
Prélude	9
Introduction	11
1. Algèbres commutatives et isomorphisme en combinatoire	11
2. Combinatoire pour la théorie des représentations	13
3. Exploration informatique et *-Combinat	16
Chapitre 1. Algèbres commutatives graduées et problèmes d'isomorphisme en combinatoire	17
1. Invariants algébriques de graphes et reconstruction	17
2. Théorie des invariants effective	22
3. Profil et algèbres d'âge des structures relationnelles	24
Chapitre 2. Combinatoire pour la théorie des représentations	29
1. Algèbres de Hecke groupes	29
2. Opérateurs de promotion sur les graphes cristallins affines	37
3. Algèbres de Kac et treillis de sous-facteurs	40
4. Polynômes harmoniques pour les opérateurs de Steenrod	43
Chapitre 3. *-Combinat, boîte à outils pour l'exploration informatique	45
1. *-Combinat, Sharing algebraic combinatorics since 2000	45
2. Démonstration courte	47
3. Modèle de développement	53
4. Conception : modélisation objet	57
5. Conception : représentations multiples	61
6. Étude de cas : objets décomposables et espèces combinatoires	64
7. Liste de publications utilisant ou concernant MuPAD-Combinat	67
Bibliographie	71

Table des figures

1	Mes sujets de recherche et thèmes avoisinants	12
2	Un module combinatoire pour l'algèbre de Hecke dégénérée $H_5(0)$	14
3	Un graphe simple et le jeu de cartes associé	17
4	Récapitulatif des conjectures pour les différentes notions de restructibilité	20
5	Matrice d'incidence des arbres versus les forêts à 4 arêtes, pour $n = 6$ sommets	21
6	Réalisation du modèle d'alcôves au niveau 1 de l'espace $\mathfrak{h}_{\mathbb{Z}}$ des copoids en type $A_1^{(1)}$	33
7	Transitivité de l'action de niveau zéro de la 0-algèbre de Hecke affine	35
8	Le graphe cristallin de forme $(2, 2)$ en type A_4	38
9	Graphes cristallins affines associés au graphe cristallin classique $B(1) \otimes B(3)$ de type A_1	40
10	Exemple de treillis de coidéaux : $KD(6)$	42
11	Le graphe cristallin affine $B^{2,2} \otimes B^{1,1}$ en type $A_2^{(1)}$	49
12	Workflow pour le développement de Sage-Combinat <i>via</i> un serveur de patches	56
13	Hierarchie des catégories (classes abstraites) dans MuPAD et MuPAD-Combinat	59
14	Graphes des conversions dans deux sessions MuPAD-Combinat typiques	62

Liste de publications

Articles dans des revues d'audience internationale avec comité de rédaction

- [1] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as quotients of affine Hecke algebras at level 0. *J. Combin. Theory Ser. A*, November 2009. doi :10.1016/j.jcta.2008.11.010, arXiv :0804.3781 [math.RT].
- [2] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as degenerate affine Hecke algebras. *Discrete Math. Theor. Comput. Sci.*, September 2008. In print.
- [3] Florent Hivert and Nicolas M. Thiéry. The Hecke group algebra of a Coxeter group and its representation theory. *Journal of Algebra*, November 2008. doi :10.1016/j.jalgebra.2008.09.039, arXiv :0711.1561 [math.RT].
- [4] Pierrick Gaudry, Éric Schost, and Nicolas M. Thiéry. Evaluation properties of symmetric polynomials. *Internat. J. Algebra Comput.*, 16(3) :505–523, 2006. <http://hal.inria.fr/inria-00000629>.
- [5] Florent Hivert and Nicolas M. Thiéry. MuPAD-Combinat, an open-source package for research in algebraic combinatorics. *Sém. Lothar. Combin.*, 51 :Art. B51z, 70 pp. (electronic), 2004. <http://mupad-combinat.sf.net/>.
- [6] Jean-Christophe Novelli, Jean-Yves Thibon, and Nicolas M. Thiéry. Algèbres de Hopf de graphes. *C. R. Math. Acad. Sci. Paris*, 339(9) :607–610, 2004. doi :10.1016/j.crma.2004.09.012, arXiv :0812.3407v1 [math.CO].
- [7] Nicolas M. Thiéry and Stéphan Thomassé. Convex cones and SAGBI bases of permutation invariants. In *Invariant theory in all characteristics*, volume 35 of *CRM Proc. Lecture Notes*, pages 259–263. Amer. Math. Soc., Providence, RI, 2004. arXiv :math/0607380 [math.AC].
- [8] Florent Hivert and Nicolas M. Thiéry. Deformation of symmetric functions and the rational Steenrod algebra. In *Invariant theory in all characteristics*, volume 35 of *CRM Proc. Lecture Notes*, pages 91–125. Amer. Math. Soc., Providence, RI, 2004. arXiv :0812.3056v1 [math.CO].
- [9] Maurice Pouzet and Nicolas M. Thiéry. Invariants algébriques de graphes et reconstruction. *C. R. Acad. Sci. Paris Sér. I Math.*, 333(9) :821–826, 2001. arXiv :0812.3079v1 [math.CO].
- [10] Nicolas M. Thiéry. Algebraic invariants of graphs : a study based on computer exploration. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 34(3) :9–20, September 2000. arXiv :0812.3082v1 [math.CO].

Articles dans des actes de conférences internationales avec comité de rédaction

- [1] Florent Hivert and Nicolas M. Thiéry. Representation theories of some towers of algebras related to the symmetric groups and their Hecke algebras. In *Proceedings of FPSAC'06 San Diego*, 2006. arXiv :math/0607391v2 [math.RT].
- [2] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras. In *Proceedings of FPSAC'05 Taormina*, 2005. arXiv :math/0601256 [math.CO].
- [3] Nicolas M. Thiéry. Computing minimal generating sets of invariant rings of permutation groups with SAGBI-Gröbner basis. In *Discrete models : combinatorics, computation, and geometry (Paris, 2001)*, *Discrete Math. Theor. Comput. Sci. Proc.*, AA, pages 315–328 (electronic). *Maison Inform. Math. Discrèt. (MIMD)*, Paris, 2001.

Articles soumis

- [1] Jason Bandlow, Anne Schilling, and Nicolas M. Thiéry. On the uniqueness of promotion operators on tensor products of type a crystals. Submitted, arXiv :0806.3131 [math.CO], June 2008.
- [2] Marie-Claude David and Nicolas M. Thiéry. Exploration of finite dimensional kac algebras and lattices of irreducible intermediate subfactors. Submitted, 83 pages, arXiv :0812.3044 [math.QA], December 2008.

Articles en préparation

- [1] Maurice Pouzet and Nicolas M. Thiéry. Algebraic invariants of graphs and reconstruction, 2008. Preprint, 35 pages.
- [2] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras I : Quasi-polynomiality of the profile. 2008. In preparation, 21 pages.
- [3] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras II : Finite generation. 2008. In preparation, 20 pages.

Thèse de doctorat et autres communications

- [1] Conrado Martínez, Xavier Molinero, and Nicolas M. Thiéry. Generació ordenada de classes d'estructures combinatòries (ordered generation of combinatorial structures). In *In Jornada de Recerca EPSEM 2006*, pages 83–84. EPSEM (Technical College of Manresa <<http://www.eupm.upc.es/>>), Remsa S.L. Manresa, April 2006. ISBN : 84-86784-05-0. ISBN : 978-84-86784-05-8.
- [2] Nicolas M. Thiéry. PerMuVAR, a library for computing in invariant rings of permutation groups. Software demonstration, MEGA 2000, Bath, UK, 2000.
- [3] Nicolas M. Thiéry. Algebraic invariants of graphs ; an experimental study. Poster, ISSAC'99, Vancouver, CANADA, 1999. Awarded best poster prize.
- [4] Nicolas M. Thiéry. *Invariants algébriques de graphes et reconstruction ; une étude expérimentale*. PhD thesis, Université Lyon I, June 1999. N° d'ordre : 167-99.

Remerciements

Je suis très honoré que François Bergeron, Bernard Leclerc et Peter Cameron aient accepté d'être rapporteurs de ce mémoire. Les nombreuses discussions que j'ai eues avec eux, lors de diverses conférences, ont toujours été captivantes et éclairantes.

L'influence de Jean-Yves Thibon sur mes travaux est évidente dans ce mémoire ; les problèmes qu'il a soulevés sont à l'origine d'un chapitre entier. Celle d'Alain Lascoux est moins visible. J'aimerais d'autant plus la faire apparaître à sa juste valeur : certes nous, les « gamins » du phalanstère, n'appliquons pas souvent à la lettre ses nombreux conseils avisés ; c'est sans doute par besoin de les mettre à notre sauce, à notre niveau, pour mieux les intégrer. Mais la démarche est là, profondément ancrée, et c'est grâce à elle que nous avons abouti à bien des résultats de ce mémoire. Après m'avoir soutenu, et avoir épaulé sans réserve le projet \ast -Combinat depuis sa création, Jean-Yves et Alain ont naturellement accepté d'être dans mon jury. Pour tout cela, et pour le phalanstère qu'ils font vivre, merci.

Je voudrais exprimer ma gratitude envers Marie-Claude David et Léonid Vainerman pour m'avoir guidé dans la découverte des facteurs et des algèbres de Kac et envers Léonid pour sa participation à mon jury. Je tiens aussi à remercier Paul Zimmermann, pour son soutien et nos multiples échanges et actions autour du calcul formel libre, en particulier pour la combinatoire. Enfin, c'est un plaisir et un honneur de compter Mireille Bousquet-Mélou dans mon jury.

Cette habilitation doit beaucoup à ces deux dernières années consacrées à la recherche ; merci à la NSF, au MSRI et au CNRS pour le financement, et surtout à tous mes collègues de l'IUT d'Orsay ; je sais l'effort qu'ils ont dû consentir pour cela. Plus généralement, je voudrais remercier toutes les équipes qui m'ont toujours accueilli à bras ouverts, et auprès desquelles je me suis formé : le Laboratoire de Probabilités, Combinatoire et Statistiques de Lyon I (Bernard Roux, Stéphan Thomassé, etc.), les laboratoires GAGE et LIX de l'École Polytechnique (Marc Giusti, Éric Schost, Pierrick Gaudry), le phalanstère de combinatoire, à Rouen et Marne-la-Vallée (Jean-Christophe Novelli, Florent Hivert, Teresa Gomez-Diaz, etc.), l'équipe MuPAD à Paderborn (Christopher Creutzig, Benno Fuchssteiner, Ralf Hillebrand, Walter Oevel, Stefan Wehmeier, etc.), les départements de mathématiques de l'Université de Californie à San Diego (Adriano Garsia, Nolan Wallach) et à Davis (Anne Schilling, Brant Jones, Jason Bandlow, Monica Varizani, Jesus de Loera, etc.), le projet NSF Affine Schubert Calculus (Grant No 0652641, Anne Schilling, Luc Lapointe, Jennifer Morse, Mark Schimozono, Mike Zabrocki, Thomas Lam). Un remerciement tout particulier va à l'équipe d'arithmétique et géométrie algébrique d'Orsay, qui m'offre chaleureusement depuis quatre ans un espace de liberté où seule la production scientifique compte, quels que soient ma thématique et mes collaborateurs. Merci spécifiquement à Jean-Benoît Bost pour sa participation à mon jury et à David Harari pour l'accompagnement de cette habilitation.

Chers codéveloppeurs de \ast -Combinat, chers coauteurs, Anne, Conrado, Éric, Florent, Jason, Jean-Yves, Jean-Christophe, Marie-Claude, Pierrick, Stéphan, Xavier, grand merci pour ce bout de chemin ensemble, hier, aujourd'hui et demain. Au fond, c'est ce chemin, plus que la destination, qui donne tout son sens humain à notre travail.

Cher Maurice, ce mémoire vous est dédié. Vous m'avez pris sous votre aile et appris à être libre et à faire miens les vers de Cyrano :

« Ne pas monter bien haut, peut-être, mais tout seul ! »

De tout cœur, merci.

Et puis ... Ce mémoire ne serait pas ce qu'il est sans le courage de ses relecteurs attentifs et dévoués : Albane, Corinne, Émilia, Florent, Jean, Sandrine. Il n'existerait tout simplement pas sans la patience d'Adèle et d'Élise pendant que papa écrivait « son petit bouquin » et sans le soutien indéfectible et réconfortant de leur maman.

Prélude

Voilà venu le temps de l'habilitation, où je suis censé démontrer que, non content d'avancer vaille que vaille dans ma propre recherche, je peux prendre quelqu'un sous mon aile. Ce présent mémoire a donc vocation à répondre à deux questions : « qui suis-je ? » et « où vais-je ? ».

« Mais sur quoi travaillez-vous au juste ? », me demande-t-on. S'il faut vraiment me mettre dans une case, je choisis la combinatoire algébrique ; ou peut-être plutôt l'algèbre combinatoire. « Mais encore ? » Avec un néophyte, je peux répondre que c'est l'art de compter en utilisant les miraculeuses propriétés de l'addition et de la multiplication ; « c'est bien, mais compter quoi ? Et pour quoi faire ? » J'y reviendrai. En revanche, avec un collègue qui aimerait me ranger dans une petite case, je suis plus perplexe. La figure 1 traduit au mieux l'image que j'ai en tête. Théorie des graphes ? Des invariants ? Des représentations ? Combinatoire bordelaise ? Fonctions symétriques ? Algèbres de Hopf, voire de Kac ? Calcul formel ? Un peu de tout cela ; mais je n'ose répondre oui à aucune de ces questions, de peur d'y passer pour un bétien. Je ne pourrais pas, à l'instar de nombreux collègues que j'envie pour l'occasion, écrire un grand mémoire de synthèse résumant leurs vastes connaissances sur *leur* sujet et ouvrant de grandes portes sur un avenir radieux pour toute une communauté derrière eux.

Suis-je un rêveur éclectique ne sachant rien sur tout ? Peut-être. J'espère que ce mémoire, présentant mes contributions à ces différents sujets de recherche, mettra en valeur ce qui les unit : outils, méthodologies, points de vue.

Que suis-je donc. Avant tout un explorateur. Foin de l'image romantique de l'archéologue devinant la huitième merveille du monde à partir de quelques tessons épars et établissant un délicat plan de fouilles sur vingt ans pour la mettre à jour millimètre par millimètre au pinceau. Non, moi c'est plutôt la dynamite et la tronçonneuse, le GPS et les drones télécommandés. En clair l'exploration informatique, lorsqu'elle s'y prête. Pragmatisme et efficacité. Je suis peut-être bête, mais mon marteau-piqueur est plus gros que le tien, et je sais le manier. Heureusement que le champ des idées est renouvelable à l'infini, sinon j'aurais changé de métier depuis belle lurette (déjà que j'ai mauvaise conscience pour le CO₂ relâché à l'occasion de mes voyages professionnels ou pour faire tourner mes calculs).

Ma stratégie favorite est simple. Choisir une belle montagne perdue dans la brume et chercher un guide pour faire équipe. Ou l'inverse. Au spécialiste du domaine de me transmettre sa science, de m'expliquer les subtilités locales du climat, de m'emmener à la frontière du connu (un problème précis et l'état de l'art environnant). À moi de sortir la tronçonneuse pour défricher (déchiffrer ?) la forêt vierge, d'inventer une nouvelle machine à chasser les nuages. Où sont les obstacles ? Les abîmes ? Les sommets les plus abrupts ? N'y aurait-il pas une vallée suspendue pour traverser le massif en douceur, un pont provisoire pour enjamber le ravin ? Nommer, aplanir grossièrement et cartographier. Définir, remarquer et conjecturer. Puis passer à autre chose. Aux alpinistes des théorèmes de vaincre les plus hauts sommets. Aux bâtisseurs de théories de faire un jardin japonais du champ de bataille jonché de faits incongrus que je laisse derrière moi.

J'apporte un savoir faire et une caisse à outils. En échange, mon équipier apporte un savoir. Un savoir que j'intégrerai d'autant mieux qu'il aura fallu que je l'implante. Pas de flou artistique possible. Ici, la collaboration interdisciplinaire fonctionne, car elle est emmenée par une succession de questions concrètes, évitant le piège des considérations générales et oiseuses. Et à la fin, grâce au logiciel libre, nous repartons tous les deux avec la caisse ; caisse qui à l'occasion s'est enrichie de nouveaux outils, peut-être même d'un nouveau forgeron. Et puis, ensemble, nous publions les résultats. Car il ne s'agit pas de se faire esclave programmeur.

Je suis avant tout chercheur. Le travail d'implantation est intéressant dans la mesure où un nouveau problème nécessite une large part de conception et d'algorithmique nouvelle. Il y a aussi un savoir-explorer. Quelle est la bonne question à poser ? Où faut-il passer en force, où en ratissant au peigne fin ? Où tenter sa chance et lancer un hameçon au hasard ? Quelle confiance accorder à des premiers signes ? Et puis celui qui tient la tronçonneuse et avance devant a toutes les chances d'être le premier à entrevoir de nouveaux phénomènes. D'autant qu'avec le temps l'œil se forme.

L'exploration informatique n'est pas une idée nouvelle en combinatoire algébrique. Schützenberger en fut un pionnier dès les années 1950. Maintenant quasiment tous les chercheurs y ont recours à un moment ou un autre, que ce soit en tapant quelques commandes `Maple`, ou en développant, trop souvent de manière isolée, des bibliothèques de plusieurs dizaines voire centaines de milliers de lignes de code. Mon rêve : mutualiser tout ces efforts de développement pour qu'à la fin chacun ait à sa disposition les meilleurs outils, tout en perdant moins de temps à faire de la technique. C'est tout le sens du projet logiciel `*-Combinat` que j'ai lancé en 2000 avec Florent Hivert et que je décrirai en détail dans le chapitre 3. La stratégie est de cristalliser une communauté transversale autour de `*-Combinat`, en tissant peu à peu des liens à l'échelle internationale. Partis à deux, nous sommes maintenant plus d'une vingtaine, avec 130 000 lignes de code. Cela nécessite de puiser dans le savoir-faire des informaticiens : d'une part, les outils et modèles de développement collaboratifs (par ex. logiciel libre) et, d'autre part, les techniques de conception (par ex. programmation orientée objet) adaptés à notre situation. En bref, industrialiser le processus pour maîtriser le changement d'échelle. Et au final permettre des calculs d'un niveau de complexité supérieur, intégrant simultanément plusieurs techniques algorithmiques (comme de l'algèbre linéaire creuse avancée, de l'élimination type base de Gröbner et des calculs combinatoires sur des objets à isomorphie près ; voir section 1), ou combinant plusieurs constructions conceptuelles (dualité, tenseurs, changements de base, torsion de (co)produits, sous-algèbres et quotients ; voir section 3). Cet effort de mutualisation existe depuis longtemps dans d'autres domaines (par exemple avec `GAP` pour la théorie des groupes). Fait nouveau, il se met en place à l'échelle des mathématiques (par exemple avec `Sage`). Ma modeste contribution est de faire avancer la situation dans mon domaine.

On l'a dit, la combinatoire algébrique se prête en général bien à l'exploration informatique. Mais pas toujours. Une bonne partie de mes recherches (algèbres d'âge, théorie autour des invariants de groupes de permutations) a été faite au tableau noir, avec une bonne vieille craie. Cependant l'approche est restée la même : explorer des exemples concrets, voir et comprendre ce qui se passe, puis abstraire autant que faire se peut ; bref tenter d'appliquer la maxime : *The art of doing mathematics consists in finding that special case which contains all the germs of generality.* —David Hilbert Quoted in N Rose *Mathematical Maxims and Minims* (Raleigh N C 1988).

L'informatique permet d'abord l'étude d'exemples plus conséquents, ce qui peut être essentiel lorsque les premiers exemples non triviaux ne sont déjà plus traitables à la main. Mais transparaissent aussi en filigrane des questions qui me tiennent à cœur : qu'est-ce qui est calculable, en pratique ? Cet objet mathématique, puis-je le modéliser sur ma machine pour pouvoir ensuite lui poser des questions intéressantes ? Jusqu'où peut-on aller avec l'exploration informatique ? J'ai ces questions en tête dès que j'aborde un nouveau sujet. Cela offre un point de vue, certes forcément réducteur, mais qui donne un angle d'attaque, un fil conducteur et une succession de prises pour rentrer dans le sujet. Cela sans jugement de valeurs ni prétention à l'universalité : c'est ce qui fonctionne, pour moi et dans une certaine gamme de problèmes.

Introduction

Ce mémoire fait la synthèse de presque quinze années de recherche, afin d'en dégager les perspectives. Ces années ont été pour moi une période de grande liberté, pendant laquelle j'ai pris le temps de me forger une voie et une démarche personnelle, à mi-chemin entre l'informatique et les mathématiques.

Ce qui m'a attiré vers la combinatoire algébrique, c'est l'ouverture vers d'autres disciplines, en mathématiques, en informatique, ou en physique théorique. Ma démarche est en effet de me construire, petit à petit, une boîte à outils, en élargissant progressivement mon champ de recherches et en tissant un réseau de collaborateurs dans des communautés variées : théorie des invariants, des graphes, des groupes, combinatoire algébrique ou non, calcul formel, etc.

Lorsque, au fil des rencontres scientifiques et des séjours, j'envisage d'aborder un nouveau sujet de recherche, je me pose deux questions : « Ma boîte à outils actuelle me donne-t-elle un point de vue original, m'offrant une chance de voir ce que d'autres n'ont pas encore vu ? » ; et « Quels outils et concepts vais-je apprendre, qui seraient susceptibles de déclencher des progrès sur des sujets en suspens ? ».

Pour aborder un nouveau sujet, je travaille systématiquement en collaboration. Mon ou mes partenaires sont les garants de l'intérêt et de l'originalité dans un domaine où je n'ai pas forcément encore de recul. En retour, j'apporte des outils et une expertise. Mon fonds de commerce est l'exploration de domaines relativement vierges : construire et étudier des exemples, repérer des conjectures. Et surtout, chercher le bon point de vue où les énoncés et, idéalement, les démonstrations s'expriment simplement.

Le langage forme souvent une barrière de communication entre domaines éloignés. C'est pourquoi, dans ma démarche, la combinatoire joue un rôle essentiel pour modéliser simplement, et souvent de manière effective, des problèmes en les abstrayant de leur contexte. Cela permet de nouer de nouvelles collaborations sur des problèmes concrets et précis. Pour la même raison, ma question favorite est : « Comment cela se calcule ? ». Si je suis capable de retranscrire le problème dans un ordinateur, c'est qu'aucune subtilité ne m'a échappé ; les deux parties sont bien sur la même longueur d'onde. La compréhension du contexte et des motivations, le plus souvent essentielle pour parvenir à une solution, vient ensuite naturellement au fur et à mesure de l'échange qui se met en place.

La figure 1 résume les sujets de recherche que j'ai abordés et leurs interconnexions. Deux thèmes principaux se dégagent. Au cœur du premier on trouve les problèmes d'isomorphisme en combinatoire et leur algébrisation. Dans le second, les modèles combinatoires deviennent un outil pour étudier des représentations d'algèbre. Un troisième thème essentiel de mon travail, sous-tendant les deux autres, est le développement d'outils pour l'exploration informatique, en particulier dans le cadre du projet logiciel international **-Combinat* que j'ai fondé en 2000.

1. Algèbres commutatives et isomorphisme en combinatoire

Le premier volet de mes recherches, présenté au chapitre 1, concerne les problèmes d'isomorphisme en combinatoire. Ces problèmes sont notoirement difficiles, l'isomorphisme de graphe étant en quelque sorte l'instance phare. Ainsi, la fameuse conjecture de reconstruction de graphes de Ulam n'est toujours pas résolue malgré un demi-siècle de recherches intenses. Le fil directeur de ce volet est : est-ce que l'encodage algébrique des problèmes d'isomorphisme peut aider à mieux les comprendre ?

1.3. Algèbres d'âge. En parallèle, j'ai élargi mes recherches, de nouveau avec Maurice Pouzet, aux algèbres d'âges des structures relationnelles. Ici, les objets combinatoires, introduits par Rolland Fraïssé, sont les restrictions finies d'une structure relationnelle infinie R (par exemple les sous-graphes finis d'un graphe infini), considérés à isomorphie près. La collection de ces objets est appelée *âge*. La fonction qui les compte par taille est le *profil* $\phi_R(n)$. En dépit de la simplicité et de la grande généralité du cadre, le comportement du profil semble très contraint.

Conjecture 1.1 (Pouzet). *Sous des hypothèses faibles, la série génératrice du profil $\varphi_R(n)$ est une fraction rationnelle dès lors que la croissance de $\varphi_R(n)$ est sous-exponentielle.*

L'encodage algébrique est donné par l'algèbre d'âge de Peter Cameron. Cet encodage permet d'exploiter la richesse des âges comme *modèles combinatoires*. Nous avons montré que l'on peut réaliser, comme algèbres d'âge, outre les invariants de groupes de permutations, de nombreuses algèbres combinatoires commutatives au cœur de travaux récents : en premier plan les *polynômes quasi-symétriques* et de nombreuses variantes.

Notre objectif est d'obtenir des informations sur le profil en utilisant l'algèbre d'âge (dont il donne la série de Hilbert). Ainsi, nous démontrons la conjecture 1.1 sous certaines conditions incluant tous les exemples précités. Plus généralement, nous cherchons à établir un dictionnaire entre propriétés combinatoires de l'âge et propriétés de l'algèbre (engendrement fini, Cohen-Macaulay, etc.). Pour cela, nous tentons de généraliser les théorèmes et outils que j'avais utilisés en théorie des invariants. Les résultats ont été annoncés au fur et à mesure à FPSAC'05 [PT05], CGCS'07¹, et font l'objet de deux publications en fin de préparation [PT08a, PT08b].

2. Combinatoire pour la théorie des représentations

Ce premier volet de ma recherche relève principalement de la combinatoire algébrique au sens strict : l'objectif est d'algébriser des objets et problèmes combinatoires pour mieux les comprendre. Dans le deuxième volet, présenté au chapitre 1, la tendance s'inverse. Le leitmotiv est la recherche de modèles combinatoires simples (mais cependant riches!) pour décrire des structures algébriques et leur représentations. En ce sens, il s'agit plutôt d'*algèbre combinatoire*. On fait le pari que beaucoup de problèmes d'algèbre ne sont difficiles qu'en apparence ; la clef est alors de trouver le bon point de vue, le bon modèle dans lequel la démonstration devient courte et élémentaire. L'exploration informatique joue donc un rôle inestimable pour essayer rapidement de nombreux points de vue. En filigrane apparaissent les algèbres de Hopf, les tours d'algèbres non commutatives, les groupes quantiques, les graphes de représentations, les systèmes de racines (affines) et les algèbres de Hecke associées.

2.1. Algèbres de Hecke groupes. Dans cette thématique, mon sujet principal est l'étude d'un nouvel objet, l'algèbre de Hecke groupe associée à un groupe de Coxeter (section 1). Pour comprendre son intérêt, il faut d'abord en situer le contexte.

Un thème récurrent du Phalanstère de combinatoire de Marne-la-Vallée est l'interprétation des algèbres de Hopf combinatoires comme groupes de Grothendieck des tours d'algèbres de dimension finie [KT97, BHT04, HNT06]. L'exemple originel, dû à Frobenius, est l'algèbre de Hopf des fonctions symétriques (cf. [Mac95, Zel81]), les fonctions de Schur étant les caractères des représentations irréductibles du groupe symétrique.

Dans le cas général de tours d'algèbres non-semi-simples, il faut distinguer entre représentations simples et projectives. Cela donne une paire d'algèbres en dualité. Ainsi, le rôle central joué par la paire d'algèbres duales Fonctions Symétriques Non Commutatives / Fonctions Quasi-symétriques vient en particulier du fait qu'elles encodent la théorie des représentations

1. International Combinatorics, Geometry and Computer Science Conference

des 0-algèbres de Iwahori-Hecke $H_n(0)$ [KT97], l’algèbre de Hecke $H_n(q)$ étant une déformation de l’algèbre du groupe symétrique avec $H_n(1) = \mathbb{C}[\mathfrak{S}_n]$. La combinatoire sous-jacente est celle des rubans et des classes de descentes dans le groupe symétrique (voir figure 2).

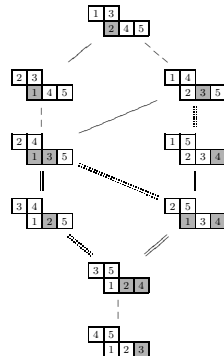


FIGURE 2. Un module combinatoire pour l’algèbre de Hecke dégénérée $H_5(0)$

Dans le même temps, l’algèbre de Hecke affine apparaît en filigrane dans de nombreux travaux du Phalanstère, en particulier comme algèbre d’opérateurs sur les polynômes. C’est par exemple un outil fondamental pour l’étude des polynômes de Macdonald. Ses modules irréductibles de dimension finie ont été classifiés par Zelevinsky [Zel80] au moyen de la combinatoire des multisegments. Dans le cas de la spécialisation centrale principale, cette combinatoire devient à nouveau celle des classes de descentes. Depuis quelques années, un objectif du groupe était donc de résoudre le problème suivant.

Problème 2.1 (Jean-Yves Thibon). *Expliquer pourquoi les représentations de la 0-algèbre de Hecke et celles de la spécialisation centrale principale de l’algèbre de Hecke affine font intervenir la même combinatoire.*

Sa résolution complète a été le fil conducteur notre étude.

À l’occasion d’un groupe de travail en 2003 où nous regardions un problème de physique faisant intervenir l’algèbre de Hecke affine (modèle de Frahm-Polychronakos), nous avons construit avec Florent Hivert, Jean-Christophe Novelli et Jean-Yves Thibon une algèbre d’opérateurs $H\mathfrak{S}_n$ en recollant la 0-algèbre de Hecke $H_n(0)$ et l’algèbre du groupe symétrique \mathfrak{S}_n via leur représentation régulière à droite. À notre surprise, le calcul sur ordinateur de petits exemples a révélé une structure riche faisant intervenir les classes de descentes. Nous nous sommes immédiatement attelés à en découvrir l’origine. Après de nombreux tâtonnements, j’ai enfin mis la main sur la bonne description intrinsèque de $H\mathfrak{S}_n$ comme algèbre d’opérateurs préservant certaines symétries (ou certaines anti-symétries). De là, nous avons déroulé le fil avec Florent Hivert : dimension, base, théorie des représentations, anneau de Grothendieck des caractères. Comme effet de bord, cela a donné notre premier exemple d’anneau de caractères n’étant pas une algèbre de Hopf, la structure d’algèbres et de cogèbres étant incompatibles ; ce fait est maintenant trivial, la dimension de $H\mathfrak{S}_n$ n’étant pas de la forme $r^n n!$ [NB07].

Pour cette étude, nous disposions depuis peu d’un outil mis au point par Florent Hivert et calculant automatiquement la théorie des représentations des premiers étages d’une tour d’algèbres. Pour expérimenter, nous avons alors considéré quelques tours d’algèbres jouets comme l’algèbre du monoïde des fonctions (de parking) croissantes. À notre grande surprise, celles-ci se sont naturellement intégrées dans le schéma, nous permettant de définir des représentations de $(H\mathfrak{S}_n)_n$ et $(H_n(q))_n$ sur les puissances extérieures de la représentation naturelle, et de retrouver comme cas particulier la tour d’algèbres de Temperley-Lieb. Ces résultats sont présentés dans [HT06]. Nous avons depuis généralisé la construction de $H\mathfrak{S}_n$ à n’importe quel

groupe de Coxeter fini. La théorie des représentations reste essentiellement inchangée : dans tous les cas, elle est Morita-équivalente à celle de l'algèbre du treillis booléen [HT08].

Nous avons finalement résolu le problème 2.1 avec Anne Schilling.

Théorème 2.2 (Hivert, Schilling, T. [HST08, HST09]). *Pour tout groupe de Weyl fini, et sauf pour quelques racines de l'unité, l'algèbre de Hecke groupe est le quotient naturel de la q -algèbre de Hecke affine via son action de niveau zéro.*

Ce quotient est de plus compatible avec la spécialisation centrale principale, et les modules simples associés de l'algèbre de Hecke affine donnent, par restriction, les modules projectifs de la 0-algèbre de Hecke.

Il reste maintenant plusieurs pistes à explorer ou en cours d'exploration : le comportement du quotient de l'algèbre de Hecke affine par son action de niveau zéro lorsque q est une racine de l'unité, les liens avec les polynômes de Macdonald non symétriques, la généralisation à tout type du lien avec les fonctions de parking croissantes, etc. Surtout, il reste à répondre à la question : la richesse de la structure des algèbres de Hecke groupes est-elle intrinsèque, ou simplement une ombre portée des algèbres de Hecke affines ?

2.2. Opérateurs de promotion sur les graphes cristallins affines. En marge du sujet que je viens de décrire, j'ai participé à trois projets de recherche. Le premier, présenté en section 2, s'y rattache directement *via* les outils utilisés (groupes de Weyl affines, actions de niveau zéro, graphes combinatoires). En effet, ceux-ci jouent un rôle essentiel dans l'étude des graphes cristallins provenant des représentations de dimension finie des groupes quantiques affines. Une problématique importante, faisant l'objet d'une conjecture de Masaki Kashiwara, est la caractérisation de ces derniers comme produits tensoriels de graphes cristallins de Kirillov-Reshetikin. Avec Anne Schilling et Jason Bandlow nous étudions le type $A_n^{(1)}$. La combinatoire sous-jacente est celle des tableaux. Nous pensons que le cœur du problème est de montrer que, sur les produits tensoriels de k tableaux, le seul opérateur de promotion est induit par celui défini sur les tableaux par Schützenberger au moyen du jeu de taquin. La démonstration pour $k = 2$ fait l'objet d'un article de 31 pages [BST08].

2.3. Algèbres de Kac. L'étude de tours d'algèbres et d'algèbres de Hopf m'a naturellement amené au deuxième projet de recherche (section 3), en collaboration avec Marie-Claude David, autour des algèbres de Kac de dimension finie. Cette catégorie d'algèbres de Hopf contient simultanément les algèbres de groupe et leurs duales, et le point de vue est proche de celui de la théorie des groupes et de la théorie de Galois. Les questions centrales sont, par exemple, la détermination du groupe d'automorphismes et surtout du treillis des sous-structures. Ce dernier point est principalement motivé par l'existence d'une correspondance de Galois entre ce treillis et celui des facteurs intermédiaires de certaines inclusions de facteurs de type II_1 . L'étude de deux familles infinies d'exemples fait l'objet d'une publication de 80 pages [DT08].

2.4. Polynômes harmoniques pour les opérateurs de Steenrod. Le dernier projet de recherche que je présente dans ce mémoire (section 4) est à l'intersection de mes deux volets de recherche. Il concerne une conjecture de Reg Wood venant de topologie algébrique et faisant intervenir l'algèbre de Steenrod. On peut la formuler comme suit :

Conjecture 2.3 (Reg Wood [Woo97], Hivert, T. [HT04a]). *Le sous-espace des polynômes p de $\mathbb{Q}[x_1, \dots, x_n]$ satisfaisant pour tout k l'équation aux dérivées partielles linéaire :*

$$\left(\left(1 + x_1 \frac{\partial}{\partial x_1} \right) \frac{\partial}{\partial x_1}^k + \dots + \left(1 + x_n \frac{\partial}{\partial x_n} \right) \frac{\partial}{\partial x_n}^k \right) p = 0$$

est isomorphe à la représentation régulière graduée du groupe symétrique. En particulier, il est de dimension $n!$.

Cette conjecture est un analogue exact d'un résultat très classique sur les coinvariants du groupe symétrique. Avec Florent Hivert, nous avons donné une formulation de cette conjecture comme analogue quantique, en construisant l'algèbre de Steenrod comme déformation non commutative de l'algèbre de Hopf des fonctions symétriques. Cela nous a permis d'en déduire des résultats partiels [HT04a]. Cependant, malgré les efforts de plusieurs chercheurs, et non des moindres, la conjecture de Reg Wood résiste toujours.

3. Exploration informatique et *-Combinat

Les projets de recherche présentés dans ce mémoire ont en commun l'exploration, et en particulier l'exploration informatique. Elle sert de guide, suggérant des conjectures, occasionnellement donnant des preuves, ou, au contraire, produisant des contre-exemples. La combinatoire algébrique s'y prête bien, car les modèles combinatoires utilisés donnent des représentations concrètes et effectives des objets mathématiques à l'étude.

De plus, on s'intéresse le plus souvent à des familles $(A_n)_{n \in \mathbb{N}}$ d'objets présentant de fortes régularités : typiquement, A_0, A_1 sont triviaux, mais les propriétés intéressantes apparaissent dès $n = 3, 4, 5$ et, si c'est le cas, ont toutes les chances de se prolonger. En ce sens, nous sommes très loin des expérimentations en arithmétique où, du fait de la combinatoire des nombres premiers, les contre-exemples apparaissent souvent très loin. En échange, nous avons le plus souvent à faire face à une explosion combinatoire : les exemples triviaux sont les seuls traitables à la main, et A_5 sera par exemple déjà à la limite de ce que les algorithmes classiques peuvent traiter.

Le défi est de contrôler l'explosion combinatoire, par la modélisation et l'algorithmique, pour gagner un ou deux crans supplémentaires. Cela se fait souvent par approximations successives. La découverte d'un nouveau modèle combinatoire ou d'une nouvelle propriété permet de mieux comprendre les objets ; en retour, cela permet de calculer plus loin et d'en découvrir de nouvelles propriétés.

Bien entendu, mener à bien de tels calculs sous-entend un important travail de programmation, et requiert une large panoplie de techniques (calcul formel, algèbre linéaire creuse, groupes et représentations, fonctions symétriques, manipulations de classes combinatoires, séries génératrices, solveurs divers, etc.). Lors de ma thèse, j'ai regretté l'absence d'une plate-forme bien établie pour la recherche en combinatoire algébrique donnant un accès aisé à tous ces outils.

Cela m'a amené à fonder en décembre 2000 le projet *-Combinat, avec l'aide de Florent Hivert puis, progressivement, de toute une équipe. Sa mission est de fournir une boîte à outils extensible pour l'exploration informatique en combinatoire algébrique, avec comme objectif affiché de fédérer les efforts de développement logiciel dans la communauté de la combinatoire algébrique [HT04b]. L'important investissement initial que m'a demandé ce projet est en train de porter ses fruits, avec une communauté à l'échelle internationale et plus d'une quarantaine de publications afférentes (voir section 7 du chapitre 3).

Je présenterai l'apport de l'exploration informatique à chacun de mes projets de recherche au fil des chapitres 1 et 2. Le chapitre 3 est de toute autre nature. J'y décrirai plus en profondeur le projet *-Combinat. Je détaillerai notamment les défis particuliers rencontrés lors de son développement, et les solutions originales que ceux-ci m'ont amené à mettre au point, tant du point de vue de l'algorithmique que de la conception ou du choix du modèle de développement.

J'espère montrer, à travers ce mémoire, comment le travail de recherche et celui d'ingénierie informatique se complètent et se renforcent mutuellement, le second apportant non seulement des solutions pratiques au premier, mais aussi une source de questions et d'inspiration.

Algèbres commutatives graduées et problèmes d'isomorphisme en combinatoire

Le fil directeur de ce chapitre est l'encodage de familles d'objets combinatoires munies d'une relation d'isomorphisme par des algèbres commutatives graduées. Dans un premier temps (section 1), les objets combinatoires sont les (multi)graphes étiquetés, et l'algèbre est une algèbre de polynômes invariants pour une certaine action par permutation du groupe symétrique. Par la suite, le cadre est généralisé à un groupe de permutation fini quelconque (section 2), puis aux âges des structures relationnelles (section 3).

La construction est toujours la même : les objets à un isomorphisme près forment la base de l'algèbre, la graduation étant donnée par la taille des objets. Le produit traduit alors toutes les manières de combiner deux objets pour en construire un plus gros ; dans certains cas, un coproduit traduit réciproquement comment un objet peut se décomposer en objets plus petits.

L'objectif premier est d'appliquer des outils algébriques à cette construction pour obtenir des informations sur les problèmes d'isomorphisme sous-jacent. Mais en retour cette construction donne des modèles combinatoires riches sur lesquels certaines propriétés algébriques peuvent être lues. Enfin la question de calcul efficace dans ces algèbres, et donc sur ces modèles combinatoires, est centrale, en particulier pour l'exploration informatique.

1. Invariants algébriques de graphes et reconstruction

1.1. Conjecture de reconstruction de graphes de Ulam. Le point de départ de ma thèse est la fameuse conjecture de reconstruction de graphes de Ulam. Elle peut être expliquée en quelques minutes à un non mathématicien. À cet effet, j'ai eu pendant des années en permanence dans ma poche le jeu de cartes (transparentes!) présenté dans la figure 3. Considérons un graphe simple (pas de boucles, pas d'arêtes multiples) non étiqueté comme celui à 5 sommets de la figure. Les cartes en dessous sont obtenues en supprimant à chaque fois un unique sommet du graphe original. Les graphes sont considérés à isomorphie près ; en particulier, on ne tient pas compte ici de la disposition géométrique des graphes. De ce fait les cartes 1 et 3 sont considérées comme identiques. L'ordre des cartes n'est pas significatif, mais on tient cependant compte de leurs répétitions.

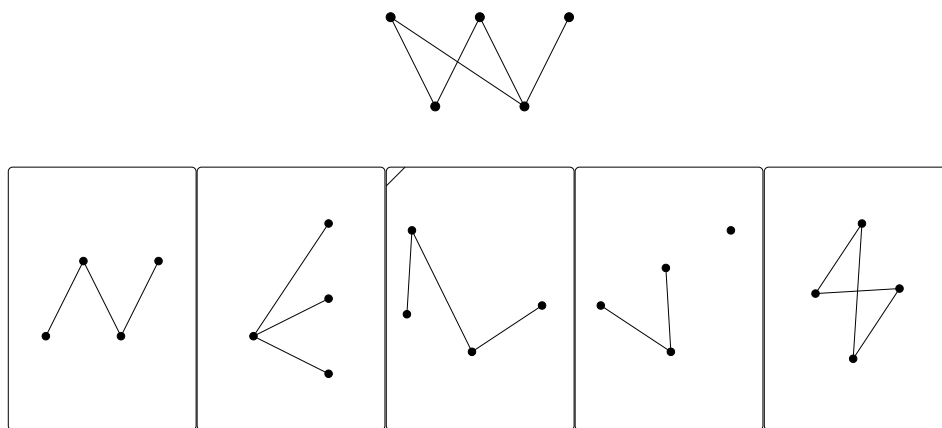


FIGURE 3. Un graphe simple et le jeu de cartes associé

Question 1.1. Est-il possible de retrouver le graphe de départ en ne connaissant que son jeu de cartes ?

C'est un problème de *reconstruction*, similaire à ce que l'on fait en tomographie (par ex. scanner médical) : reconstruire complètement un objet à partir d'un certain nombre de vues partielles. Avant même d'attaquer l'aspect algorithmique « comment reconstruire ? » il faut déjà répondre à la question « est-il possible de reconstruire ? ». Autrement dit, est-ce que le jeu contient suffisamment d'informations pour déterminer entièrement le graphe. Si c'est le cas, le graphe est dit *reconstructible*. Les deux graphes simples à deux sommets ($\bullet \bullet$ et $\bullet \text{---} \bullet$) ont le même jeu, et ne sont donc pas reconstructibles. Y en a-t-il d'autres ?

Conjecture 1.2 (Ulam 1941 [Ula60]). *Tous les graphes simples à au moins trois sommets sont reconstructibles.*

McKay a vérifié [McK97] cette conjecture sur ordinateur pour tous les graphes à au plus 11 sommets (il y en a plus d'un milliard!). C'est maintenant un vieux problème qui a attiré les meilleurs chercheurs de la théorie des graphes et qui est à l'origine d'une vaste littérature [Bon91] avec une multitude de variantes (reconstruction par sommets, par arêtes, etc.). Et pourtant il n'est toujours pas résolu.

Pourquoi ce problème est-il important en général ? Même s'il n'a pas d'application directe, il concentre l'une des difficultés de la reconstruction (ici l'isomorphie) dans le modèle le plus simple imaginable. De ce fait, il participe à la taxonomie générale des problèmes de reconstruction, avec pour objectif d'établir quelles sont les difficultés intrinsèques du sujet. C'est un guide essentiel pour le praticien dans le choix d'un bon modèle pour un problème donné de reconstruction ; si ce modèle contient le problème de Ulam, il peut savoir immédiatement qu'il va au devant de difficultés.

Pourquoi ce problème était-il intéressant pour une thèse ? Vu son historique, il n'était évidemment pas question de l'aborder de front. En revanche, il fournit un excellent cas test pour l'utilisation de nouveaux outils pour traiter de l'isomorphie. Je m'explique. Une des techniques usuelles pour étudier des objets sous l'action d'un groupe est de considérer les quantités qui restent invariantes sous cette action. Par exemple, le nombre d'arêtes d'un graphe ne change pas lorsque l'on renumérote ses sommets. Un des tous premiers résultats de la théorie de la reconstruction est que le nombre d'arêtes d'un graphe est un *invariant reconstructible* (i.e. entièrement déterminé par le jeu du graphe). L'idée est très simple ; il suffit de moyenner le nombre d'arêtes sur les sous-graphes du jeu pour obtenir le nombre total d'arêtes. Plus tard, Tutte [Tut79] a démontré que le déterminant (et plus généralement son polynôme caractéristique) d'un graphe était aussi reconstructible, et Pouzet avait noté que la preuve consistait à démontrer que le déterminant s'exprimait par sommes et produits de paramètres sur les graphes du jeu.

Jusqu'où peut-on espérer généraliser cette technique de preuve ? Pour tenter de répondre à cette question, il est naturel d'introduire l'algèbre des *invariants polynomiaux de graphes*, c'est-à-dire l'algèbre $\mathcal{I}_n := \mathbb{C}[x_{\{i,j\}}]^{\mathfrak{S}_n}$ des polynômes en les $\binom{n}{2}$ variables $(x_{\{i,j\}})_{i < j}$ qui sont invariants sous l'action naturelle du groupe symétrique \mathfrak{S}_n . Un tel invariant peut être évalué sur un graphe simple en remplaçant chaque $x_{\{i,j\}}$ par 1 s'il y a une arête entre i et j dans le graphe, et 0 sinon. Par un résultat général de théorie des invariants, les invariants polynomiaux séparent les graphes à isomorphie près : deux graphes g et g' sont isomorphes si et seulement si ils donnent la même valeur à tous les invariants polynomiaux ; en fait, il suffit d'un nombre fini de polynômes invariants pour séparer. Dans ce cadre, le nombre d'arêtes (correspondant à l'invariant $\sum_{i < j} x_{\{i,j\}}$), ou le déterminant sont des invariants *algébriquement reconstructibles* : ils s'expriment par sommes et produit à partir d'invariants appliqués aux graphes du jeu. Ce qui soulève naturellement la question suivante.

Question 1.3 (Pouzet [Pou77, Pou79]). Pour $n \geq 3$, les polynômes invariants sont-ils tous algébriquement reconstituables ?

Une réponse positive entraînerait une réponse positive à la conjecture de Ulam. Prudence, donc. Faute de moyens d'investigation, cette question était restée vierge (à part pour $n = 3$ pour lequel on se ramène facilement aux polynômes symétriques usuels). L'apparition au début des années 90 d'outils effectifs de calcul dans les invariants [Kem93, Stu93] a motivé mon travail de thèse : *Étudier, tant d'un point de vue théorique que par l'exploration, l'algèbre des invariants sur les graphes (série de Hilbert, systèmes générateurs, etc.), et évaluer ce que la théorie des invariants peut dire sur les problèmes d'isomorphisme et de reconstruction de graphe, et en particulier sur la question 1.3.*

1.2. Reconstruction algébrique de graphes. Commençons par l'aspect reconstruction. Pour une vue synthétique, voir la figure 4.

J'ai montré que la réponse est positive pour $n \leq 5$ (et très probablement pour $n = 6$), j'ai quelque peu étendu la liste des invariants classiques de graphes algébriquement reconstituables, et donné des propriétés générales sur l'algèbre des invariants algébriquement reconstituables. De ces dernières, on déduit que pour $11 \leq n \leq 18$ et très certainement au delà, la réponse à la question 1.3 est négative. Cette approche est-elle donc vaine ? Sans le dire, nous avons fait ci-dessus un choix : considérer l'algèbre des invariants polynomiaux sur \mathbb{C} afin de pouvoir appliquer les résultats de la théorie des invariants. Cet objet est plus gros que nécessaire ; comme les graphes sont simples, $x_{\{i,j\}}$ ne prend que les valeurs 0 et 1. Il aurait d'abord été possible de travailler modulo 2, mais alors la théorie des invariants devient nettement plus ardue. Une autre option aurait été de considérer à la place l'algèbre des graphes simples, obtenue en quotientant par $x_{\{i,j\}}^2 = x_{\{i,j\}}$. C'est ce qu'avaient fait avant moi Kocay [Koc82] et Mnukhin [Mnu92] (voir aussi [Cam96]) ; c'est aussi la direction reprise par la suite par Buchwalder et Mikkonen [MB07]. Au final, leurs résultats sont pour l'instant de la même teneur que les miens : le problème est ardu, et une fois obtenue la reconstruction de quelques invariants explicites, des remarques simples autour des graphes non connexes et des bornes passablement lâches, on ne peut guère aller au delà. Il y a cependant deux différences importantes : d'une part, Mnukhin a démontré que, dans l'algèbre des graphes simples, l'analogie de la question 1.3 est équivalente à la conjecture de Ulam [Mnu92]. En revanche, on perd la graduation, un outil essentiel en théorie des invariants.

Il reste un endroit pour lequel je suis convaincu que la reconstruction algébrique a son mot à dire : une conjecture de Kocay sur la reconstituabilité du nombre d'arbres couvrants d'un type donné [Koc82, Conjecture 5.1]. On est ici au seuil du connu, les arbres étant les plus petits graphes connexes. D'ailleurs, les différentes variantes de l'algèbre coïncident pour l'essentiel à cet endroit là. J'ai obtenu quelques résultats partiels dans cette direction, et je ne résiste pas à mentionner ici ma conjecture préférée issue de cette recherche :

Conjecture 1.4. Soit M_n la matrice d'incidence $(m_{f,a})_{f,a}$ dont les lignes sont indexées par les forêts étiquetées f à n sommets et $n - 2$ arêtes, et les colonnes sont indexées par les arbres a à n sommets (et donc $n - 1$ arêtes) avec $m_{f,a} = 1$ si f est un sous-graphe de a . Alors, M_n est de rang maximal, ses lignes étant linéairement indépendantes.

Même conclusion dans le cas non étiqueté, en prenant pour $m_{f,a}$ le nombre d'occurrences de f dans a (voir figure 5). Ce deuxième point est un corollaire du premier.

J'ai vérifié cette conjecture sur machine jusqu'à $n = 19$ dans le cas non étiqueté. La construction des matrices, très creuses, a été faite en utilisant Nauty et un script Perl. Le rang a été calculé par Jean-Guillaume Dumas à l'aide de Linbox [DV02]. Pour $n = 19$ cela donne une matrice de dimension 241029×317955 , occupant environ 20 Mo de mémoire ; le calcul a duré cinq jours sur un PC à 1 GHz. Cette série de matrices a été utilisée comme banc

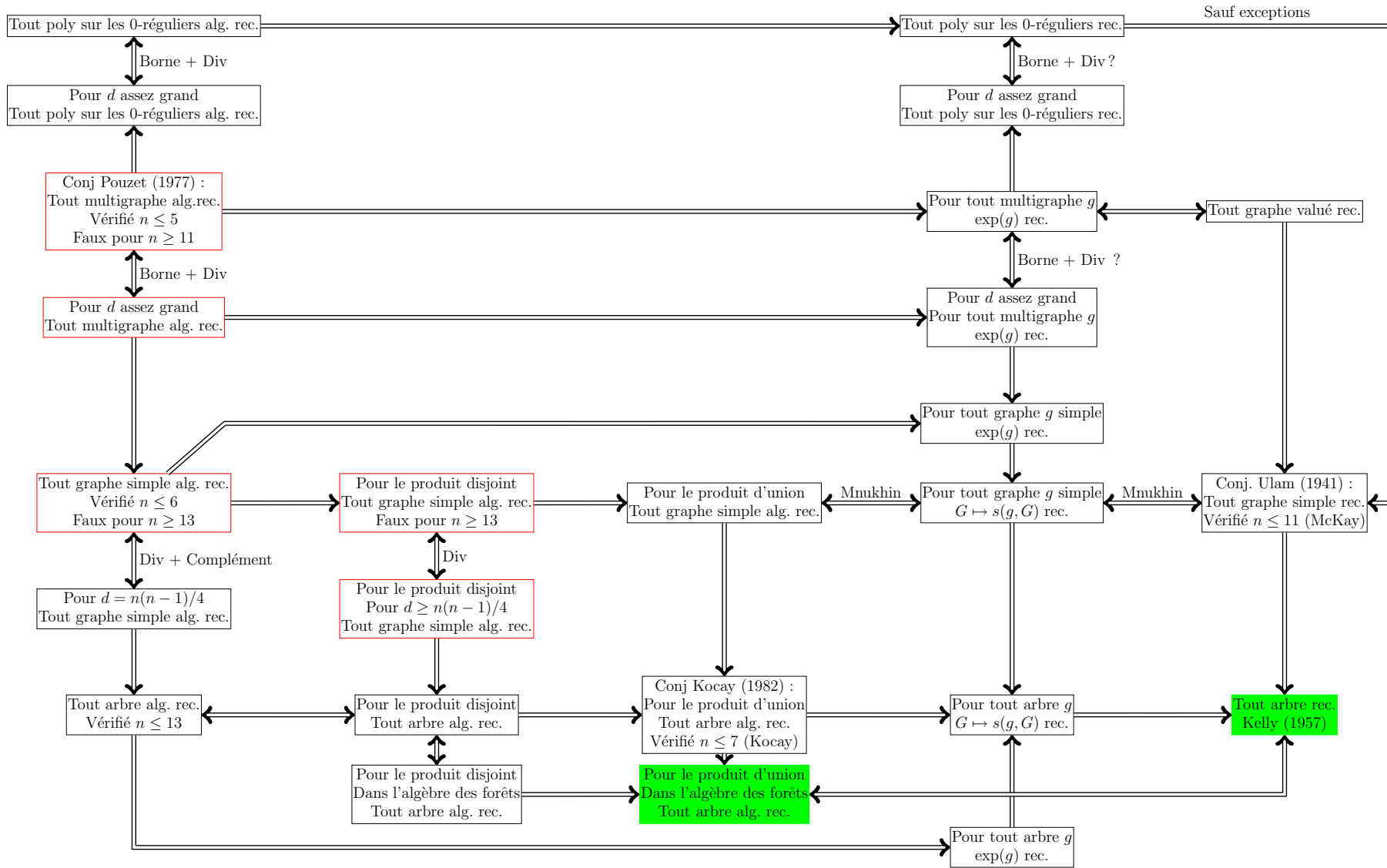


FIGURE 4. Récapitulatif des conjectures pour les différentes notions de restructibilité par sommets et de leurs relations. Nous ne connaissons pas le statut des réciproques non indiquées. Abréviations : rec.=reconstructible; alg. rec.=algébriquement reconstructible; exp(g)= polynôme invariant associé à g .













						
5	1	
.	1	.	.	1	.	
.	3	2	4	1	.	
.	.	2	.	.	2	
.	.	1	.	2	2	
.	.	.	1	1	1	

FIGURE 5. Matrice d'incidence des arbres versus les forêts à 4 arêtes, pour $n = 6$ sommets

d'essai de matrices très creuses pour `Linbox`, confirmant en particulier une analyse théorique sur des matrices aléatoires de la pertinence de l'algorithme de Wiedemann pour ce type de matrices [DV02, DSW03].

Dans le cas étiqueté, j'ai démontré une borne minimale sur le rang qui permet de conclure jusqu'à $n = 7$. Ironiquement, le calcul sur machine s'arrête au même endroit (il y a n^{n-2} arbres étiquetés!).

1.3. Algèbre des invariants de graphes. Les quelques résultats de reconstruction que j'ai obtenus justifient-ils cinq ans d'efforts et 300 pages de thèse? Comme je l'ai dit, la conjecture de Ulam était surtout un cas test, un angle d'attaque, pour étudier l'algèbre \mathcal{I}_n des invariants de graphes. Les résultats de cette étude sont présentés dans ma thèse, dans [Thi99], et sont repris dans [DK02]. J'ai donné des propriétés générales sur l'algèbre \mathcal{I}_n , ainsi que des résultats obtenus par exploration informatique pour n petit. Cela a suggéré plusieurs conjectures (système de paramètres de petit degré, unimodalité) que j'étudie en détail. Je fais aussi un tour d'horizon de variantes de \mathcal{I}_n , obtenant par exemple un système générateur très simple pour le corps des fractions invariants et infirmant à l'occasion un lemme de Grigoriev [Gri79, Lemma I].

L'exploration informatique s'est révélée beaucoup plus ardue que prévu. Illustrons ce point; pour $n \leq 3$, \mathcal{I}_n est une algèbre libre, en fait une algèbre de polynômes symétriques. Le traitement de $n = 4$, à la main avait donné lieu à la publication [ACG96]. Les logiciels existants en 1999 [Kem99] permettaient aussi de traiter $n = 4$ par le calcul en une seconde, mais ne donnaient aucune information pour $n \geq 5$. Le système générateur minimal que j'ai obtenu pour \mathcal{I}_5 est constitué de 57 polynômes en 10 variables de degrés jusqu'à 10. Traiter $n = 6$ complètement est encore hors de portée.

1.4. Conclusions de cette étude. \mathcal{I}_n est très loin d'une algèbre libre. À la différence des polynômes symétriques, elle semble dépourvue de description combinatoire riche (multiples bases dont de Schur, liens avec la théorie des représentations, etc.). Elle est aussi beaucoup trop grosse pour espérer des applications (en particulier algorithmique) à des problèmes d'isomorphie.

La théorie des invariants donne rapidement des informations structurelles (engendrement fini, borne sur les degrés, structure de Cohen-Macaulay, etc.), mais ces informations générales restent très grossières en pratique. Ainsi, pour \mathcal{I}_5 , la borne théorique est de 42 au lieu de 10.

Il apparaît clairement que les techniques actuelles de calcul d'invariants [DK02] buttent sur les limites intrinsèques de l'élimination (bases de Gröbner et variantes), alors même que les applications en combinatoire requièrent l'étude d'exemples de taille plus importante.

2. Théorie des invariants effective

2.1. Invariants de groupes de permutations. Les conclusions de mon étude des invariants de graphes m'ont amené à développer des outils (bibliothèque PerMuVAR [Thi00b] pour MuPAD) pour étudier les invariants de groupes de permutations et à m'intéresser aux aspects effectifs de la théorie des invariants.

D'une part, j'ai mis au point un nouvel algorithme de calcul de systèmes générateurs de ces invariants [Thi01], basé sur des techniques d'élimination respectant les symétries (algorithme de type F4 [Fau99] pour les bases SAGBI-Gröbner [Mil98]). J'y reviendrai dans les perspectives.

D'autre part, j'ai obtenu avec Stéphan Thomassé un résultat structurel sur le comportement de ces invariants vis-à-vis de l'élimination que je décris maintenant. Les bases SAGBI [KM89, RS90] sont les analogues, pour les sous-algèbres des anneaux de polynômes, des bases de Gröbner pour les idéaux. Comme pour ces dernières, elles s'appuient sur l'élimination vis-à-vis d'un ordre sur les termes. Par exemple, la démonstration usuelle du théorème fondamental des fonctions symétriques se fait par élimination, typiquement vis-à-vis de l'ordre lexicographique. De fait, les polynômes symétriques élémentaires e_1, \dots, e_n forment une base SAGBI finie de l'anneau des polynômes symétriques en n variables.

Contrairement aux bases de Gröbner, il n'y a pas de théorème de finitude, et c'est une question ouverte de déterminer pour quelles sous-algèbres et quels ordres on obtient une base SAGBI finie. Le théorème suivant indique que, pour un groupe de permutation non trivial, la base SAGBI est toujours infinie.

Théorème 2.1 (T., Thomasse [TT04]). *Soient G un groupe de permutation agissant sur les variables x_1, \dots, x_n et $<$ un ordre sur les termes quelconque de l'anneau des polynômes $\mathbb{C}[x_1, \dots, x_n]$. Alors la base SAGBI de l'algèbre des invariants $\mathbb{C}[x_1, \dots, x_n]^G$ vis-à-vis de $<$ est finie si et seulement si G est le groupe symétrique (ou un produit direct de tels groupes).*

Nous citons un commentaire du référé anonyme : "[this result] has been desired for a while ; thus it can be said that this paper closes one of the chapters in the book of invariant theory [...] the proof is very beautiful". J'ai depuis généralisé ce résultat au cas des groupes monomiaux et à de nombreuses algèbres d'âge [PT05, PT08b].

2.2. Complexité d'évaluation des fonctions symétriques. D'un autre côté, suite à une courte collaboration entre Florent Hivert et moi-même d'une part et Pierrick Gaudry (LIX) et Éric Schost (STIX) d'autre part, ces derniers ont pu utiliser des algorithmes sur les fonctions symétriques pour accélérer notablement certains calculs sur les courbes hyperelliptiques [GS04]. Cela nous a amenés à entreprendre une étude de complexité précise de l'évaluation de polynômes avec symétries dans le modèle SLP (Straight Line Programm) [GST06]. Nous avons ainsi montré comment, connaissant le coût d'évaluation d'un polynôme symétrique $P(x_1, \dots, x_n)$ décrit par un SLP, on peut donner le coût d'évaluation de P en fonction des valeurs des n fonctions symétriques élémentaires en ces variables. La même technique permet d'obtenir le coût d'évaluation des coefficients de la décomposition d'un polynôme P dans une base des polynômes sur les polynômes symétriques (Schur-Schubert, etc.).

2.3. Perspectives : calcul d'invariants par transformée de Fourier. Les techniques d'évaluation que j'ai apprises lors de ce dernier projet vont resservir pour le calcul d'invariants

de groupes de permutations. Je décris maintenant un projet que j'ai à l'esprit depuis 2004, et qui fait l'objet de la thèse de mon étudiant Nicolas Borie.

Le théorème fondamental de la théorie des invariants, démontré par Hilbert, est l'existence de systèmes finis de générateurs pour l'algèbre des invariants $\mathbb{C}[x_1, \dots, x_n]$ d'un sous-groupe fini de $GL(n)$. La démonstration originale était non constructive. La théorie des invariants effective s'est développée depuis une quinzaine d'années [Kem93, DK02, Kin07], avec pour objectif d'obtenir des algorithmes (et des implantations) efficaces pour la théorie des invariants; le problème typique étant le calcul d'un système minimal de générateurs. L'application première est l'exploration informatique d'exemples.

La stratégie usuelle utilise la décomposition de Hironaka de l'algèbre comme module libre sur les invariants primaires $\Theta_1, \dots, \Theta_n$ pour se ramener à calculer dans le quotient $\mathbb{C}[x_1, \dots, x_n]^G / \langle \Theta_1, \dots, \Theta_n \rangle$ qui est de dimension finie. Pour calculer dans ce quotient, une option est de calculer une base de Gröbner de l'idéal engendré par $\Theta_1, \dots, \Theta_n$ dans $\mathbb{C}[x_1, \dots, x_n]$. C'est par exemple ce qui est implanté dans **Magma**. Pour des groupes de matrices avec n petit, cela se révèle très efficace. En revanche, dès que le nombre de variables grandit (ce qui est souvent le cas pour des applications en combinatoire; par exemple l'algèbre \mathcal{I}_5 requiert 10 variables), ce calcul devient inabordable. Le problème central est que le calcul de la base de Gröbner casse les symétries, et force à travailler dans l'algèbre $\mathbb{C}[x_1, \dots, x_n]$ tout entière, laquelle est de grande dimension dès que l'on monte en degré.

Dans le cas des groupes de permutations, il est possible d'utiliser une variante des bases de Gröbner qui préserve les symétries (bases SAGBI-Gröbner) [Thi01]. Cela permet de calculer un système générateur de \mathcal{I}_5 en quelques minutes; mais *vérifier* que ce système est effectivement générateur reste inabordable sans manipulations spécifiques. En effet le calcul complet nécessite toujours de l'algèbre linéaire dans essentiellement toute l'algèbre des invariants en degré 22 (dimension 174403 à comparer à 20160075 pour les polynômes, et 1 pour le quotient). Il apparaît ainsi clairement que les techniques actuelles de calcul d'invariants butent sur les limites intrinsèques de l'élimination.

Aussi paraît-il judicieux d'introduire d'autres points de vue. L'objectif est d'évaluer une nouvelle stratégie, dans le cas des groupes de permutations, et de manière plus générale des sous-groupes de groupes de réflexions. L'idée est de spécialiser les variables aux racines de l'unité (transformée de Fourier) : cela élimine de facto deux des obstacles principaux actuels : calculs de produits sur les monômes (convolution sur le groupe) et calculs dans le quotient par les invariants primaires (ici, les polynômes symétriques). Ainsi, le nombre de points d'évaluations est exactement la dimension du quotient. Ainsi, pour le problème précédent, on est directement ramené à des calculs dans une algèbre de dimension 30240. Qui plus est, cette algèbre est munie du produit de Hadamard qui est rapide et préserve les structures creuses.

La première étape est de rédiger une démonstration complète de la validité de la stratégie, et d'en obtenir une implantation grossière. Ceci afin de tester concrètement l'approche par des bancs d'essais comparatifs avec les implantations existantes. Il faudra aussi comparer avec d'autres approches par évaluation [Col97, GST06, DSW08].

Une fois la stratégie validée, le champ d'optimisations est très ouvert : comment choisir des invariants dont la transformée de Fourier est creuse, est-il judicieux de représenter les invariants par SLP, peut-on réduire, par filtration, le nombre de points d'évaluations lorsque l'on s'intéresse uniquement aux invariants d'un degré donné, etc. Les progrès viendront principalement de l'étude théorique, sachant que cette approche fait naturellement apparaître des objets combinatoires intéressants comme les spécialisations principales des polynômes de Schur et de Schubert sur un alphabet de la forme $\frac{1}{1-q}$ (voir par exemple [Lit06, BD08]). Cette spécialisation principale joue aussi un rôle naturel dans les descriptions des polynômes de Macdonald et de Schubert par leurs propriétés d'évaluation [Las07].

3. Profil et algèbres d'âge des structures relationnelles

Le travail que je décris dans cette section est un élargissement naturel de l'utilisation d'outils provenant de la théorie des invariants pour traiter algébriquement d'autres problèmes d'isomorphisme en combinatoire. Réalisé en collaboration avec Maurice Pouzet, il s'inscrit dans la lignée de ses travaux sur le profil des structures relationnelles (voir [Pou06] et [Pou08] pour des articles de synthèse).

3.1. Âge et profil d'une structure relationnelle. Une *structure relationnelle* est une paire $R := (E, (\rho_i)_{i \in I})$, où E est un ensemble (le *domaine* de R) et ρ_i est une famille de relations m_i -aires sur E . Typiquement R est un graphe simple : l'ensemble de ses sommets est donné par E et l'ensemble de ses arêtes est décrit par une unique relation ρ_1 binaire ($m_1 = 2$) et symétrique. Nous prendrons comme exemple la somme directe $3K_\infty$ de trois graphes complets infinis.

Sur tout sous-ensemble A de E , R induit par restriction une sous-structure relationnelle sur A . Les notions d'*isomorphisme*, et de *type d'isomorphie*, sont définies naturellement. L'ensemble $\mathcal{A}(R)$ des types d'isomorphie des restrictions finies de R , appelé *âge de R* , a été introduit par Rolland Fraïssé (voir [Fra00]). Le *profil* de R est la fonction φ_R qui compte pour chaque entier n le nombre $\varphi_R(n)$ de types d'isomorphie de sous-structures de R induites sur les ensembles à n éléments [Fra71, Exercice 8 p. 113], [Pou78].

Dans $3K_\infty$, les restrictions sont de nouveau des sommes directes de trois graphes complets au plus. Un type d'isomorphie de taille n peut donc être décrit par une partition de l'entier n à trois parts au plus. Ainsi, la série génératrice de $\varphi_{3K_\infty}(n)$ est donnée par :

$$(1) \quad \sum_{n \in \mathbb{N}} \varphi_{3K_\infty}(n) Z^n = \frac{1}{(1-Z)(1-Z^2)(1-Z^3)}.$$

Si I est fini, $\varphi_R(n)$ est nécessairement fini. Afin de modéliser des exemples venant de l'algèbre ou de la théorie des groupes, il est cependant nécessaire d'autoriser des ensembles d'index I infini. Le profil étant fini dans ces exemples, *nous faisons toujours l'hypothèse que E est infini et que le profil $\varphi_R(n)$ est fini.*

3.2. Algèbre d'un âge. La construction de l'algèbre d'âge de Peter Cameron suit un paradigme classique de réalisation d'algèbre sur les mots, à cela près qu'elle se réalise sur les ensembles [Cam97]. On considère le sous-espace des combinaisons linéaires formelles (éventuellement infinies, mais de degré borné) de sous-ensembles de R , que l'on muni d'un produit commutatif gradué en étendant par linéarité le produit d'union disjoint sur les ensembles : $AB = A \uplus B$ si $A \cap B = \emptyset$ et $AB = 0$ sinon. L'algèbre d'âge a alors pour base les « sommes sur orbites » $m_{\bar{A}} = \sum_{A \in \bar{A}} A$, où \bar{A} parcourt les éléments de l'âge $\mathcal{A}(R)$. Les propriétés de la restriction garantissent qu'il s'agit bien d'une sous-algèbre ; celle-ci est bien entendue graduée connexe. De plus, par construction, la série génératrice $\sum_n \varphi_R(n) Z^n$ est la série de Hilbert de $\mathbb{Q} \cdot \mathcal{A}(R)$.

Par cette construction, les structures relationnelles deviennent un modèle combinatoire pour les algèbres commutatives graduées. Ce modèle est riche, permettant de réaliser, outre les invariants de groupes de permutations, de nombreuses algèbres combinatoires commutatives au cœur de travaux récents : en premier plan les *polynômes quasi-symétriques* qui sont au centre de la théorie des algèbres de Hopf combinatoires [ABS06] et de nombreuses variantes, comme par exemple les *polynômes quasi-symétriques de graphes* que j'ai introduits et étudiés en 2004 avec J.-C. Novelli et J.-Y. Thibon [NTT04]. Mais aussi, par exemple, l'*algèbre des arbres planaires* de Gerritzen [DG04, Ger04a, Ger04b].

La stratégie est de mettre à jour des liens entre propriétés combinatoires de la structure relationnelle et propriétés algébriques de l'algèbre d'âge. Par exemple, Maurice Pouzet a démontré, suite à une conjecture de Peter Cameron [Cam97] dans le cadre des groupes, que l'algèbre d'âge est essentiellement toujours intègre [Pou08]. Il s'ensuit alors, même si cela n'est pas la démonstration la plus courte, que le profil est une fonction non décroissante.

3.3. Décomposition monomorphe finie. Nous nous intéressons principalement à un nouveau cadre, celui des structures relationnelles admettant une *décomposition monomorphe finie*. Cette condition est suffisamment large pour couvrir la plupart des exemples mentionnés ci-dessus (sont exclus l'algèbre des arbres ainsi que les fonctions quasi-symétriques et leurs variantes sur un alphabet infini, car leur croissance est exponentielle). Dans ce cas :

Proposition 3.1 (Pouzet, T. [PT05, PT08b]). *Soit R une structure relationnelle admettant une décomposition monomorphe finie. Alors, l'algèbre d'âge $\mathcal{Q}\mathcal{A}(R)$ se plonge dans les polynômes en un nombre fini de variables (ou un quotient trivial de ceux-ci).*

Notre premier résultat principal donne des informations très précises sur le profil.

Théorème 3.2 (Pouzet, T. [PT05, PT08a, PT08b]). *Soit R une structure relationnelle admettant une décomposition monomorphe finie avec k composantes infinies. Alors, l'âge est un langage rationnel, et le profil de la relation est une fraction rationnelle de la forme suivante :*

$$(2) \quad \frac{P(Z)}{(1-Z) \cdots (1-Z^k)},$$

avec $P \in \mathbb{Z}[Z]$ et $P(1) \neq 0$. En particulier $\varphi_R(n) \approx n^{k-1}$.

Ce théorème confirme en particulier la conjecture 1.1 dans ce cadre.

Le résultat précédent serait essentiellement trivial si l'algèbre d'âge était toujours finiment engendrée. C'est loin d'être le cas, et notre deuxième résultat principal est une caractérisation combinatoire de ce fait.

Théorème 3.3 (Pouzet, T. [PT05, PT08a, PT08b]). *Soit R une structure relationnelle admettant une décomposition monomorphe finie. Alors, l'algèbre d'âge est finiment engendrée si et seulement si la décomposition monomorphe est récursivement minimale. Dans ce cas, l'algèbre est un module de type fini sur une sous-algèbre jouant un rôle similaire à celle des polynômes symétriques.*

Plus généralement, notre objectif est d'étudier jusqu'à quel point il est possible de généraliser chacun des théorèmes et outils que j'ai utilisés en théorie des invariants (voir table 1).

Les démonstrations reposent essentiellement sur des généralisations des techniques d'algèbres de Stanley-Reisner utilisées dans [GS84, Thi00a] pour étudier les invariants de groupes de permutations, sur les ordres d'élimination et sur la théorie de Ramsey.

Cette recherche est basée sur l'exploration d'une multitude d'exemples. C'est toute la richesse des structures relationnelles : il y a une grande souplesse et, selon les contraintes que l'on se fixe, on peut construire toutes sortes d'exemples exotiques. Le prix à payer est qu'il n'y a pas de bonne structure de données générique pour représenter une structure relationnelle ; en dehors de cas particuliers, le calcul sur machine est impuissant. De ce fait, en dehors de quelques calculs élémentaires de séries, l'exploration a été réalisée entièrement au tableau noir.

3.4. Perspectives. Une des approches favorites du Phalanstère est de réaliser les algèbres de Hopf étudiées comme quotients ou sous-algèbres de l'algèbre des mots non-commutatifs. La combinatoire sous-jacente devient alors habituellement simple, ce qui permet de donner des démonstrations élémentaires de la plupart des propriétés algébriques. Cette approche se complète bien avec l'approche opéradique de Jean-Louis Loday consistant en particulier à casser les opérations (produit, coproduits) en plusieurs sous-opérations (algèbre dendriforme

ou tridendriforme), de façon à faire apparaître les algèbres comme provenant de l'action d'une opérade libre sur un petit nombre de générateurs.

Nous essayons avec Jean-Christophe Novelli d'appliquer ces deux approches à l'algèbre des arbres planaires de Lothar Gerritzen [**DG04**, **Ger04a**, **Ger04b**]; cet exemple est intéressant, car il est à la fois combinatoirement très proche des nôtres (j'ai montré que la base de cette algèbre est en bijection naturelle avec le quotient des fonctions de parking par les relations hypoplaxiques), tout en ayant des propriétés algébriques singulières (le seul coproduit connu n'est pas coassociatif). La réalisation que j'ai déjà obtenue en terme d'algèbre d'âge (autrement dit sur les ensembles) semble un bon premier pas vers cette réalisation sur les mots.

Cela mène naturellement à des questions sur les algèbres d'âge, et en tout premier : quelles conditions doit-on imposer sur la structure relationnelle pour pouvoir définir naturellement un coproduit coassociatif sur l'algèbre d'un âge, et en faire ainsi une algèbre de Hopf ?

Un autre problème ouvert important, et difficile, est de caractériser sous quelles conditions ces algèbres sont de Cohen-Macaulay. En effet, s'il est connu depuis longtemps que les invariants de groupes de permutations sont de Cohen-Macaulay en toute caractéristique, la démonstration pour les polynômes quasi-symétriques, même sur les rationnels, est récente [**GW03**]. Existe-t-il une explication unifiée à ces deux phénomènes au niveau de la combinatoire des structures relationnelles ?

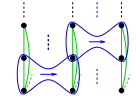
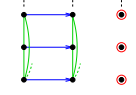
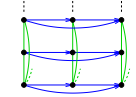
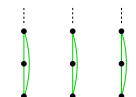
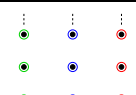
	Structure relationnelle	isomorphismes locaux	Série de Hilbert	Engendrement fini	Borne sur le degré	dimension de Krull	Sym-module	Cohen-Macaulay	SAGBI finie
$ X < \infty$			$\frac{P(Z)^{-\epsilon \mathbb{Z}[Z]}}{(1-Z) \cdots (1-Z^{ X_\infty })}$	jamais ^a	$= \infty^a$	$\leq X_\infty $	non ^b	non ^b	non ^b
Optimalement héréditaire			$\frac{P(Z)^{-\epsilon \mathbb{Z}[Z]}}{(1-Z) \cdots (1-Z^{ X_\infty })}$	oui	$< \infty$	$ X_\infty $	presque	non ^b	non ^b
Formes préservées			$\frac{P(Z)^{-\epsilon \mathbb{Z}[Z]}}{(1-Z) \cdots (1-Z^{ X_\infty })}$	oui	$< \infty$	$ X_\infty $	oui	non ^b	non ^b
Polynômes r -quasi-symétriques [Hiv04]			$\frac{P(Z)^{-\epsilon \mathbb{N}[Z]}}{(1-Z) \cdots (1-Z^{ X })}$	oui	$\leq \frac{ X (X +2r-1)}{2}$	$ X $	oui	oui	non
Invariants d'un groupoïde de permutations G		$G \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{P(Z)^{-\epsilon \mathbb{Z}[Z]}}{(1-Z) \cdots (1-Z^{ X })}$	oui	$\leq \frac{ X (X +1)}{2}$	$ X $	oui	non ^b	jamais ^a
Exemple non Cohen-Macaulay		$\langle 1 \mapsto 2 \rangle \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{1+Z^2+Z^3-Z^4}{(1-Z)^2(1-Z^2)}$	oui	2	$ X $	oui	non	non
Polynômes quasi-symétriques [Ges84]		$\text{Inc} \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{P(Z)^{-\epsilon \mathbb{N}[Z]}}{(1-Z) \cdots (1-Z^{ X })}$	oui	$\leq \frac{ X (X +1)}{2}$	$ X $	oui	oui [GW03]	non
Invariants d'un groupe de permutations G		$G \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{P(Z)^{-\epsilon \in \mathbb{N}[Z]}}{(1-Z) \cdots (1-Z^{ X })}$	oui	$\leq \frac{ X (X -1)}{2}$	$ X $	oui	oui	jamais ^a [TT04]
Polynômes symétriques		$\mathfrak{S}_n \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{1}{(1-Z) \cdots (1-Z^{ X })}$	oui	$ X $	$ X $	oui	oui	oui
Polynômes		$\text{id} \wr \mathfrak{S}_{\mathbb{N}}$	$\frac{(1+Z) \cdots (1+Z+\cdots+Z^{ X })}{(1-Z) \cdots (1-Z^{ X })}$	oui	1	$ X $	oui	oui	oui

TABLE 1. Synthèse de nos résultats sur les algèbres d'âge

a. sauf mention contraire explicite dessous

b. Ici, "non" signifie "pas toujours" : il y a des exemples et des contre-exemples

Combinatoire pour la théorie des représentations

« Make everything as simple as possible, but not simpler. »

Albert Einstein

Ce chapitre présente mes travaux en théorie des représentations. Les objets en jeu sont les tours d'algèbres de dimension finie, les groupes de Coxeter ou de Weyl et leurs algèbres de Hecke (affine), les algèbres de Hopf et groupes quantiques. Le leitmotiv est la recherche de modèles combinatoires simples (mais cependant riches!) pour décrire ces structures algébriques et leur représentations : tableaux, systèmes de racines, modèles d'alcôves, et graphes de modules combinatoires et en particulier graphes cristallins. Le pari est que la plupart des problèmes (mais pas tous!) ne sont difficiles qu'en apparence ; la clef est alors de trouver le bon point de vue, le bon modèle dans lequel la démonstration devient courte et élémentaire. Ces modèles permettent aussi une description constructive, voire effective, des structures algébriques, permettant leur exploration informatique. En retour, l'exploration informatique joue un rôle inestimable pour essayer rapidement de nombreux points de vue, jusqu'à trouver le bon.

1. Algèbres de Hecke groupes

Cette section présente le sujet central de mes recherches en théorie des représentations : l'algèbre de Hecke groupe d'un groupe de Coxeter, obtenue par recollement de son algèbre de groupe et sa 0-algèbre de Hecke. Comme je l'ai mentionné dans l'introduction, l'intérêt que j'ai porté à cette algèbre avec mes collaborateurs avait une double motivation : comprendre les liens entre les représentations de la 0-algèbre de Hecke et de l'algèbre de Hecke affine suggérés par la combinatoire sous-jacente commune des classes de descentes (problème 2.1), et construire de nouveaux exemples d'algèbres (si possible de Hopf) comme groupes de Grothendieck des caractères de tours d'algèbres. À cela se rajoutait la curiosité devant l'existence d'une structure très riche là où nous ne l'attendions pas forcément. Enfin, c'était l'occasion pour moi de comprendre en profondeur des outils (systèmes de racines, algèbres de Hecke affines) qui me seront utiles pour d'autres projets.

1.1. Algèbre de Hecke groupe d'un groupe de Coxeter. Commençons par quelques préliminaires pour définir l'algèbre de Hecke groupe d'un groupe de Coxeter W . Notons $(s_i)_{i \in S}$ les réflexions simples de W , et w_0 son élément maximal (lorsque W est fini). On réalise la représentation régulière à droite de W en faisant agir les opérateurs s_i sur l'espace vectoriel $\mathbb{C}W$ par $w.s_i := ws_i$ ¹. De la sorte, on voit $\mathbb{C}[W]$ comme sous-algèbre de $\text{End}(\mathbb{C}W)$. De même, la 0-algèbre de Hecke peut être réalisée comme la sous-algèbre de $\text{End}(\mathbb{C}W)$ engendrée par les opérateurs π_i définis par $w.\pi_i := ws_i$ si i n'est pas une descente de w , et $w.\pi_i = w$ sinon. Une variante serait de prendre les opérateurs $\bar{\pi}_i$ qui suppriment des descentes. Les opérateurs π_i satisfont les relations de tresse et sont des projecteurs. Ils engendrent un monoïde de taille $|W|$ qui forme une base de la 0-algèbre de Hecke.

En type A , les opérateurs s_i agissent par transposition sur les positions, tandis que les opérateurs π_i et $\bar{\pi}_i$ peuvent être interprétés respectivement comme opérateurs élémentaires

1. La plupart des actions étant ici à droite, le produit fg de deux opérateurs f et g dénote sa composition de gauche à droite, de sorte que $x.fg = (x.f).g$.

d'antitri et de tri à bulle :

$$(3) \quad \begin{array}{ccc} \begin{array}{c} 13854627 \\ s_2 \downarrow \quad \downarrow s_6 \\ 18354267 \\ s_2 \downarrow \quad \downarrow s_6 \\ 13854627 \end{array} & \begin{array}{c} 13854627 \\ \pi_2 \downarrow \quad \downarrow \pi_6 \\ 18354627 \\ \pi_2 \downarrow \quad \downarrow \pi_6 \\ 18354627 \end{array} & \begin{array}{c} 13854627 \\ \bar{\pi}_2 \downarrow \quad \downarrow \bar{\pi}_6 \\ 13854267 \\ \bar{\pi}_2 \downarrow \quad \downarrow \bar{\pi}_6 \\ 13854267 \end{array} \end{array}$$

De manière générale, l'algèbre de (Iwahori)-Hecke générique $H(W)(q_1, q_2)$, engendrée par des opérateurs T_i satisfaisant les relations de tresses ainsi que la relation quadratique $(T_i - q_1)(T_i - q_2) = 0$, peut être construite par interpolation par : $T_i := (q_1 + q_2)\pi_i - q_1s_i$.

De la sorte, on a réalisé simultanément sur le même espace les représentations régulières à droite de toutes les algèbres de Hecke de W , ce qui permet de les recoller.

Définition 1.1 (Algèbre de Hecke groupe [HT08]). *L'algèbre de Hecke groupe est la sous-algèbre de $\text{End}(CW)$ engendrée par les opérateurs s_i et π_i , pour i dans S .*

Cette définition originelle n'est pas très informative. Elle a cependant le mérite de permettre quelques calculs sur machine. Ceux-ci suivis d'une recherche sur l'encyclopédie des suites d'entiers [Se03] suggèrent une combinatoire sous-jacente forte : la dimension de cette algèbre compterait le nombre de paires d'éléments de W sans descentes communes (suite A000275 en type A : 1, 3, 19, 211, ...), tandis que la dimension du quotient semi-simple par le radical serait donnée par la somme des carrés des tailles des classes de descentes (suite A060350 : 1, 2, 10, 88, ...).

Afin de démontrer ces propriétés, nous avons besoin d'une description plus conceptuelle de cette algèbre. Commençons par le rang 1. On montre facilement que la sous-algèbre parabolique $\mathbb{C}[\pi_i, \bar{\pi}_i, s_i]$ de HW est de dimension 3 et que les relations sont données par :

$$(4) \quad \begin{array}{l} s_i\pi_i = \pi_i, \quad s_i\bar{\pi}_i = \bar{\pi}_i, \\ \bar{\pi}_i\pi_i = \pi_i, \quad \pi_i\bar{\pi}_i = \bar{\pi}_i, \\ \pi_i s_i = \bar{\pi}_i, \quad \bar{\pi}_i s_i = \pi_i, \\ \pi_i + \bar{\pi}_i = 1 + s_i. \end{array}$$

En particulier, on peut prendre comme générateurs n'importe quel choix de deux opérateurs dans $\{\pi_i, \bar{\pi}_i, s_i\}$. De plus, on peut prendre $\{1, s_i, \pi_i\}$ comme base. Enfin, on note que la droite $\langle 1 - s_i \rangle$ de CW est stabilisée par l'action de l'algèbre.

Notre premier résultat est une généralisation de ces remarques, qui, comme prévu, fait intervenir les classes de descentes

Théorème 1.2 (H.,T. [HT08]). *Une base de HW est donnée par*

$$\{w\pi_{w'} \mid w, w' \in W \text{ et } D_R(w) \cap D_L(w') = \emptyset\},$$

où $D_L(w)$ et $D_R(w)$ désignent respectivement l'ensemble des descentes à gauche et à droite d'un élément w de W .

HW est l'algèbre des opérateurs de $\text{End}(CW)$ préservant les antisymétries à gauche. Sa transposée HW^* est l'algèbre des opérateurs de $\text{End}(CW)$ préservant les symétries à gauche.

La forme de la base suggère une démonstration par règle de redressement. De fait, nous avons une conjecture sur les relations de cette algèbre, mais nous n'avons pas de preuve de terminaison pour le système de réécriture associé (mais cela a-t-il un intérêt ?).

En fait, ce théorème donne avant tout une définition alternative plus conceptuelle de l'algèbre HW . C'est la découverte de cette définition, *via* l'exploration informatique, qui a permis

de progresser. La démonstration du théorème dans sa globalité est en effet élémentaire, grâce à deux propriétés de triangularité en dualité : l'une du côté de la base, et l'autre du côté des relations linéaires imposées par la préservation des antisymétries.

Cela illustre une idée chère à Alain Lascoux : étudier une algèbre *via* ses représentations concrètes comme algèbre d'opérateurs plutôt que *via* une présentation par générateurs et relations.

1.2. Théorie des représentations. Une fois trouvée la bonne description de l'algèbre de Hecke groupe HW , les suggestions de l'exploration informatique permettent de dérouler sa théorie des représentations ; cette théorie est très uniforme et complètement indépendante du type.

Étant donné un sous-ensemble I de S , notons

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -v, \forall i \in I\}$$

le sous-espace des vecteurs de $\mathbb{C}W$ antisymétriques à gauche pour tout i dans I . Par construction, P_I est un module pour HW . Sa dimension est donnée par la taille de la classe de descente $|S \setminus I|$. Par inclusion, la famille $(P_I)_{I \subset S}$ forme un treillis de modules pour HW anti-isomorphe au treillis booléen ($I \subset J \implies P_J \subset P_I$).

Il est utile pour la suite de choisir une base de $\mathbb{C}W$ compatible par restriction avec chaque P_I . On peut prendre par exemple :

$$\left\{ v_w := \sum_{w' \in W_{S \setminus D_L(w)}} (-1)^{l(w')} w' w \mid w \in W \right\},$$

qui a le bon nombre de vecteurs avec les bonnes antisymétries. Pour être snob, on pourrait aussi prendre la base de Kazhdan-Lusztig.

Proposition 1.3 (H.T. [HT08]). *Une base de HW est donnée par $\{e_{w,w'} \mid D_L(w) \subset D_L(w')\}$, où les $e_{w,w'}$ dénotent les unités matricielles de $\text{End}(\mathbb{C}W)$ vis-à-vis de la base v_w .*

Cette proposition réalise l'algèbre de Hecke groupe comme *algèbre d'un digraphe*. Rappelons que l'algèbre d'un digraphe g est l'algèbre dont la base $\{v_{e,f}\}$ est indexée par les couples e, f de sommets de g tels qu'il existe un chemin de e à f , et dont le produit est donné par $v_{e,f} v_{e',f'} = \delta_{f,e'} v_{e,f'}$. C'est le quotient naturel de l'algèbre des chemins lorsque l'on ne conserve que l'information sur les extrémités des chemins.

Le théorème suivant, décrivant la théorie des représentations de l'algèbre de Hecke groupe HW , est essentiellement un corollaire de la proposition 1.3.

Théorème 1.4 (Hivert, T. [HT08]).

- (i) La famille $(e_{w,w})$ forme une décomposition maximale de l'identité en idempotents orthogonaux ;
- (ii) L'algèbre HW est Morita équivalente à l'algèbre du treillis booléen ;
- (iii) Les P_I sont les modules projectifs indécomposables ;
- (iv) Les modules simples sont obtenus par quotient des modules projectifs $S_I := P_I / \sum_{J \supset I} P_J$; leurs éléments sont anti-symétriques à gauche pour $i \in I$ et symétriques à gauche pour $i \notin I$.

Par restriction, ils donnent exactement :

- Les représentations de Young de forme ruban de W ;
- Les modules projectifs de $H(W)(0)$.

En type A , on peut calculer explicitement les règles d'induction et de restriction pour la tour d'algèbre $(H\mathfrak{S}_n)_n$. Les structures d'algèbres et de cogèbres correspondantes sur groupes de Grothendieck obtenus redonnent des bases connues et des nouvelles bases des fonctions symétriques non commutatives. Cependant ces structures d'algèbres et de cogèbres ne sont pas compatibles, de sorte que l'on obtient pas de nouvelle algèbre de Hopf comme nous l'espérons à l'origine [HT08].

1.3. Algèbres de Hecke groupe et algèbres de Hecke affines. Le dernier point du théorème 1.4 établit un lien clair entre les représentations de l'algèbre de Hecke groupe, et celles de la 0-algèbre de Hecke. Pour clore le problème 2.1, il reste à établir un lien entre l'algèbre de Hecke groupe et l'algèbre de Hecke affine. C'est l'objet du résultat suivant : une description alternative, dans le cas des groupes de Weyl, de l'algèbre de Hecke groupe comme quotient naturel de l'algèbre de Hecke affine. Il s'ensuit que les modules simples de l'algèbre de Hecke groupe sont aussi les modules simples de la spécialisation centrale principale de l'algèbre de Hecke affine.

Théorème 1.5. *Soit W un groupe de Weyl affine (éventuellement tordu), et \mathring{W} le groupe de Weyl classique associé. Soit $\text{cl} : H(W)(q_1, q_2) \rightarrow H\mathring{W}$ le morphisme défini par l'action de niveau zéro de W sur \mathring{W} . Supposons que $q_1, q_2 \neq 0$ et que $q := -\frac{q_1}{q_2}$ n'est pas une racine k -ième de l'unité avec $k \leq 2 \text{ht}(\theta^\vee)$. Alors, le morphisme cl est surjectif et fait de l'algèbre de Hecke groupe $H\mathring{W}$ un quotient de l'algèbre de Hecke affine $H(W)(q_1, q_2)$.*

De plus, le morphisme cl factorise par la spécialisation centrale principale de l'algèbre de Hecke affine.

La signification de l'action de niveau zéro sera précisée plus loin ; quant à la borne $\text{ht}(\theta^\vee)$, nous nous contenterons de préciser qu'elle est linéaire en le rang de W avec une petite constante. Par ailleurs, le rôle particulier des racines de l'unité n'est pas surprenant dans le contexte des algèbres de Hecke.

La démonstration de ce théorème pour q générique repose sur un lemme combinatoire que nous allons d'abord présenter en type A . L'identification ultérieure de la représentation de niveau zéro avec une certaine représentation calibrée de série principale permet de réduire aux petites racines de l'unité les valeurs de q exceptionnelles.

1.3.1. Type A : transitivité du tri à bulle circulaire. Nous avons vu que les opérateurs π_i agissent par antitri à bulle élémentaire. En particulier, partant d'une permutation quelconque, par exemple 51432, on peut par tri à bulle la transformer en la permutation maximale 54321. Cela revient à descendre dans le permutohédre. Par contre, l'opération inverse est impossible ; on ne peut pas remonter.

Écrivons maintenant la permutation 54321 sur un cercle :

$$\begin{array}{cc} & 5 \\ 1 & 4 \\ & 2 \quad 3 \end{array}$$

Cela introduit naturellement une nouvelle position où l'on peut agir, entre la dernière lettre et la première. Y appliquant la même règle que pour les autres positions, nous obtenons :

$$\begin{array}{cc} & 1 \\ 5 & 4 \\ & 2 \quad 3 \end{array}$$

Notons donc π_0 l'opérateur correspondant. Il est clair que l'action de π_0 tend à faire remonter les permutations dans le permutohèdre. Peut-on toujours remonter complètement ?

Lemme 1.6 (H.T. 2005 [HST09]). *Les opérateurs π_0, \dots, π_{n-1} agissent transitivement sur le groupe symétrique \mathfrak{S}_n .*

La démonstration de ce lemme repose sur un algorithme de tri à bulle circulaire récursif².

1.3.2. *Application du lemme combinatoire en type A.* Quel rapport avec l'algèbre de Hecke groupe? Les opérateurs π_0, \dots, π_n agissant sur \mathfrak{S}_n satisfont les relations de la 0-algèbre de Hecke affine $\tilde{H}_n(0)$, dont le diagramme de Dynkin est un cercle. Ils définissent en fait un morphisme cl de $\tilde{H}_n(0)$ dans l'algèbre de Hecke groupe. Ce morphisme est-il surjectif? L'algèbre de Hecke groupe agissant transitivement sur \mathfrak{S}_n , le lemme 1.6 est une condition nécessaire. Nous avons montré, *via* la construction d'une base triangulaire appropriée de $H\mathfrak{S}_n$, qu'elle est en fait suffisante.

Théorème 1.7 (H.T. 2005 [HST09]). *L'action des opérateurs π_0, \dots, π_n sur \mathfrak{S}_n définit un morphisme surjectif de la 0-algèbre de Hecke affine $\tilde{H}_n(0)$ sur l'algèbre de Hecke groupe $H\mathfrak{S}_n$.*

De ce fait, le morphisme de la q-algèbre de Hecke affine $\tilde{H}_n(q)$ dans l'algèbre de Hecke groupe $H\mathfrak{S}_n$ défini par interpolation naturelle est surjectif pour q suffisamment générique.

1.3.3. *Cadre géométrique.* L'action des opérateurs π_0, \dots, π_n en type A a un pendant géométrique qui permet de la définir pour tout type. Soit W un groupe de Weyl. Il est commode de travailler dans l'espace des copoids \mathfrak{h} . À chaque racine α est associée une coracine $\alpha^\vee \in \mathfrak{h}$ et un hyperplan $H_\alpha \subset \mathfrak{h}$ qui coupe \mathfrak{h} en deux demi-espaces H_α^+ et H_α^- . La coracine et l'hyperplan définissent une réflexion s_α . Ils définissent aussi une projection demi-linéaire π_α qui fixe H^- , et envoie H^+ sur H^- par la réflexion s_α . Fixons un choix de racines simples α_i . À chaque chambre de l'arrangement d'hyperplans est associé naturellement un élément de W , de telle sorte que l'action des opérateurs $s_i := s_{\alpha_i}$ et $\pi_i := \pi_{\alpha_i}$ sur les chambres est cohérente avec l'action combinatoire de ces mêmes opérateurs sur W .

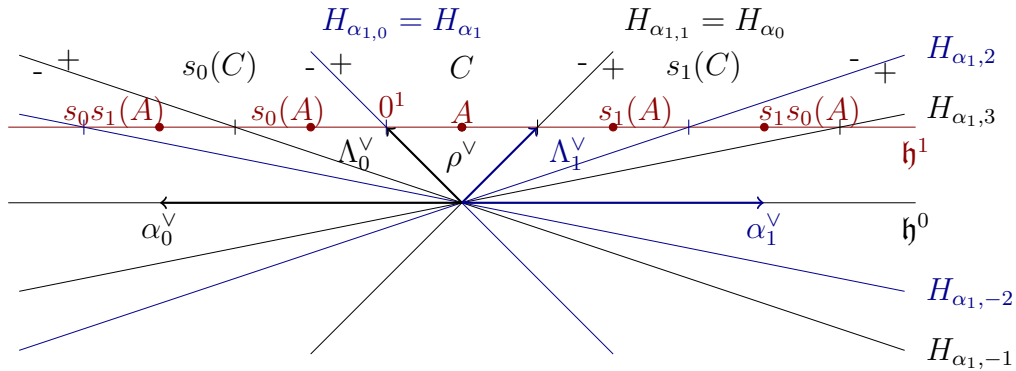


FIGURE 6. Réalisation du modèle d'alcôves au niveau 1 de l'espace $\mathfrak{h}_{\mathbb{Z}}$ des copoids en type $A_1^{(1)}$

Supposons maintenant que W soit un groupe de Weyl affine. De ce fait, les coracines sont toutes dans un même hyperplan \mathfrak{h}^0 (voir figure 6 pour le type $A_1^{(1)}$). Cet hyperplan n'a qu'un nombre fini de chambres, qui sont en correspondance avec un groupe de Weyl classique \check{W} . L'action cl du groupe de Weyl affine W sur \check{W} est appelée usuellement *action de niveau zéro* (en référence au niveau ℓ des hyperplans affines \mathfrak{h}^ℓ parallèles à \mathfrak{h}^0). L'image cl(W) de W par l'action est simplement le groupe classique \check{W} .

² voir <http://inst-mat.italca.cl/fpsac2008/talks/Hivert-Schilling-Thiery.pdf> pour une animation expliquant son fonctionnement

Cette même construction géométrique définit aussi une action de niveau zéro des opérateurs π_0, \dots, π_n de la 0-algèbre de Hecke $H(W)(0)$ sur le groupe de Weyl classique \dot{W} . Mais contrairement à ce qui se passe pour le groupe, $\text{cl}(H(W)(0))$ est plus gros que $H(\dot{W})(0)$, car l'opérateur π_0 ne s'exprime pas en fonction de π_1, \dots, π_n . *La dégénérescence de l'algèbre de Hecke affine via l'action de niveau zéro est non triviale.*

On retrouve alors le même lemme combinatoire qu'en type A .

Théorème 1.8 (S. T. 2008 [HST09]). *Les opérateurs π_0, \dots, π_n agissent transitivement sur le groupe de Weyl classique \dot{W} .*

Nous avons donné au cas par cas des algorithmes récursifs de tri-antitri pour les types classiques dans le même esprit qu'en type A . Nous avons aussi vérifié sur ordinateur, pour tous les types exceptionnels, l'existence d'un algorithme utilisant le même schéma de récurrence. Pour E_7 et E_8 , il a fallu utiliser astucieusement la structure des classes à droite; vérifier directement la forte connexité du graphe de l'action des opérateurs n'était évidemment pas souhaitable (696 729 600 sommets). Enfin, nous avons donné une démonstration géométrique indépendante du type. Les idées sous-jacentes s'inspirent de notes privées de Kashiwara [Kas08] sur les représentations de dimensions finies des groupes quantiques, réinterprétées dans le contexte des chemins d'alcôves. La figure 7 illustre cette démonstration pour tous les groupes de Weyl de rang 2.

On réobtient comme conséquence de ce théorème un fait connu de Kashiwara :

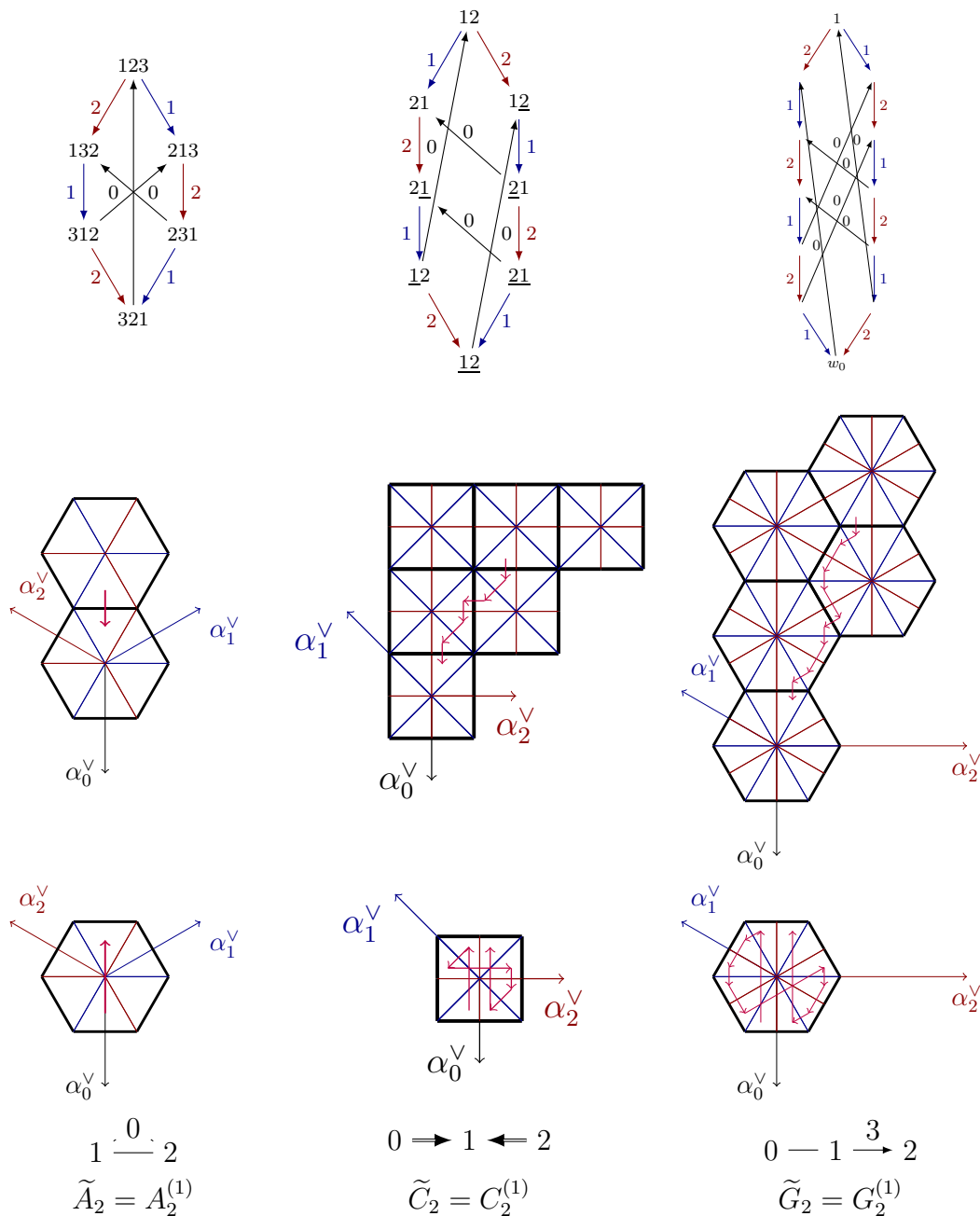
Corollaire 1.9. *Les graphes cristallins affines finis (tels que ceux étudiés plus loin dans la section 2) sont fortement connexes.*

1.3.4. *Représentations de série principale de l'algèbre de Hecke affine.* La fin de la démonstration du théorème 1.5, pour q générique, est une généralisation directe du type A . Pour réduire aux petites racines de l'unité les valeurs de q exceptionnelles, nous avons utilisé, sur la suggestion d'Arun Ram, une autre approche.

Le point de départ est que w_0 dans $\mathbb{C}\dot{W}$ est un vecteur propre pour le tore commutatif $\mathbb{C}[Y^{\alpha_i}]$ de l'algèbre de Hecke affine engendré par les opérateurs de Cherednic Y^{α_i} . On peut alors utiliser une construction classique, due elle aussi à Cherednic, qui permet de construire de nouveaux vecteurs propres grâce aux opérateurs d'entrelacement τ_i (des déformations des T_i qui commutent presque avec les Y_j). Cela revient à utiliser un graphe de Yang-Baxter pour un bon choix de paramètres spectraux (voir [Las03, section 10.7]).

Nous avons alors montré que, lorsque q n'est pas une petite racine de l'unité, les valeurs propres sont suffisamment différentes (représentation *calibrée*) pour garantir que l'on a diagonalisé simultanément l'action des Y_i sur $\mathbb{C}W$. Plus précisément, la représentation de niveau zéro de l'algèbre de Hecke affine est un cas particulier de *représentation de série principale* $M(t)$ (voir par exemple [Ram03, section 2.5]), pour le caractère $t : Y^{\lambda^\vee} \mapsto q^{-\text{ht}(\lambda^\vee)}$. La vérification de la surjectivité du morphisme cl se fait alors grâce à un simple calcul de dimension; celui-ci relie le nombre de vecteurs propres où s'annulent les opérateurs τ_i^2 avec la combinatoire des descentes de \dot{W} .

1.4. Exploration informatique. Dans cette recherche, l'ordinateur a été principalement un outil d'exploration : où y a-t-il de la structure? Quelles conjectures faire? Sur quelles propriétés s'appuyer? Quel est le bon point de vue? Par exemple, la vérification de l'existence d'un algorithme de tri-antitri récursif pour tous les types exceptionnels a fortement motivé la recherche d'une démonstration géométrique. Au final, la plupart des démonstrations sont élémentaires. Par exemple, il n'est pas difficile de dérouler la théorie des représentations de l'algèbre de Hecke groupe, une fois que l'on a vu que la combinatoire sous-jacente est celle des descentes dans le groupe de Coxeter. Ce travail a été aussi pour moi l'occasion, et c'était l'un



En haut : Graphe de l'action de niveau zéro de $\pi_0, \pi_1, \dots, \pi_n$ sur le groupe de Weyl classique \check{W} (notation par permutations signées avec $\underline{2} := -2$).

Milieu : Les alcôves dans l'espace ambiant, avec un plus court chemin d'alcôve descendant depuis une alcôve $w(A)$ dans la chambre dominante telle que $cl(w) = w_0$ jusqu'à l'alcôve fondamentale A .

En bas : Le graphe du haut peut être réalisé géométriquement par le *tore de Steinberg*, quotient des alcôves par les translations, ou de manière équivalente par identification des faces opposées du polygone fondamental. Le chemin d'alcôve de la figure du milieu devient alors un chemin de retour depuis la chambre anti-dominante $w_0(A)$ vers la chambre dominante A .

FIGURE 7. Transitivité de l'action de niveau zéro de la 0-algèbre de Hecke affine $H(W)(0)$ sur le groupe de Weyl classique \check{W} .

des objectifs, de comprendre et d'implanter les systèmes de racines, groupes de Coxeter et de Weyl. Des outils préexistaient pour les premiers, principalement dans le cas fini, par exemple dans `GAP` ou `Maple`. L'algorithmique en est fortement inspirée. La conception en revanche est complètement nouvelle. Elle permet de manipuler simultanément et de manière naturelle les différentes réalisations classiques du réseau des racines ou des poids. De plus, la majorité du code est complètement générique ; il ne dépend que des données de la matrice de Cartan. De ce fait, il s'applique aussi, lorsque cela fait sens, aux cas affine ou de Kac-Moody, et à terme au cas non cristallographique. Cela a fourni une base solide pour l'implantation des cristaux, chemins d'alcôves et algèbres de Hecke affines.

1.5. Perspectives. La structure riche de l'algèbre de Hecke groupe est maintenant bien comprise. L'existence de trois définitions équivalentes (par générateurs, comme algèbre d'opérateurs préservant certaines (anti)symétries, par quotient de l'algèbre de Hecke affine) en fait un objet naturel, qui éclaire les liens entre les représentations de la 0-algèbre de Hecke et de l'algèbre de Hecke affine. En dehors de cela, est-elle utile ? Nous pensons qu'elle est susceptible de fournir un modèle combinatoire simple (*via* les classes de descentes de \mathring{W}) pour mieux comprendre certaines représentations de dimension $|\mathring{W}|$ de l'algèbre de Hecke affine (coinvariants, harmoniques pour l'algèbre de Steenrod, etc.).

Nous présentons ici plusieurs projets de recherche en cours qui dans cette direction, ou tendent à généraliser la structure de l'algèbre de Hecke groupe à des algèbres et monoïdes proches.

1.5.1. *Algèbres de Hecke affines aux racines de l'unité.* L'énoncé du théorème 1.5 soulève immédiatement le problème suivant.

Problème 1.10. *Déterminer l'ensemble des racines de l'unité q pour lesquelles le morphisme $\text{cl} : H(W)(q) \mapsto H\mathring{W}$ n'est pas surjectif.*

Au printemps dernier, j'ai donné l'exploration informatique de ce problème comme sujet de Master 2 à Nicolas Borie. Les calculs sont lourds ; par exemple, traiter \mathfrak{S}_n nécessite de faire de l'algèbre linéaire sur des vecteurs qui sont des matrices $n! \times n!$, le tout sur une extension algébrique des rationnels. L'essentiel de son travail a été d'élaborer des stratégies mathématiques pour exploiter au mieux les résultats partiels.

À l'instant, il semble que la borne du théorème 1.5 soit assez précise : la plupart des petites racines de l'unité, mais pas toutes, semblent donner lieu à un morphisme non surjectif. C'est en particulier toujours le cas pour $q = -1$. Parallèlement au sujet principal de thèse que je lui ai confié, Nicolas Borie va continuer à étudier ce problème. La résolution de difficultés techniques dans `Sage` devrait permettre de mener les calculs suffisamment loin pour établir une conjecture exacte. D'un autre côté, le cas $q = -1$ doit pouvoir être démontré, et cela devrait donner une idée plus fine de la difficulté du cas général.

1.5.2. *Monoïde des fonctions décroissantes de type C .* Lorsque nous avons étudié la théorie des représentations de $H\mathfrak{S}_n$, nous disposions depuis peu d'un outil, mis au point par Florent Hivert, calculant automatiquement la théorie des représentations des premiers étages d'une tour d'algèbres. Pour expérimenter, nous avons alors considéré quelques tours d'algèbres jouets comme l'algèbre $\mathbb{C}[\text{NDF}_n]$ (resp. $\mathbb{C}[\text{NDPF}_n]$) du monoïde des fonctions (resp. fonctions de parking) croissantes. À notre grande surprise, celles-ci se sont naturellement intégrées dans un diagramme. Cela nous a permis de définir des représentations de $(H\mathfrak{S}_n)_n$ et $(H_n(q))_n$ sur les puissances extérieures de la représentation naturelle, et de retrouver comme cas particulier la tour d'algèbres de Temperley-Lieb $(\text{TL}_n)_n$:

$$(5) \quad \begin{array}{ccccccc} H_n(-1) & \xrightarrow{\quad} & H_n(0) & \xrightarrow{\quad} & H_n(1) = \mathbb{C}[\mathfrak{S}_n] & \xrightarrow{\quad} & H_n(q) & \xrightarrow{\quad} & H\mathfrak{S}_n \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ TL_n & \xrightarrow{\quad} & \mathbb{C}[\text{NDPF}_n] & \xrightarrow{\quad} & \mathbb{C}[\mathfrak{S}_n] \circlearrowleft \wedge \mathbb{C}^n & \xrightarrow{\quad} & H_n(q) \circlearrowleft \wedge \mathbb{C}^n & \xrightarrow{\quad} & \mathbb{C}[\text{NDF}_n] \end{array}$$

Avec Tom Denton, doctorant à UC Davis sous la direction de Anne Schilling, nous essayons actuellement de généraliser ce schéma à d'autres types. Cela a déjà fait apparaître des fonctions de parking croissantes signées de type C, dont le monoïde est auto-injectif, et dont la dimension semble être donnée par la suite A086618 de l'encyclopédie des suites d'entiers [Se03].

1.5.3. *Monoïde des opérateurs de tri-antitri.* Avec Anne Schilling, Jean-Christophe Novelli et Florent Hivert, nous avons entrepris l'étude du monoïde $M(W) = \langle \pi_i, \bar{\pi}_i \rangle$ (et non de l'algèbre) engendré par les opérateurs π_i et $\bar{\pi}_i$. À cela deux motivations. D'une part, Philippe Gaucher nous a contactés après être avoir reconnu dans [HT08] les premiers termes de la suite $|M(\mathfrak{S}_n)|$. Selon lui, une meilleure compréhension de ce monoïde pourrait avoir des applications dans la modélisation de processus concurrents par des méthodes de topologie algébrique [Gau08].

D'autre part, il semblerait que ce monoïde ait une structure intrinsèque, même si sa taille n'est pas connue. En effet, le calcul dans les petits cas de sa théorie des représentations indique que les modules simples seraient indexés par les éléments du groupe de Coxeter et que, en type A, la somme des dimensions des modules simples serait donnée par la suite A006245 [Se03] qui compte les réseaux de tris réduits. Une compréhension complète de sa théorie des représentations devrait permettre d'obtenir une formule par sommation pour sa taille $|M(W)|$. Elle pourrait aussi suggérer une description plus conceptuelle de ce monoïde, prérequis pour mieux le comprendre.

1.5.4. *Polynômes de Macdonald non symétriques.* Les opérateurs d'entrelacement ont été originellement introduits pour construire les polynômes de Macdonald non symétriques, ces derniers étant définis comme vecteurs propres simultanés des opérateurs Y^{α_i} de Cherednick de l'algèbre de Hecke affine. Cela suggère bien entendu d'interpréter la base de vecteurs propres de CW obtenus dans la section 1.3.4 comme polynômes de Macdonald. Pour que cette interprétation soit naturelle, il faudrait, en type A, réaliser $\mathbb{C}\mathfrak{S}_n$ comme module quotient de l'algèbre des polynômes sous l'action de l'algèbre de Hecke affine par différences divisées isobares. Il est probable que cette première étape découle directement de la construction de la représentation polynomiale de l'algèbre de Hecke doublement affine (coinvariants).

Il reste à comprendre comment agissent dans cette réalisation les différents opérateurs de l'algèbre de Hecke groupe, à commencer par les permutations, mais surtout T_0 . En effet, la construction des polynômes de Macdonald par opérateurs d'entrelacement τ_1, \dots, τ_n ne donne comme sous-produit qu'une description simple de l'action de T_1, \dots, T_n sur les polynômes de Macdonald, mais pas de T_0 . En revanche, dans l'action de l'algèbre de Hecke groupe sur CW, les opérateurs T_0, T_1, \dots, T_n jouent des rôles très symétriques, et cela pourrait éclairer le rôle de T_0 .

2. Opérateurs de promotion sur les graphes cristallins affines

En parallèle avec notre travail sur les relations entre algèbres de Hecke groupe et algèbres de Hecke affine, mon séjour à l'Université de Californie à Davis a été l'occasion d'une collaboration avec Jason Bandlow et Anne Schilling sur les *graphes cristallins*. Ce sont des graphes orientés, comme celui de la figure 8, qui ont été introduits en théorie des représentations par Masaki Kashiwara afin d'encoder l'essentiel de la structure des modules pour les groupes quantiques $U_q(G)$ lorsque q tend vers 0. Ces graphes jouent en particulier un rôle important en physique

mathématique, en lien avec les modèles intégrables sur réseaux (voir, par exemple [HKO⁺02]). Les arêtes indexées par i décrivent l'action des opérateurs descendants f_i , les opérateurs montants e_i étant les inverses locaux.

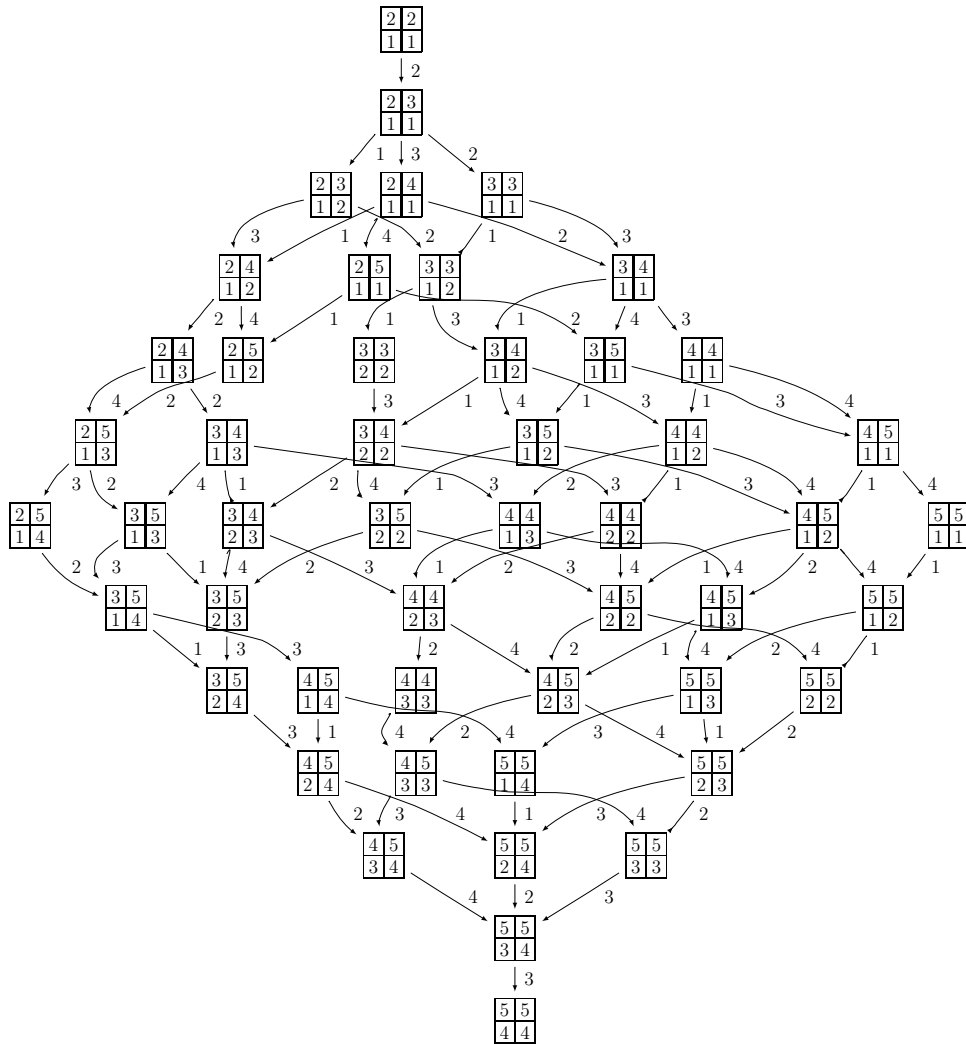


FIGURE 8. Le graphe cristallin de forme $(2, 2)$ en type A_4 ; une flèche f_i remplace un i par un $i + 1$, selon une règle contrainte par la préservation des conditions de croissance au sens large le long des lignes et au sens strict le long des colonnes

La théorie pour les modules de plus haut poids (ce qui est toujours le cas lorsque G est une algèbre de Lie de dimension finie) est essentiellement complète : les modules possèdent toujours une base cristalline, on a des modèles combinatoires (par exemple les tableaux en type A, les chemins de Littlemann en d'autres types), ainsi qu'une caractérisation locale complète des graphes qui sont cristallins [Ste03]. En particulier, tout graphe cristallin $B(\lambda)$ peut être obtenu par produit tensoriel de graphes élémentaires correspondant aux poids fondamentaux. En revanche, pour les algèbres de Lie affine (ou plus généralement de Kac-Moody), de nombreuses questions restent ouvertes comme : quels modules admettent une base cristalline, quels graphes sont cristallins, etc. Par exemple, une partie des travaux récents d'Anne Schilling concerne la construction de modèles combinatoires pour les modules $B^{r,s}$ de Kirillov-Reshetikhin [FSS07, Sch08, OS08], qui sont en lien avec les modèles intégrables sur réseaux en physique mathématique, et semblent jouer le rôle des graphes cristallins élémentaires.

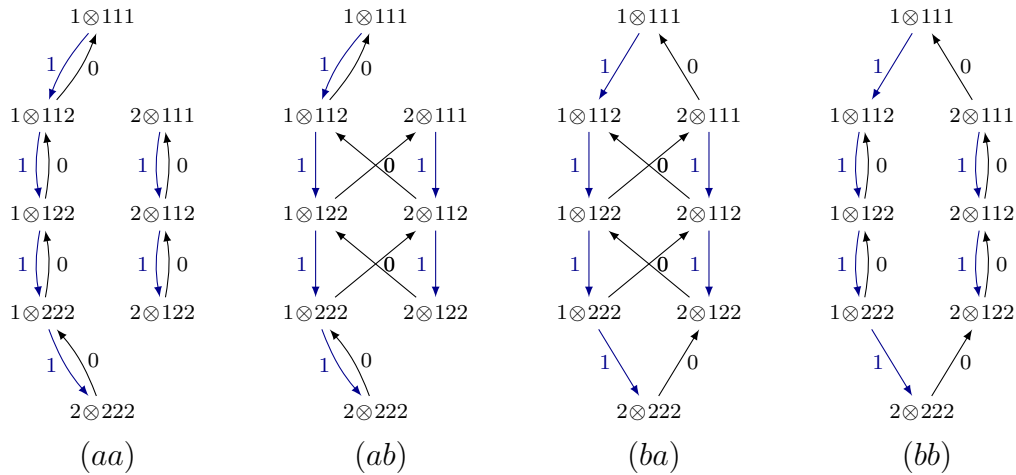


FIGURE 9. Les quatre graphes cristallins affines associés au graphe cristallin classique $B(1) \otimes B(3)$ de type A_1 . Le graphe cristallin $B^{1,1} \otimes B^{3,1}$ correspond à (bb). Les autres ne proviennent pas de $U_q(\widehat{sl_2})$ -modules.

exploration informatique de recherche systématiquement tous les opérateurs de promotion, et j'ai pris en charge l'algorithmique et la programmation de cette part. Dans l'état actuel des connaissances, cette recherche ne peut se faire que par recherche exhaustive, mais en utilisant des heuristiques de coupe de branches (séparation et évaluation), avec propagation de contraintes liées aux propriétés des opérateurs cristallins et exploitation partielle des symétries. Par exemple, pour le cas difficile de $B(1)^{\otimes 4}$ en type A_2 , où l'espace de recherche est a priori de taille 144473849856000, avec $2!3!3! = 72$ symétries, l'algorithme explore 115193 branches en 5 heures et 26 minutes (sur un PC Linux à 2 GHz), utilisant 16Mo de mémoire. Au final, on obtient 8 opérateurs de promotion connexes isomorphes : 9 symétries sur les 72 ont pu être exploitées pour réduire l'espace de recherche. Encore une fois, l'explosion combinatoire est importante ; le rôle de ce travail a été de rendre *possible* (éventuellement au prix de quelques semaines de calcul) l'étude d'exemples non triviaux, sans lesquels nous n'aurions pas établi la conjecture 2.2. Par la suite, et une fois rentré dans le sujet, j'ai apporté quelques clefs essentielles de la démonstration. Étant familier avec les graphes et les problèmes d'isomorphie, j'ai de plus pu simplifier ou abstraire certains passages techniques.

3. Algèbres de Kac et treillis de sous-facteurs

3.1. Sous-facteurs. Ma collaboration avec Marie-Claude David a commencé à mon arrivée à Orsay en 2004, grâce aux mots clefs « tours d'algèbres », « diagrammes de Bratelli » et « calculs dans les algèbres de Hopf ». Marie-Claude David s'intéresse aux inclusions de *facteurs* de type II_1 obtenus via l'action d'une algèbre de Kac (Hopf-von Neumann) de dimension finie.

Un facteur est une algèbre de von Neumann dont le centre est trivial. Les facteurs ont été classifiés par Murray et von Neumann (1943), Connes (1976), et finalement Haagerup. En dimension finie n , il existe à isomorphisme près un unique facteur : $M_n(\mathbb{C})$, et un unique plongement de ce facteur dans lui-même. De même, il existe un unique facteur R dit hyperfini de type II_1 ; on peut le construire comme limite d'une tour d'algèbres $M_n(\mathbb{C})$ emboîtées, ou comme algèbre du groupe S_∞ des permutations de \mathbb{N} à support fini. En revanche, ce facteur de type II_1 admet de multiples plongements dans lui-même.

L'étude des inclusions $N \subset M$ d'indice fini, où N et M sont des facteurs hyperfinis de type II_1 et M est un module de dimension finie sur N a connu un essor considérable [JS97] depuis les articles fondateurs [Jon83, Jon85]. Dans ceux-ci, Vaughan Jones développe les

outils fondamentaux (théorie de l'indice, construction de base, etc.) et en dérive un nouvel invariant polynomial pour les noeuds (et une médaille Fields). En particulier, l'inclusion est décrite par une tour d'algèbres de dimensions finies (la tour dérivée). Souvent, les premières algèbres (profondeur finie), voire la première (profondeur deux) sont suffisantes pour caractériser l'inclusion. *Via* le diagramme de Brattelli de cette tour d'algèbre on peut construire un invariant de l'inclusion qui est un diagramme de Dynkin.

Le rôle de cette tour d'algèbre est similaire à celui du groupe d'automorphisme pour une inclusion de corps. Cela donne lieu à une correspondance de Galois pour le treillis des facteurs intermédiaires $N \subset P \subset M$. En profondeur deux, le rôle du groupe est joué par une *algèbre de Kac* de dimension finie (une algèbre de Hopf involutive semi-simple), ou plus généralement un *groupoïde quantique*; celui du treillis des sous-groupes est joué par le treillis des *sous-algèbres coidéales* (ou pour faire court *coidéaux*) [NV00]. À son tour, chaque coidéale est caractérisé par un élément singulier, son *projecteur de Jones*, de sorte que l'on est ramené à étudier un treillis d'idempotents particuliers de l'algèbre de Kac.

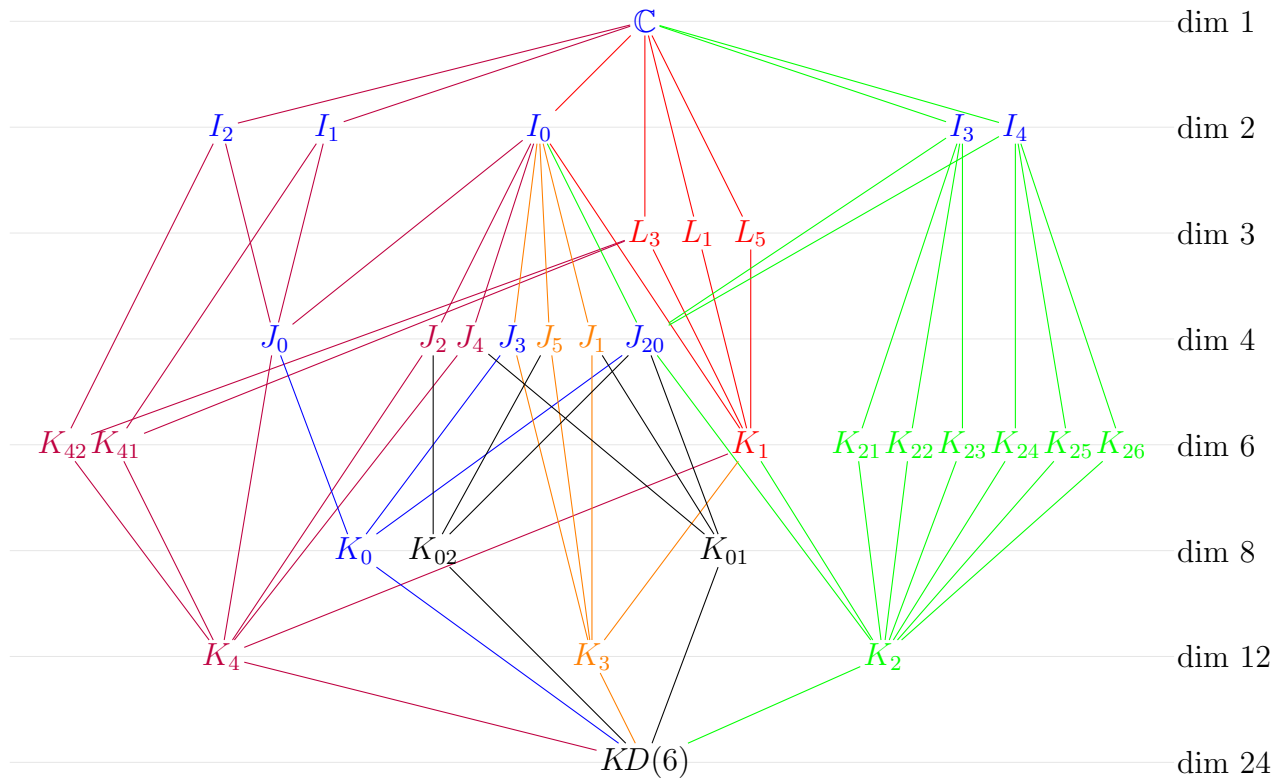
L'apparition des algèbres de Kac dans ce contexte est naturelle. Elles ont en effet été introduites précisément pour donner un cadre commun contenant à la fois les algèbres de groupes (qui sont les algèbres de Kac cocommutatives) et leurs duales (qui sont les algèbres de Kac commutatives) (voir par exemple [ES92]), sachant que dans les deux cas, les coidéaux de l'algèbre correspondent aux sous-groupes.

3.2. Études d'exemples d'algèbres de Kac. En dehors des treillis des groupes, il existe peu d'exemples explicités de treillis non triviaux de sous-facteurs. Marie-Claude David avait auparavant établi à la main le treillis des coidéaux pour la plus petite algèbre de Kac non triviale (de dimension 8) [KP66]. Nous étudions ensemble [DT08] deux familles d'exemples construites par Leonid Vainerman par déformation d'algèbres de groupes de dimension $4n$ [Vai98] par des 2-pseudo cocycles. En dehors des formules pour les opérations de Kac (coproduit, coinvolution, etc.), rien n'était connu sur ces algèbres, pour la simple raison que les calculs nécessaires à leur exploration sont très lourds. L'exploration informatique nous a permis de conjecturer, puis de démontrer, que ces familles sont en fait isomorphes pour n pair. Nous avons aussi pu obtenir leurs groupes d'automorphismes et montrer que, pour n impair, les algèbres de ces deux familles sont autoduales. Nous avons une description complète du treillis en petite dimension (en gros $n \leq 7$, voir figure 10 pour un exemple en dimension 24), pour n premier et, conjecturalement, pour n impair, ainsi que des résultats partiels pour n pair. Nous en dérivons, par le calcul des diagrammes de Brattelli, les graphes principaux d'un certain nombre d'inclusions.

3.3. Exploration informatique. Cette collaboration est un cas exemplaire de ma stratégie favorite. Marie-Claude David a apporté un sujet intéressant, une culture générale, un savoir calculer à la main impressionnant et, point de départ indispensable, un problème concret de calcul. J'ai apporté un outil et un savoir explorer dans un domaine où, contrairement à la combinatoire algébrique, l'ordinateur n'est pas un outil traditionnel. J'étais donc particulièrement attiré par les questions « qu'est-ce qui est calculable ? » « jusqu'où ? », « dans quel but ? ». Comme d'habitude, l'ordinateur a été un *outil d'exploration* et en particulier un *outil créatif*, permettant avec les bonnes questions et les bons algorithmes de deviner ex-nihilo des formules d'isomorphisme comme :

$$\begin{aligned}\phi(\lambda(a)) &= a + 1/4(a - a^{-1})(a^n - 1)(1 - ba^m), \\ \phi(\lambda(b)) &= 1/2[b(a^n + 1) + i(a^n - 1)]a^m,\end{aligned}$$

À titre plus exceptionnel, dans ce domaine où les lemmes tendent à être techniques et les formules intrinsèquement lourdes, il s'est révélé être aussi un *outil de preuve* quasiment indispensable, avec l'élaboration de stratégies pour ramener le cas général à n suffisamment

FIGURE 10. Exemple de treillis de coidéaux : $KD(6)$

petit pour pouvoir vérifier sur machine. Même à la machine, les calculs sont difficiles. Ils nécessitent tout d'abord une réflexion algorithmique. De fait, la construction des treillis de coidéaux est encore loin d'être automatisée. Ils nécessitent aussi une infrastructure d'un haut niveau d'abstraction (voir les exemples en section 2.4). L'effort de développement que j'ai consenti à cette occasion a permis de compléter considérablement l'algorithmique et l'infrastructure pour les algèbres de Hopf et la théorie des représentations des algèbres de dimension finie dans \ast -Combinat ; cela a déjà trouvé d'autres applications.

3.4. Perspectives. L'étape suivante, que nous entreprenons avec Leonid Vainerman et son étudiante en thèse Camille Mével, est l'étude des exemples plus généraux venant de *groupoïdes quantiques*, qui correspondent à des inclusions non irréductibles. Cela permettra d'obtenir, pour de petites tailles, tous les treillis provenant d'inclusions d'indice fini et de profondeur finie ; celles-ci peuvent en effet toujours être réalisées comme inclusions intermédiaires d'inclusions de profondeur 2, en général non irréductibles.

De manière générale, les domaines environnants sont pratiquement vierges d'exploration informatique. Cela ouvre la porte à une multitude de problèmes intrinsèquement simples, non résolus jusqu'ici faute de moyens techniques appropriés, et posant des problèmes d'algorithmique ou de modélisation intéressants. Une mine d'or potentielle, en particulier pour des étudiants aimant à la fois les mathématiques et l'informatique. Par exemple : Leonid Vainerman m'a demandé comment déterminer le groupe d'automorphismes d'une certaine algèbre obtenue par produit croisé de deux algèbres déformées du groupe symétrique \mathfrak{S}_n . Pour $n = 2$, le résultat est trivial, les groupes étant commutatifs. Résoudre $n = 3$ (dimension 36) a de bonnes chances de suggérer la solution pour tout n . Pour cela, il faudrait trouver un algorithme de complexité raisonnable pour calculer tous les isomorphismes entre deux algèbres de Hopf (ou de Kac) de dimension finie. De même, il serait urgent de tester ce que les outils de géométrie algébrique

réelle peuvent apporter à l'automatisation de la construction automatique des projecteurs de Jones.

4. Polynômes harmoniques pour les opérateurs de Steenrod

Assistant en juin 2000 à l'école « Interactions between Algebraic Topology and Invariant Theory » à Ioannina, mon attention avait été captivée lorsque Reg Wood [Woo97, Woo98, Woo01] présenta une conjecture que l'on peut réécrire sous la forme :

Conjecture 4.1. *Le sous-espace des polynômes p de $\mathbb{Q}[x_1, \dots, x_n]$ satisfaisant pour tout k l'équation aux dérivées partielles linéaire :*

$$\left(\left(1 + x_1 \frac{\partial}{\partial x_1} \right) \frac{\partial}{\partial x_1} + \dots + \left(1 + x_n \frac{\partial}{\partial x_n} \right) \frac{\partial}{\partial x_n} \right)^k p = 0$$

est isomorphe à la représentation régulière graduée du groupe symétrique. En particulier, il est de dimension $n!$.

En effet, comme l'avait mentionné Reg Wood, en oubliant les termes $x_i \frac{\partial}{\partial x_i}$, ce sous-espace devient celui des *harmoniques des polynômes symétriques* (une réalisation des coinvariants du groupe symétrique) ; dans ce cadre, le résultat est bien connu, avec une multitude de techniques de démonstration (voir par exemple [GH94]).

Cette conjecture de Reg Wood n'est pas un phénomène isolé, bien au contraire. Elle est fortement liée à toute une mouvance de problèmes où interviennent des polynômes harmoniques, problèmes que j'avais déjà rencontrés lors de mes travaux sur les algèbres d'invariants de groupes de permutations [AB03, ABB04, Hai01]. En gros, d'où vient-elle ? Les algèbres d'invariants de groupes finis en petite caractéristique apparaissent naturellement comme cohomologie d'espaces topologiques. Les opérateurs de Steenrod (d'origine topologique, mais de nature algébrique : ils engendrent une algèbre de Hopf, l'algèbre de Steenrod) agissent sur les invariants, permettant d'en construire de nouveaux. Un programme important est de décrire tous les invariants atteignables de la sorte. Rien que pour le groupe trivial, c'est-à-dire en agissant simplement sur les polynômes, le problème est difficile ; à l'heure actuelle, il n'y en a même pas de description conjecturale. La conjecture de Reg Wood est l'analogue de ce problème en caractéristique 0. Sa résolution donnerait certainement des pistes d'attaque pour le cas modulaire.

Étudier cette conjecture était pour moi l'occasion rêvée d'utiliser sérieusement, et donc de mieux maîtriser, un grand nombre d'outils. D'abord ceux de la combinatoire algébrique phalanstérienne : représentations du groupe symétrique et des algèbres de Lie, déformations non commutatives et interpolations entre algèbres de Hopf (groupes quantiques), polynômes de Schubert et opérateurs sur les polynômes (différences divisées avec les algèbres de Hecke affine en arrière plan). Et ensuite des outils du calcul formel : algorithmique dans les algèbres de Weyl ou de Ore et bases de Gröbner semi-commutatives. C'était aussi un bon problème pour commencer à travailler avec Florent Hivert, ce que nous souhaitions de longue date.

Afin d'interpoler entre les deux problèmes, nous avons défini la q -algèbre de Steenrod par déformation non commutative de l'algèbre des polynômes symétriques. Cela revient à introduire un facteur q devant les termes $x_i \frac{\partial}{\partial x_i}$ des équations de la conjecture 4.1. On appelle q -harmoniques les solutions de ces équations déformées ; à $q = 0$, ce sont les harmoniques usuels pour les polynômes symétriques, et à $q = 1$ les polynômes de la conjecture 4.1. Cela permet d'utiliser des arguments de spécialisation pour réduire cette conjecture comme suit.

Proposition 4.2 (Hivert, T. [HT04a]). *Pour démontrer la conjecture 4.1, il suffit de montrer que :*

- (1) *La dimension des q -harmoniques est au moins $n!$.*
 (2) *La dimension des 1-harmoniques est au plus $n!$.*

Jusqu'ici, la conjecture avait été testée complètement jusqu'à $n = 3$. La méthode brutale pour tester la conjecture nécessite de faire de l'algèbre linéaire sur les polynômes de degré $\binom{n}{2}$, ce qui fait un espace de dimension $\binom{\binom{n}{2}+n-1}{n-1}$. L'exploitation des symétries pour décomposer l'espace des polynômes en petits sous-espaces (composantes de Garnir) nous a permis de mener une exploration informatique complète jusqu'à $n = 5$, et partielle jusqu'à $n = 9$. Cela a confirmé la conjecture pour $q = 1$, et en fait pour tout q sauf pour quelques rationnels de la forme $-\frac{a}{b}$ avec a et b petits. Mais la suite ne s'est pas déroulée comme prévu : même si nous avons obtenu quelques résultats dans des cas particuliers (par exemple en petit degré), la conjecture a résisté à tous nos efforts, toutes les techniques du cas usuel se révélant inapplicables [HT04a].

En 2006, Adriano Garsia (grand spécialiste des polynômes harmoniques) trouva cette conjecture magnifique lorsque nous la lui avons exposée. Il s'y est depuis intéressé avec François Bergeron et Nolan Wallach. Ceux-ci ont obtenu des résultats partiels supplémentaires, comme une borne supérieure sur la dimension du sous-espace des solutions. Ils ont aussi découvert que la fameuse ex-conjecture de $n!$ concernant les harmoniques pour l'action diagonale de \mathfrak{S}_n sur deux jeux de paramètres semblait aussi admettre une généralisation de type Steenrod. Cela renforce l'idée que la conjecture de Reg Wood n'est pas un accident isolé.

Nous avons nous-mêmes mis de côté cette conjecture en attendant que l'étude d'un autre problème nous apporte le bon éclairage. Nous restons en effet convaincus que cette conjecture n'est pas intrinsèquement difficile et qu'il suffira de poser la bonne question à l'ordinateur pour dérouler une preuve élémentaire et constructive.

*-Combinat, boîte à outils pour l'exploration informatique

Le chapitre qui vient est de nature différente des précédents. En particulier, il ne contient pas de résultats de recherche à proprement dit. J'y décris différents aspects de mon travail autour du projet logiciel *-Combinat que je coordonne depuis sa création en 2000.

La mission de ce projet est de fournir une boîte à outils extensible pour l'exploration informatique en combinatoire algébrique, et promouvoir la mutualisation de code entre les chercheurs de ce domaine.

Mon rôle, secondé par Florent Hivert, est multiple : choix de la plateforme et du modèle de développement, animation de la communauté, repérage d'intérêts communs et mise en relation, formation permanente, assistance aux développeurs (conception, modélisation mathématique, algorithmique), veille sur la qualité (tests, revues de code), développement d'une vision globale garante de la cohérence interne, mise en place de l'architecture logicielle (compilation, documentation, distribution, tests), travail de fond sur le langage (paradigmes et idiomes), veille technologique (repérage des techniques et composants les plus intéressants à intégrer), promotion et valorisation. En bref, libérer les autres contributeurs des contingences informatiques pour qu'ils puissent se concentrer sur leur tâche : traduire leur expertise mathématique en composants logiciels réutilisables à même de résoudre naturellement et efficacement leurs problèmes de calculs. Enfin, je me dois d'insuffler une dynamique, en étant en permanence en première ligne, tout en rentabilisant mes propres développements dans des projets de recherche.

Après un historique (section 1), et une démonstration courte (section 2), je présente quelques problématiques spécifiques à nos besoins en calculs tant du point de vue du modèle de développement (section 3) que de la conception (sections 4 et 5).

Dans ce domaine, nous sommes des praticiens. Autant que possible, nous avons essayé de réutiliser des techniques et outils préexistants. Mais dans certains cas, nous avons été amenés à développer des solutions originales. Leur conception a été guidée puis validée par notre expérience sur le terrain, tout en restant au plus près des mathématiques. Il serait maintenant souhaitable d'avoir une analyse théorique de notre travail, par exemple dans le cadre d'études de cas par des doctorants en informatique, pour expliquer pourquoi nos solutions sont effectivement adaptées aux besoins ou, au contraire, pour en proposer de meilleures.

Ce chapitre se poursuit avec une étude de cas : comment une collaboration internationale de chercheurs sur le modèle du libre permet sur le long terme de bouleverser un outil effectif essentiel : les objets décomposables ou plus généralement les espèces combinatoires (section 6). Enfin, on trouvera en section 7 une liste de publications ayant utilisé ou concernant MuPAD-Combinat.

1. *-Combinat, Sharing algebraic combinatorics since 2000

1.1. Pourquoi MuPAD-Combinat. Toutes mes recherches ont en commun l'utilisation d'outils informatiques, et notamment du calcul formel, dans des domaines propices aux explosions combinatoires. L'exploration informatique sert de guide, suggérant des conjectures, ou au contraire produisant des contre-exemples. Elle permet d'étudier des exemples suffisamment conséquents pour être représentatifs ; ces exemples sont le plus souvent intraitables à la main, et souvent hors de portée des algorithmes classiques. Cela nécessite de trouver les outils mathématiques appropriés pour développer de nouveaux algorithmes, sachant que seule l'efficacité de l'implantation finale décide de la pertinence de ces outils. Réciproquement, il est indispensable

que le logiciel modélise les objets mathématiques au plus près, afin de pouvoir exprimer des problèmes complexes dans un langage naturel.

Bien entendu, mener à bien de tels calculs sous-entend un important travail de programmation, et requiert une large panoplie de techniques. Ainsi, lors de ma thèse, j'avais été amené à utiliser les logiciels suivants : **Magma** (invariants, groupes de permutations), **GAP** (groupes, représentations), **Cocoa**, **Gb**, **FGb**, **Macaulay** (bases de Gröbner efficaces), **Nauty**, **Graphlet**, **Lydia** (théorie des graphes), **ALP**, **Linbox** (algèbre linéaire creuse exacte), **Maple** (calcul formel) et ses modules (**combstruct** : objets décomposables, **gfun** : séries génératrices, **ACE**, **SF** : fonctions symétriques, etc.), **MuPAD** (calcul formel) et son module *mu-EC* (fonctions symétriques). Pour aller plus loin j'avais besoin non seulement de les utiliser simultanément, mais aussi de les combiner aisément les uns avec les autres. Une tâche d'*intégration* bien trop ambitieuse pour un seul individu.

À mon sens, il y avait et il y a toujours un besoin, à l'échelle de la combinatoire algébrique, pour une boîte à outils de référence, libre, largement diffusée, développée par une communauté de chercheurs pour leurs propres besoins et à l'échelle internationale. En bref, jouant le rôle fédérateur de **GAP** pour la théorie des groupes. Elle doit de plus être basée sur une plateforme généraliste et un langage de programmation répandu et de qualité (en particulier orienté objet), réutiliser le maximum de composants existants, et permettre une synergie avec les domaines avoisinants (calcul formel, graphes, groupes, etc.). Il est à noter qu'une telle boîte à outils, en tout cas dans ces fondamentaux, est utile pour l'enseignement. Elle a aussi des applications potentielles en physique mathématique (calculs dans les algèbres de Hopf et opérades venant des problèmes de renormalisation). De nombreux logiciels existaient, certains importants et bien établis, mais aucun ne répondait au cahier des charges.

1.2. Les débuts de MuPAD-Combinat. J'en avais besoin pour mes recherches et je n'étais pas le seul. C'est un projet qui m'intéressait, et j'en avais peut-être les compétences, à condition de m'associer aux bons experts. J'ai cherché des partenaires, établi un cahier des charges détaillé, et longtemps soupesé les plateformes disponibles [**HT04b**]. Ce fut la naissance de **MuPAD-Combinat** en décembre 2000, d'abord avec Florent Hivert et le soutien du Phalanstère de Marne-la-Vallée et de l'équipe du système de calcul formel **MuPAD** à Paderborn. Puis progressivement se sont rajoutés François Descouens, Teresa Gomez Diaz, Jean-Christophe Novelli du Phalanstère, ainsi que Christophe Carré, Éric Laugerotte, Houda Abbad et Janvier Nzeutchap de Rouen, Frédéric Chapoton de Lyon ; Patrick Lemeur de Montpellier. L'équipe s'est enfin internationalisée avec Xavier Molinéro de Barcelone, Mike Zabrocki de Toronto, Anne Schilling et Quiang Wang de Davis. Sans compter indirectement tous les auteurs des logiciels que nous avions intégrés (**ACE**, μ -**EC**, **CS**, **Symmetriza**, **Nauty**, etc.).

La collaboration avec Anne Schilling, commencée lors de FPSAC 2006 a été exemplaire. Ayant une bonne expérience de la programmation et de l'exploration informatique, elle souhaitait implanter dans **MuPAD-Combinat** une bibliothèque sur les graphes cristallins. De nombreux courriels suivirent, où d'un côté elle m'expliquait la théorie et les règles de calculs dans les graphes cristallins, et en retour je lui donnais des conseils de conception et d'algorithmique. Trois mois plus tard, elle avait le matériel nécessaire pour tester ses conjectures sur les graphes cristallins de Kirillov-Reshetikhin en type D affine [**Sch08**], et fit le commentaire suivant sur une liste de diffusion à propos de son expérience avec **MuPAD-Combinat** : « For me personally, it has been great, since it already has a lot of features and, having the support from Nicolas and others, helped me to write code that I needed in such a way that it can be reused by others, and I myself could build on it (which has never been the case before with programs I had written in Maple or Mathematica). Nicolas definitely has the ability to factor out main features, which is necessary for such a large scale project. »

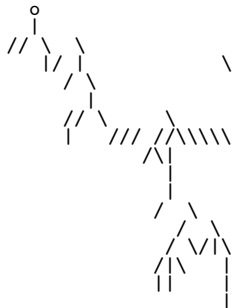
Nous pouvons aussi juste compter ces arbres, une opération beaucoup plus rapide² :

```
>> trees::count(6)
```

42

Voici un arbre aléatoire. L'affichage en ASCII 2D est loin d'être parfait, mais la structure de données interne, elle, est robuste.

```
>> trees::random(50)
```



Les algorithmes sous-jacents ne sont pas spécifiques aux arbres. Ils s'appliquent à toute autre famille d'objets pouvant être définis récursivement par une grammaire. Voici par exemple la relation de récurrence pour le nombre d'arbres binaires. Elle est calculée automatiquement à partir de la grammaire pour ces arbres, et est exploitée pour faire du comptage efficace :

```
>> r := binaryTrees::grammar::recurrenceRelation():
>> assume(n>0):
>> u(n) = factor(op(solve(r, u(n)),1))
```

$$u(n) = \frac{2 u(n-1) (2n-1)}{n+1}$$

On reconnaît bien évidemment la récurrence usuelle des nombres de Catalan.

2.2. Graphes cristallins. Nous allons maintenant manipuler des graphes cristallins, comme ceux de la section 2. Ce sont encore des classes combinatoires, avec des opérations algébriques supplémentaires. Nous définissons deux graphes cristallins de Kirillov-Reshetikhin de type $A_2^{(1)}$:

```
>> C1 := crystals::kirillovReshetikhin(2,2,["A",2,1]):
>> C2 := crystals::kirillovReshetikhin(1,1,["A",2,1]):
```

Leurs éléments sont des tableaux semi-standard :

```
>> C1::list()
```

```
-- +---+---+ +---+---+ +---+---+ +---+---+ +---+---+ +---+---+ --
| | 2 | 2 | | 2 | 3 | | 2 | 3 | | 3 | 3 | | 3 | 3 | | 3 | 3 | | | | | | |
| +---+---+ | +---+---+ | +---+---+ | +---+---+ | +---+---+ | +---+---+ |
| | 1 | 1 | | | 1 | 1 | | | 1 | 2 | | | 1 | 1 | | | 1 | 2 | | | 2 | 2 | |
-- +---+---+ +---+---+ +---+---+ +---+---+ +---+---+ +---+---+ --
```

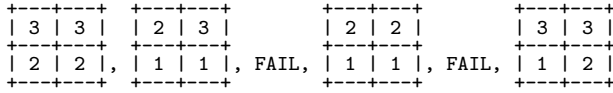
sur lesquels agissent des opérateurs montants e_i et descendants f_i :

```
>> x := C1::list()[3]
```

```
+---+---+
| 2 | 3 |
| +---+---+
| 1 | 2 |
| +---+---+
```

```
>> x::e(0), x::e(1), x::e(2), x::f(0), x::f(1), x::f(2)
```

2. En tout cas pour n grand; mais alors, on n'obtient pas d'information fondamentale sur l'univers.



Après avoir manipulé leurs éléments, nous faisons des calculs sur les graphes cristallins eux même. Par exemple, nous vérifions que le graphe C_1 a un unique automorphisme et que les graphes C_1 et C_2 ne sont pas isomorphes :

```
>> C1::isomorphisms(C1)
      [proc g(x) ... end]

>> C1::isomorphisms(C2)
      □
```

Comme son nom l'indique, la méthode `C1::isomorphisms` renvoie la liste de tous les isomorphismes, sous forme de fonctions que l'on pourrait appliquer aux éléments de C_1 .

Pour conclure, nous construisons le produit tensoriel des graphes cristallins C_1 et C_2 (en fait leur produit cartésien, ce qui correspond au produit tensoriel des modules dont ils indexent les bases), et en demandons une représentation graphique (en \LaTeX). Ce type de représentation est bien entendu un outil important d'exploration.

```
>> operators::setTensorSymbol("#"):
>> C := C1 # C2:
>> viewTeX(C::TeXClass())
(voir figure 11 pour le résultat)
```

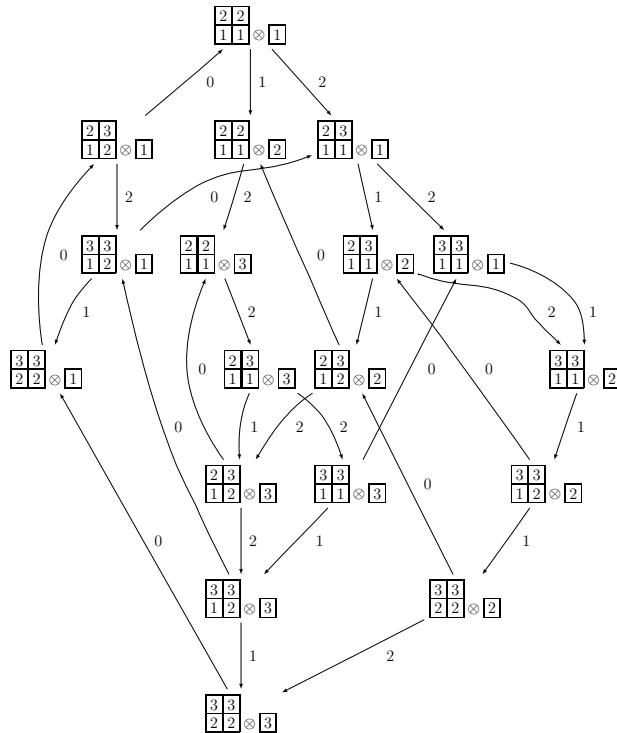


FIGURE 11. Le graphe cristallin affine $B^{2,2} \otimes B^{1,1}$ en type $A_2^{(1)}$

Cet exemple reste un jouet. De par leur construction, les graphes cristallins sont de taille explosive. Afin de permettre des calculs poussés, l'implantation utilise systématiquement l'évaluation paresseuse. Ainsi, seule la partie du graphe effectivement étudiée est dépliée en mémoire.

Les calculs précédents illustrent une autre spécificité importante de nos besoins. Appelons *domaine* un ensemble muni d'opérations (par exemple l'anneau des matrices $M_n(\mathbb{R})$, l'anneau de polynômes $\mathbb{C}[x, y, z]$, les fonctions de \mathbb{C} dans \mathbb{C}). En calcul numérique ou formel, les manipulations concernent principalement les objets de ces domaines (pivot de Gauss, base de Gröbner, intégration). En revanche, en combinatoire algébrique (mais aussi en théorie des groupes), la richesse des calculs vient le plus souvent de la manipulation *simultanée* des éléments des domaines (les tableaux avec l'action des opérateurs cristallins) et des domaines eux-mêmes (les graphes cristallins). Cela sera discuté dans la section 4.3

2.3. Algèbre de mélange sur les arbres. Nous avons vu que les classes combinatoires ont une interface systématique. Cela est bien entendu souhaitable pour l'utilisateur. Mais cela permet surtout de les utiliser comme briques logicielles pour des constructions plus avancées. Nous illustrons cela en donnant une implantation de l'algèbre graduée dont la base est indexée par les arbres et le produit est induit par le produit de mélange sur les arbres (c'est la réalisation de l'algèbre de Loday-Ronco dans la base p) :

```

1  domain ShuffleAlgebraOnTrees(R = Dom::ExpressionField(): Cat::Ring)
2      category Cat::GradedAlgebraWithBasis(R);
3      inherits Dom::FreeModule(trees, R);
4
5      oneBasis := trees::zero;
6
7      mult2Basis :=
8      proc(t1: trees, t2: trees)
9          local t;
10         begin
11             dom::plus(dom::term(t) $ t in trees::shuffle(t1,t2));
12         end_proc;
13 end_domain

```

La ligne 1 précise que la construction de l'algèbre prend un paramètre R , l'anneau de base, avec une valeur par défaut, le corps des expressions. La ligne 2 fait une promesse au système : l'objet implanté est une algèbre graduée. La ligne 3 précise la structure de données pour les éléments : des vecteurs sur R indexés par les arbres. Il reste à tenir les promesses, en spécifiant l'unité (ligne 5) et la règle de calcul du produit, ici exprimée sur la base. La syntaxe nécessite un apprentissage, mais au final tout le code est signifiant mathématiquement.

2.4. Quelques calculs dans les algèbres de dimension finie. Nous poursuivons en présentant quelques calculs typiques sur des algèbres. Ici, nous travaillons sur l'algèbre de Kac $KD(n)$ obtenue par déformation de l'algèbre du groupe diédral D_{2n} étudiée en section 3 du chapitre 2. Il n'est pas nécessaire de connaître les détails, l'objectif des exemples étant uniquement d'illustrer le niveau d'abstraction typique des calculs que nous souhaitons mener.

L'algèbre $KD(n)$ est pour l'instant implantée dans une feuille de travail séparée que l'on charge ici :

```

>> read("experimental/2005-09-08-David.mu"):
KD(3) modélise l'algèbre abstraite; après l'avoir construite

>> KD3 := KD(3):

```

nous demandons quelles sont ses propriétés connues :

```

>> KD3::categories

```

```
[Cat::HopfAlgebraWithSeveralBases(Q(II, epsilon)),
TwistedDihedralOrQuaternionGroupAlgebra(3),
Cat::AlgebraWithSeveralBases(Q(II, epsilon)),
Cat::Algebra(Q(II, epsilon)),
Cat::ModuleWithSeveralBases(Q(II, epsilon)),
Cat::Ring, Cat::Module(Q(II, epsilon)),
Cat::DomainWithSeveralRepresentations, Cat::Rng, Cat::SemiRing,
Cat::LeftModule(KD(3, Q(II, epsilon))),
Cat::LeftModule(Q(II, epsilon)), Cat::RightModule(Q(II, epsilon)),
Cat::UseOverloading, Cat::FacadeDomain, Cat::SemiRng, Cat::Monoid,
Cat::AbelianGroup, Cat::SemiGroup, Cat::CancellationAbelianMonoid,
Cat::AbelianMonoid, Cat::AbelianSemiGroup, Cat::Object,
Cat::BaseCategory]
```

À quelques exceptions près, ce sont des informations mathématiques.

Pour faire des calculs, nous avons besoin d'une représentation concrète de cette algèbre. Celle utilisant la base du groupe est modélisée par `KD3 : :group`. Nous commençons par introduire des notations courtes pour ses générateurs :

```
>> [aa,bb] := KD3::group::algebraGenerators::list()
          [B(a), B(b)]
```

Les quelques calculs suivants, dans la base du groupe, montrent que le produit n'est pas déformé :

```
>> bb^2
          B(1)

>> aa^2, aa^6, bb*aa
          B(a^2), B(1), B(a^5 b)
```

```
>> (1 - aa^3)*(bb + aa^3) + 1/2*bb*aa^3
          1/2 B(a^3 b)
```

De part la théorie des représentations, l'algèbre $KD(3)$ admet une autre représentation concrète comme algèbre de matrices par blocs, modélisées par `KD3 : :matrix`. Nous utilisons ici l'isomorphisme entre les deux représentations (transformée de Fourier, d'où la présence de `epsilon` qui représente une racine ϵ de l'unité).

```
>> KD3::M(aa + 2*bb)
```

```
+-----+
| 3,  0,  0,  0,  0,      0,      0,      0      |
| 0, -1,  0,  0,  0,      0,      0,      0      |
| 0,  0, -3,  0,  0,      0,      0,      0      |
| 0,  0,  0,  1,  0,      0,      0,      0      |
| 0,  0,  0,  0, epsilon,  2,      0,      0      |
| 0,  0,  0,  0,  2,  1 - epsilon,  0,      0      |
| 0,  0,  0,  0,  0,      0,  epsilon - 1,  2      |
| 0,  0,  0,  0,  0,      0,      2,  -epsilon  |
+-----+
```

Cet isomorphisme n'a été explicitement implanté que dans un sens (en donnant l'image de a et b); l'isomorphisme inverse est construit automatiquement par inversion de matrices.

Le coproduit est déformé par un cocycle du sous-groupe $\langle a^3, b \rangle$. Pour les éléments de ce sous-groupe, le coproduit n'est donc pas déformé ($\#$ dénote le produit tensoriel \otimes) :

```
>> coproduct(aa^3), coproduct(bb)
          B(a^3) # B(a^3), B(b) # B(b)
```

Par contre, le coproduit de a est compliqué :

```
>> coproduct(aa)
```

$$\frac{1}{16} B(a^4 b) \# B(a^4 b) + \left(-\frac{\text{II}}{8} - \frac{1}{16} \right) B(a^4 b) \# B(a^5) + \left(\frac{\text{II}}{8} - \frac{1}{16} \right) B(a^4 b) \# B(a^2 b) + \dots$$

... 100 lignes coupées ...

$$-\frac{1}{16} B(a) \# B(a^2) + \frac{7}{16} B(a) \# B(a)$$

Pour illustrer des calculs tensoriels typiques, nous vérifions que l'antipode est correcte, c'est-à-dire que la formule $\mu \circ (\text{id} \otimes S) \circ \Delta$ redonne bien la counité de l'algèbre de Hopf :

```
>> K := KD3::G: // un simple raccourci
>> checkAntipode := K::mu @ ( K::id # K::antipode ) @ K::coproduct:
```

```
>> checkAntipode(x) $ x in K::basis::list()
```

```
B(1), B(1), B(1), B(1), B(1), B(1), B(1), B(1), B(1), B(1), B(1), B(1)
```

Notre étude concernait les coidéaux de $KD(3)$, c'est à dire des sous-algèbres « stables à droite » pour le coproduit. Notons e les unités matricielles de l'algèbre.

```
>> e := KD3::e:
```

Nous calculons maintenant une base du coidéal $K_2 := I(e_1 + e_2)$ engendré par $e_1 + e_2$:

```
>> K2basis := coidealAndAlgebraClosure([ e(1)+e(2) ])
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

On reconnaît $e_1 + e_2$ dans le premier élément.

Manipuler directement une telle base n'est pas commode pour étudier les propriétés algébriques de K_2 . Nous construisons donc un objet qui modélise cette sous-algèbre, en promettant à MuPAD qu'il s'agit d'une sous-algèbre de Hopf (ce que nous savions par ailleurs).

```
>> K2 := Dom::SubFreeModule(K2basis,
>> [Cat::FiniteDimensionalHopfAlgebraWithBasis(KD3::coeffRing)]):
```

Nous pouvons maintenant demander si cette sous-algèbre est commutative ou cocommutative :

```
>> K2::isCommutative(), K2::isCocommutative()
```

```
TRUE, FALSE
```

Nous en déduisons que c'est forcément l'algèbre duale d'une algèbre de groupe que nous souhaitons retrouver. Pour cela, nous construisons l'algèbre duale de K_2 :

```
>> K2dual := K2::Dual():
```

L'algorithmique sous-jacente est triviale : les opérateurs sont définis par simple transposition de ceux de l'algèbre originale. Mais il est pratique de ne pas avoir à s'en soucier.

Calculons les éléments de type groupe de cette algèbre (ici, l'algorithmique est non triviale!) :

```
>> K2dual::groupLikeElements()
```

```
[B([1, 1]), B([7, 7]), B([3, 3]), B([8, 8]), B([5, 5]) + -II B([6, 5]), B([5, 5]) + II B([6, 5])]
```

Ces éléments forment un groupe, le groupe intrinsèque. Pour le déterminer, il reste à reconnaître la règle de produit.

```
>> G := K2dual::intrinsicGroup():
>> G::list()
```

```
[[], [1], [1, 1], [2], [1, 2], [1, 1, 2]]
```

G modélise le groupe; ses éléments ont été exprimés en fonction de générateurs choisis au hasard. C'est brutal, MuPAD n'ayant que des fonctionnalités élémentaires pour les groupes. Cela est cependant suffisant ici : on reconnaît à nouveau un groupe diédral. Nous finissons en calculant à titre de vérification quelques informations sur la théorie des représentations :

```
>> K2dual::isSemiSimple()
```

```
TRUE
```

```
>> K2dual::simpleModulesDimensions()
```

```
[2, 1, 1]
```

Résumons ce que nous avons vu. Pour explorer rapidement des structures algébriques (par exemple une algèbre de dimension finie), nous avons besoin de poser, si possible en quelques minutes, des questions d'un relativement haut niveau d'abstraction (l'algèbre est-elle semi-simple?).

Cela nécessite de modéliser simultanément l'algèbre *et* ses éléments, en s'appuyant sur une représentation concrète; celle-ci provient usuellement d'une construction basée sur des modèles combinatoires. Souvent, l'algorithmique sous-jacente est alors une simple agglomération d'éléments simples (Euclide pour l'arithmétique dans une extension algébrique, quelques produits s'appuyant sur des règles combinatoires, un pivot de Gauss). Il serait cependant trop fastidieux (si ce n'est impossible en pratique) de devoir s'y ramener explicitement à chaque fois.

Nous avons donc besoin d'un système souple et expressif, permettant de poser naturellement des questions sur des constructions avancées, obtenues par composition de briques de base. Vue la multitude de modèles combinatoires intéressants, les briques et les constructions doivent être les plus génériques possible, sans cependant entraver la complexité algorithmique. À ce titre, la conception joue un rôle tout aussi important que l'algorithmique. Enfin, il devrait être facile de faire appel à des outils optimisés d'autres domaines mathématiques (théorie des groupes, algèbre linéaire creuse, etc).

3. Modèle de développement

3.1. Quelques mots d'ordre de *-Combinat. Afin d'éclairer la philosophie de *-Combinat, nous donnons, en les expliquant, quelques-uns de ses mots d'ordre.

Toute ligne de code doit être écrite en vue d'une application immédiate à un projet de recherche, tout en ayant une vision à long terme.

L'objectif est de garantir que le code est utile, qu'il a été testé, que la conception et la fonctionnalité ont été validées par la pratique, et que le travail fourni sera valorisé scientifiquement par des publications. Ce dernier point est essentiel pour maintenir la motivation, et éviter des frustrations ou des frictions quant à la paternité du code.

Dans le même temps, il faut viser un code générique et réutilisable, afin que les investissements soient rentabilisés sur le long terme.

Écrit par une communauté informelle de chercheurs, pour les chercheurs.

Ce point est avant tout un corollaire du point précédent. De plus, des relations informelles, dans une certaine convivialité, encouragent l'entraide, les échanges, et favorisent l'émergence d'un corpus d'expertise qui bénéficie à tous.

Sous licence libre.

Trop de logiciels sont morts suite à une volonté politique de leurs institutions de tenter une commercialisation inadéquate avec fermeture du code. Toute application industrielle de *-Combinat est la bienvenue! Notre longue et fructueuse collaboration avec SciFace est la meilleure preuve de notre pragmatisme. Il serait par exemple parfaitement acceptable de commercialiser des services d'expertise autour de *-Combinat si le besoin s'en faisait sentir. Mais l'objectif premier est de mutualiser les efforts de développement des chercheurs pour la recherche. Le contrôle du logiciel doit rester entre les mains des chercheurs. Pour cela la meilleure garantie, outre l'utilisation d'une licence libre type GPL, est de multiplier les institutions concernées et les sources de financements pour qu'aucune ne soit dominante.

Le coût de développement de *-Combinat est essentiellement humain. Que chaque chercheur y participe sur la base du volontariat, et à la hauteur de ses besoins propres, justifie en soi l'investissement de son institution. Cela permet à *-Combinat de fonctionner avec un budget direct très limité.

Fédérateur à l'échelle internationale.

Trop de logiciels sont restés confidentiels à cause de rivalités entre communautés. Il est important d'éviter qu'une communauté ne se l'approprie entraînant un rejet par les autres. Pour cela, il faut franchir rapidement les frontières (géographiques ou thématiques), par la base, c'est-à-dire en convaincant un par un les chercheurs.

Géré par des chercheurs permanents.

Trop de doctorants ont été sacrifiés, la charge de développement ne leur permettant pas de publier. Réciproquement, trop de logiciels ont été jetés après le départ de leur responsable ex-doctorant. Le travail de fond sur une bibliothèque de cette ampleur peut exiger des investissements massifs rentables uniquement sur le long terme. Maintenir dans le même temps une productivité scientifique régulière demande de l'expérience. En revanche, les doctorants sont bien placés pour intervenir en périphérie, en développant les outils dont ils ont besoin pour leur propre recherche. Cela doit se faire sous la supervision amicale d'un permanent dont le rôle est de garantir une bonne conception et l'intégration à long terme.

Programmation extrême.

En passant outre la dénomination pompeuse, les démarches de la programmation extrême s'appliquent fort bien à notre type de développement : refactorisation permanente, revues de code, programmation guidée par les tests, intégration constante. Le développement doit en effet s'adapter aux besoins très variables des projets de recherche dont les progrès sont par nature imprévisibles. Seule réserve : de par la dispersion géographique des chercheurs, la programmation par paires ne peut avoir lieu que ponctuellement (ateliers, etc.).

Rendre trivial en pratique ce qui est trivial en théorie.

L'expérience montre que la plupart des calculs repose à 90% sur une myriade de petits détails triviaux mathématiquement (calcul de la longueur d'une permutation ; extension par linéarité d'un opérateur, construction du dual d'une algèbre, etc.). Un rôle essentiel de la boîte à outils est de prendre en charge la majorité d'entre eux pour que le programmeur puisse se concentrer sur le cœur de son problème. C'est à ce niveau là que la mutualisation fonctionne le mieux, soutenue par une modélisation de haut niveau au plus près du langage mathématique.

Algorithmique et conception sont les mamelles de l'efficacité.

La perception de l'efficacité dépend fortement des besoins. Comme nous l'avons vu, le plus souvent en exploration informatique nous avons à contrôler une explosion combinatoire. De ce fait, une optimisation n'est intéressante que si elle permet de calculer vraiment plus loin, ce qui nécessite un gain en complexité algorithmique. D'autre part, du point de vue

du chercheur, la mesure perçue du temps inclut non seulement le temps du calcul lui-même, mais aussi le temps d'exprimer la question. Enfin, nous devons traiter une grande variété de problèmes, chaque nouvelle question apportant son lot de spécificités. Notre situation est donc très différente d'une bibliothèque comme `GMP` ou `Linbox` dont le rôle est de traiter un nombre limité de problèmes (une dizaine pour `Linbox` : calcul de rang, de déterminant, etc), mais ce de manière extrêmement optimisée.

Notre priorité est donc à une bonne complexité, à la souplesse et à l'expressivité. De ce fait, c'est l'algorithmique et la conception qui priment.

Notre chance est que, la plupart du temps, notre code a principalement un rôle de pilotage à partir de briques de base dont la granularité est relativement élevée (arithmétique sur des entiers longs, algèbre linéaire, etc.). De ce fait, les surcoûts éventuels liés à l'utilisation d'un langage souple et de haut niveau (appels de méthodes, résolution de surcharge, code interprété, etc.) sont négligeables par rapport aux calculs sous-jacents. Ainsi, il est possible de gagner sur les deux tableaux, à condition que la plateforme permette simultanément de programmer à un haut niveau et d'utiliser des briques de base externes hautement optimisées.

Les opérations combinatoires élémentaires (par exemple le calcul du nombre de descentes d'une permutation) sont une exception notable ; ici la granularité est petite (accès aux éléments d'une liste et manipulations de petits entiers). Cela peut être géré de manière satisfaisante si la plateforme permet de plus de compiler les sections critiques en fournissant éventuellement des annotations supplémentaire (par ex. `Sage` + `Cython`).

3.2. Distribution et outils de développement. Rendre `MuPAD-Combinat` facile à installer et à utiliser par le plus grand nombre était un point essentiel pour développer notre communauté. La distribution de `MuPAD-Combinat` m'a demandé un travail considérable :

- développement d'une architecture de compilation aux standards GNU (`automake`, `autoconf`, etc.) pour les modules dynamiques (C/C++), pour la documentation (`LATEX` puis XML avec adaptation de scripts de l'équipe `MuPAD` en `Java`, `Perl` et `Ruby`) et pour les tests ;
- compilation pour toutes les plateformes : Linux, MacOS X, Windows (via `cygwin` sur une machine virtuelle `QEMU`), voire `Zaurus` ; cela a demandé un certain nombre de correctifs ou d'adaptations sur des logiciels que nous intégrions (`Symmetriza`, `Nauty`, `lrcalc`) ;
- création de paquetages `rpm` et `deb` pour Linux, d'un installateur `NSIS` pour Windows, d'un CDROM vif basé sur `Morphix` ;
- codéveloppement de `MuPACS`, un mode `MuPAD` pour `emacs`, avec dévermineur et aide intégrés ;
- mise en place d'outils collaboratifs sur `sourceforge` (site Web, listes de diffusions, serveur `CVS` puis `subversion`, `Wiki`, etc.).

Ce travail d'ingénierie, très gourmand en temps, était difficile à déléguer faute de compétences. De ce fait, en 2007 comme en 2008, la sortie stable traditionnelle pour `FPSAC` n'a pas eu lieu, malgré un grand nombre de fonctionnalités nouvelles. Fort heureusement, l'intégration dans le projet `Sage` nous permet maintenant de mutualiser ces aspects techniques ; ils sont maintenant entièrement gérés par un ingénieur merveilleusement compétent et motivé (Michael Abshoff, merci mille fois!).

Le seul point restant à gérer est du ressort du modèle de développement : le choix des outils collaboratifs et leur bonne utilisation. L'intégration dans `Sage` a été l'occasion d'une évolution pour pallier quelques dérives. J'illustre ici l'une d'entre elles.

La plupart du temps, l'implantation des outils nécessaires à un calcul donné est transverse, impliquant quelques aspects combinatoires et algébriques qui sont du ressort de `*-Combinat`, et d'autres touchant plus au cœur de la plateforme (correctifs de bogues, algèbre linéaire, polynômes, etc.). L'organisation de `MuPAD-Combinat` comme *bibliothèque* pour `MuPAD` ne permet

pas d'isoler ces modifications : nous avons dû dupliquer un grand nombre de fichiers de la bibliothèque de MuPAD pour y intégrer un mélange de modifications, certaines indubitables, d'autres plus expérimentales. Dans la pratique la gestion de la réintégration ultérieure de ces modifications dans la version officielle de MuPAD est manuelle, fastidieuse, et source d'erreurs. De fait, MuPAD-Combinat et MuPAD étaient progressivement en train de diverger.

Le modèle de développement de Sage résout ce problème en s'organisant autour des patches³ : chacun de ceux-ci isole une modification relativement atomique (ajout d'une fonctionnalité, correctif d'un bogue), mais qui peut concerner plusieurs fichiers. Dans ce cadre, Sage-Combinat devient avant tout une sous-communauté de Sage partageant une collection de patches expérimentaux qui sont développés et utilisés en commun jusqu'à être suffisamment matures pour être intégrés dans Sage. Par exemple, le patch jetant les bases des systèmes de racines a bénéficié des contributions (et donc de l'expertise et du point de vue) de cinq chercheurs sur une durée de deux mois avant d'être intégré dans Sage ; entre-temps, son utilisation par d'autres membres, en particulier dans le cadre d'un autre patch sur les graphes cristallins, avait validé sa conception.

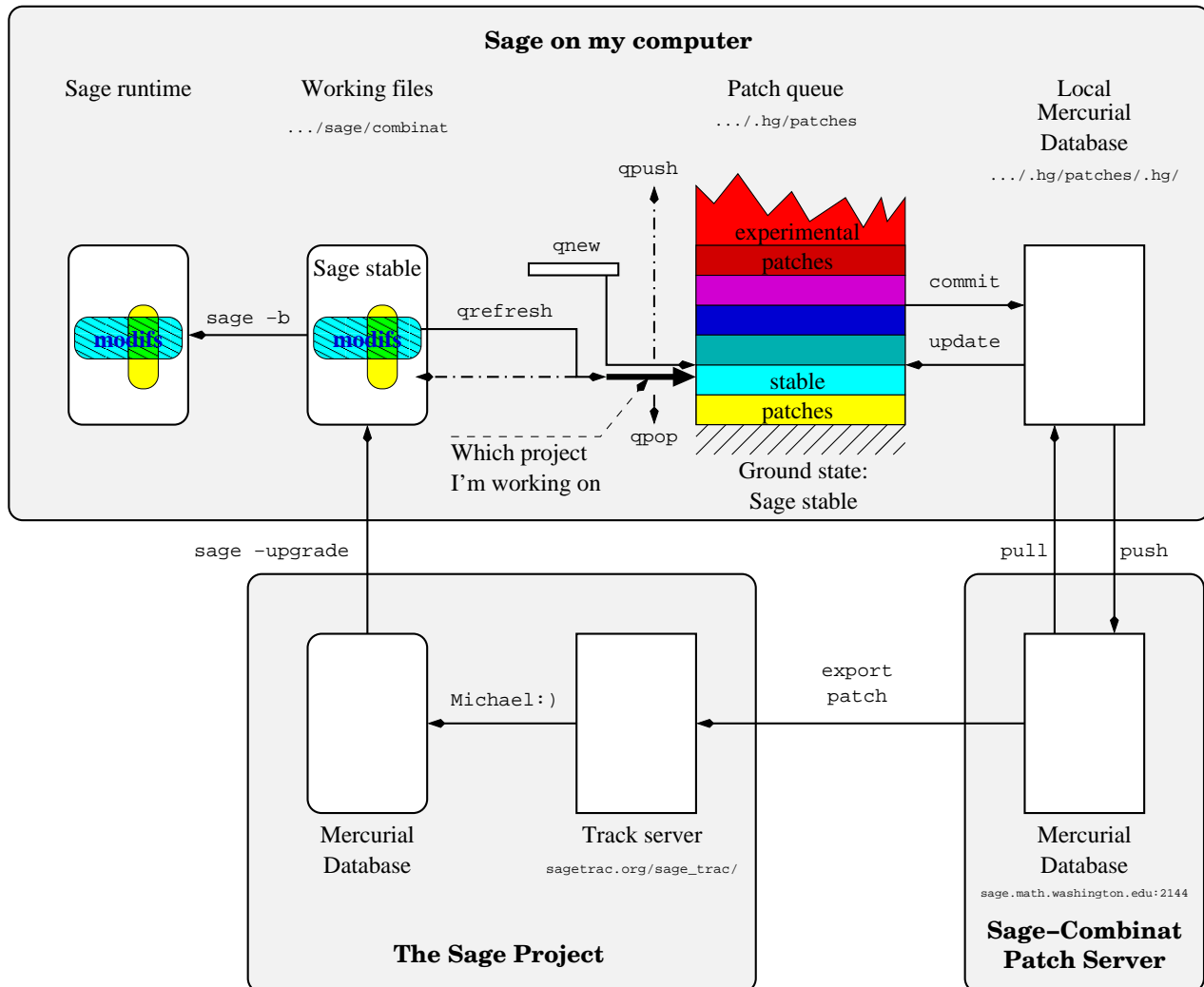


FIGURE 12. Workflow pour le développement de Sage-Combinat *via* un serveur de patches ; voir <http://wiki.sagemath.org/combinat/Mercurial> pour les explications

Sage-Combinat est la première sous-communauté de **Sage** ayant besoin de s'organiser de la sorte. Le choix avec Jason Bandlow, Mike Hansen et Anne Schilling du bon outil collaboratif (les piles⁴ de patchs du système de gestion de version décentralisé **Mercurial**) et d'un bon workflow⁵ (voir figure 12) a nécessité plusieurs semaines d'expérimentation et surtout la grande patience de nombreux cobayes, afin de concilier des contraintes inédites :

- Encourager une intégration rapide dans **Sage** pour s'adapter à son rythme de développement rapide (une nouvelle version par mois).
- S'adapter à la dispersion géographique et thématique des chercheurs et à des rythmes de travail très irréguliers ; un patch peut rester dormant quelques mois (enseignement, habilitation) ; d'autres peuvent être développés en trois jours par quatre personnes à l'occasion d'une conférence.
- Encourager la collaboration autour de chaque patch.
- Contrôler la complexité et garantir la robustesse, pour que des utilisateurs ou contributeurs occasionnels puissent en faire une utilisation simple.

4. Conception : modélisation objet

4.1. Importance de la programmation orientée objet.

« La programmation orientée objet (POO) ou programmation par objet, est un paradigme de programmation informatique qui consiste en la définition et l'assemblage de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. »

[**Wik**, Orienté objet]

Je rajoute, dans notre contexte, tout objet mathématique sur lequel on souhaite faire un calcul : un nombre, un vecteur, un arbre, une partition ; mais aussi, j'y reviendrai en section 4.3, un groupe, une classe combinatoire, une algèbre.

Pour un informaticien, l'utilisation de la programmation orientée objet (apparue dès les années 1970!) semble aller de soi pour un projet logiciel de l'ampleur d'un système de calcul formel. Certains de ces systèmes, comme **Axiom**, l'ont d'ailleurs intégrée très tôt. Singulièrement, son utilité reste encore contestée, certains mettant en valeur que des systèmes comme **Maple** (ou **Mathematica**), au succès d'ailleurs incontestable jusqu'ici, en particulier en combinatoire, semblent fonctionner très bien sans. À cela, deux facteurs me semble-t-il. Le premier est qu'ils ont été conçus avec très peu de structures de données : nombres (entiers, flottants, etc.), listes, tables, matrices, et surtout l'universelle expression. Ces structures sont implantées dans le noyau, sans possibilité d'extension. À quelques inconvénients près (opérateur $&*$ pour la multiplication de matrices), cela est effectivement suffisant pour l'essentiel du calcul formel universitaire de licence : algèbre linéaire, calculus, etc. Au prix de contorsions, beaucoup de problèmes peuvent s'y ramener. On peut aussi songer à la gamme d'applications de **Matlab**, autour d'une seule structure de données (en caricaturant à peine) : la matrice à coefficients flottants. L'autre facteur est que l'essentiel du code mathématique de **Maple**, incluant les multiples contributions externes est composé de petites bibliothèques indépendantes.

Mais la programmation orientée objet devient fondamentale à notre échelle :

- **MuPAD-Combinat** contient plus de 600 briques logicielles (600 domaines, ou classes concrètes, pour 5000 méthodes et 130k lignes de code), que l'on peut composer entre elles. Cela permet, par exemple, de construire une algèbre dont la base est indexée par des arbres et les coefficients sont dans une extension algébrique $\mathbb{Q}[\epsilon]$ avec ϵ une racine de l'unité

4. littéralement des files (*patch queue*), mais dans la pratique, nous les utilisons plutôt comme piles

5. Quel nom affreux ! Flot de développements ?

(voir la démonstration en section 2). En `Maple`, calculer avec des matrices à coefficients dans $\mathbb{Z}/2\mathbb{Z}$ est déjà malaisé.

- `MuPAD-Combinat` étend considérablement la hiérarchie de catégories (classes abstraites) de `MuPAD` (voir figure 13). Cela a permis de factoriser une grande quantité de code, en le rendant générique, alors que, par exemple, 90% du code de `ACE` était dupliqué entre les bibliothèques de fonctions symétriques commutatives et non commutatives.

Les mathématiques sont au cœur de la discussion. D'une part, la hiérarchie de catégories *modélise* naturellement la hiérarchie usuelle des catégories mathématiques (d'où le nom). D'autre part le choix de la catégorie où est implantée une fonctionnalité donnée est conditionné par le niveau d'abstraction auquel l'algorithme utilisé s'applique. Par exemple, l'algorithme de calcul du radical, que nous avons utilisé à l'origine pour des algèbres de type Hecke, a été implanté au niveau des algèbres de dimension finie. Plus tard, il a été remonté au niveau des algèbres non nécessairement associatives (un simple déplacement du code), ce qui permet de l'appliquer aux algèbres de Lie.

4.2. Améliorations de l'héritage multiple en MuPAD. La modélisation de la hiérarchie usuelle des catégories mathématiques requiert un mécanisme d'héritage multiple. Par exemple, une algèbre est à la fois un anneau et un module sur le corps de base. En `MuPAD`, ce mécanisme se situe naturellement au niveau des catégories (classes abstraites). L'ordre dans lequel on recherche une méthode dans la hiérarchie est notoirement sensible. D'un point de vue mathématique, on souhaite que cet ordre soit au moins une extension linéaire de la hiérarchie; on veut en effet garantir qu'une implantation spécialisée d'une méthode dans une sous-catégorie A surcharge systématiquement celle générique définie dans une sur-catégorie B. La construction de notre hiérarchie de catégories a révélé que ce n'était pas le cas dans `MuPAD`. L'ordre de recherche suivait en effet un parcours en largeur (comme jusqu'à très récemment en `Python`), mais leur hiérarchie originale était suffisamment simple et proche d'être graduée pour que cela n'ait jamais posé de problème.

C'est un exemple typique de modification, plutôt simple mais profonde et donc sensible, que j'ai apportée à `MuPAD`, avec la bénédiction de ses développeurs. De même, j'ai rajouté un système d'initialisation des domaines, extensible par greffons au niveau de chaque catégorie. Il permet typiquement de faire des déclarations de conversions implicites ou de surcharge pour les opérateurs adéquats de la catégorie. J'ai aussi introduit une notation objet plus usuelle, objet : `:methode()`.

4.3. Modélisation des domaines. Dans cette section, j'explique pourquoi nous avons atteint les limites du modèle objet de type Domaine/Catégorie de `Axiom` et `MuPAD`.

Nous avons vu dans la démonstration (section 2 et en particulier 2.2) que nous souhaitions mener simultanément des calculs sur les éléments d'un domaine et sur les domaines eux-mêmes (un *domaine* étant un ensemble muni d'opérations, comme par exemple l'anneau des matrices $M_n(\mathbb{R})$, l'anneau de polynômes $\mathbb{C}[x, y, z]$, un groupe ou un graphe cristallin).

Traditionnellement, ce sont les éléments des domaines que l'on manipule, mettons les matrices dans $M_n(\mathbb{R})$. Ces éléments sont donc modélisés par des objets. Le domaine, $M_n(\mathbb{C})$ lui n'a pas d'existence. À la rigueur, on peut considérer que la classe des matrices le modélise, mais typiquement dans un langage typé statiquement, cette classe peut ne pas avoir d'existence à l'exécution.

Dans le modèle objet d'`Axiom`, repris par la suite par `MuPAD`, les domaines prennent une existence à part entière. Ils sont modélisés par des classes. Cela requiert plusieurs points techniques qui excluent l'implantation de ce modèle dans la plupart des langages de programmation classiques. Tout d'abord, les classes sont des entités de premier niveau dans le langage. On peut les stocker dans des variables, les passer en paramètre à des fonctions, etc. D'autre part,

ces classes sont paramétrées : pour modéliser $M_{10}(\mathbb{R})$, on construit, à l'aide d'un foncteur générique pour les matrices carrées, la classe des matrices $10 \otimes 10$ sur le corps des réels (glissons sur le problème de représentation des réels par des flottants). On peut de plus créer au vol de nouvelles classes, éventuellement en grand nombre. Enfin, les classes sont réflexives : on peut, à l'exécution, demander leurs propriétés (ainsi, l'algorithmique pour les matrices pourra changer selon si l'anneau de base est un corps, un anneau principal, etc. À cet effet, le modèle introduit un deuxième niveau de classes (les catégories), jouant en gros le rôle de classes abstraites. Point fort du système, ces catégories associent aux domaines des informations mathématiques.

Ce modèle fonctionne très bien en calcul formel traditionnel. Les domaines servent à décrire la représentation concrète des éléments (structure de données et opérations de bas niveau), tandis que les catégories décrivent les propriétés mathématiques et les algorithmes génériques associés. Mais nous en avons atteint les limites.

Prenons un exemple : les ordres partiels. Il y a d'un côté les petits ordres partiels, que l'on veut décrire en stockant explicitement le graphe sous-jacent. On les modélise donc naturellement par des objets dans une classe « PetitOrdrePartiel ». Cette classe implante une opération « prédécesseurs immédiats(O,x) » qui, étant donné un ordre partiel O (une instance de PetitOrdrePartiel) et un de ses sommets x renvoie les successeurs immédiats de x dans O . La classe PetitOrdrePartiel hérite d'une classe OrdrePartiel qui implante des opérations génériques à partir des opérations de base ; par exemple « section initiale(O,x) » renvoie tous les éléments plus petits que x dans O . Maintenant se présente une autre situation : l'ensemble P des partitions est lui aussi muni d'une structure d'ordre partiel (le treillis de Young). Mais cette fois, dans notre modèle, P est une classe ; elle ne peut donc pas être elle-même une instance de la classe OrdrePartiel. On ne peut donc pas réutiliser les opérations génériques qui y sont implantées.

Autre exemple : la résolution de surcharge, par exemple dans une expression $a*b$ se fait généralement en considérant les domaines, donc les classes, de a et b . Comment doit-on procéder pour une opération $A \otimes B$ qui prend deux domaines ? Quelle est la classe d'une classe ?

Quel est le cœur du problème ? Dans **Axiom** ou **MuPAD**, on modélise la relation « x est un élément du domaine X » par « x est une instance de la classe x ». Mais alors les domaines et les éléments ne jouent plus des rôles semblables. Introduire une double hiérarchie n'est qu'un palliatif. Comment modéliser, ce qui est naturel et utile mathématiquement, des ordres partiels dont les éléments sont eux même des ordres partiels ? Ou des monoïdes de monoïdes de monoïdes ? Comment implanter du code générique fonctionnant quel que soit le niveau du domaine considéré ?

Dans **Sage** (tout comme dans **Magma** dont il est inspiré), le modèle choisi est différent et, je pense, plus adapté à notre besoin. Au fond, on en revient à la base de la programmation objet : tout concept mathématique que l'on souhaite manipuler doit être représenté par un objet ; et cela vaut aussi pour un groupe, un ordre partiel, ou $M_n(\mathbb{C})$. La relation « x est un élément du domaine X » est une relation entre deux objets.

Cela dit, la conception d'une hiérarchie de catégories systématique n'en est qu'à ses débuts dans **Sage**, et il reste de nombreuses questions pratiques à régler. Pour commencer, le fait qu'un domaine soit dans une certaine catégorie (mettons une algèbre de dimension finie) donne des algorithmes génériques à la fois pour les éléments du domaine (test d'inversibilité) et pour le domaine lui-même (calcul de la théorie des représentations). Peut-on éviter l'établissement d'une double hiérarchie de classes en parallèle, l'une donnant les opérations sur les éléments et l'autre sur les domaines ? Comment gérer, lors de la construction d'un produit cartésien $C = A \times B$ toutes les propriétés et opérations de C induites par celles de A et B (des groupes, des classes combinatoires, etc.) ?

Cette réflexion doit bien entendu se faire en lien avec des experts. Nous renvoyons par exemple à [HR04] pour des réflexions complémentaires sur la modélisation objet pour le

calcul formel, en particulier dans le cas plus contraignant d'un langage fonctionnel à typage statique.

5. Conception : représentations multiples

5.1. Représentations multiples. Une stratégie classique du calcul formel est d'avoir plusieurs représentations pour les objets, et de convertir de l'une à l'autre selon le calcul à réaliser. Typiquement, on peut représenter un polynôme univarié comme combinaison linéaire de monômes ou par évaluation sur des points bien choisis (transformée de Fourier), la seconde représentation donnant un calcul de produit en temps linéaire.

En combinatoire algébrique, cette idée est démultipliée; il y a par exemple plus d'une vingtaine de bases implantées pour les fonctions symétriques dans **-Combinat* (voir figure 14), chacune ayant son utilité propre. On veut typiquement faire des calculs comme :

```
>> S := examples::SymmetricFunctions();
>> x := S::p( S::m[1] * ( S::e[3]*S::s[2] + 1 ) ) # S::e[2,1]
      p[1] # e[2, 1] + 1/6 p[3, 2, 1] # e[2, 1] + 1/6 p[3, 1, 1, 1] # e[2, 1] -
      1/4 p[2, 2, 1, 1] # e[2, 1] - 1/6 p[2, 1, 1, 1, 1] # e[2, 1] + 1/12 p[1, 1, 1, 1, 1, 1] # e[2, 1]
```

où # est le symbole utilisé pour les produits tensoriels et, par exemple, $S : : s$ modélise l'algèbre des fonctions symétriques exprimées sur la base des fonctions de Schur.

Il est alors hors de question d'implanter toutes les conversions possibles. De même, il n'est pas envisageable d'implanter toutes les opérations possibles (produit, coproduit, antipode, omega, etc) dans toutes les bases; souvent on ne connaît pas (encore) de meilleur algorithme que de changer de base et d'y faire le calcul! Cela devient vital lorsque l'on souhaite explorer informatiquement une nouvelle algèbre de Hopf ou une nouvelle base.

Il est aussi néfaste de devoir préciser, pour chaque calcul, les conversions à utiliser. Cela requiert de la part de l'utilisateur (novice ou expérimenté) une connaissance approfondie de ce qui est implanté. Et surtout, cela fige le code dans un état donné, ne lui permettant pas de bénéficier automatiquement d'implantations ultérieures de conversions ou d'opérations.

Cette réflexion, guidée par la pratique, nous a amenés Florent Hivert et moi à dégager l'idée clef suivante :

L'implantation d'une algèbre (de Hopf) doit être appréhendée comme une base de données évolutive de représentations (différentes bases) et de formules, règles de calculs ou autres algorithmes (Jacobi-Trudi, Pieri Littlewood Richardson, etc.). À charge pour le système de choisir, pour chaque calcul, les changements de représentations et les algorithmes appropriés.

En somme, l'esprit est de modéliser l'analogie du « *formulaire raisonné des fonctions symétriques* » [LS85], tout en encapsulant l'ensemble des opérations et conversions effectivement codées, pour en faire un détail d'implantation.

La réalisation de cette idée passe par trois points : un mécanisme de conversions implicites, un mécanisme de surcharge multiparamètres, et une infrastructure basée sur ces mécanismes pour construire aisément des domaines avec plusieurs représentations. La difficulté était la mise au point de modèles souples, robustes et généralistes, faisant des choix raisonnables, prédictibles et avant tout corrects mathématiquement, le tout avec de bonnes performances. La simplicité était de mise. Par exemple, nous avons choisi de ne pas essayer d'inclure dans la base de données des informations de complexité d'algorithmes pour guider le choix du système. Il est préférable de laisser *in fine*, le contrôle à l'utilisateur avancé, lui permettant, lorsque cela s'avère nécessaire, de rajouter quelques règles supplémentaires (par ex : pour calculer le produit d'une fonction de Schur par une fonction puissance, passer par telle base et utiliser tel algorithme), plutôt que de l'inciter à ajuster des paramètres de complexité pour corriger des travers éventuels. Cela laisse typiquement la porte ouverte à l'utilisation de stratégies différentes selon la taille des données, avec ajustement d'un paramètre de seuil.

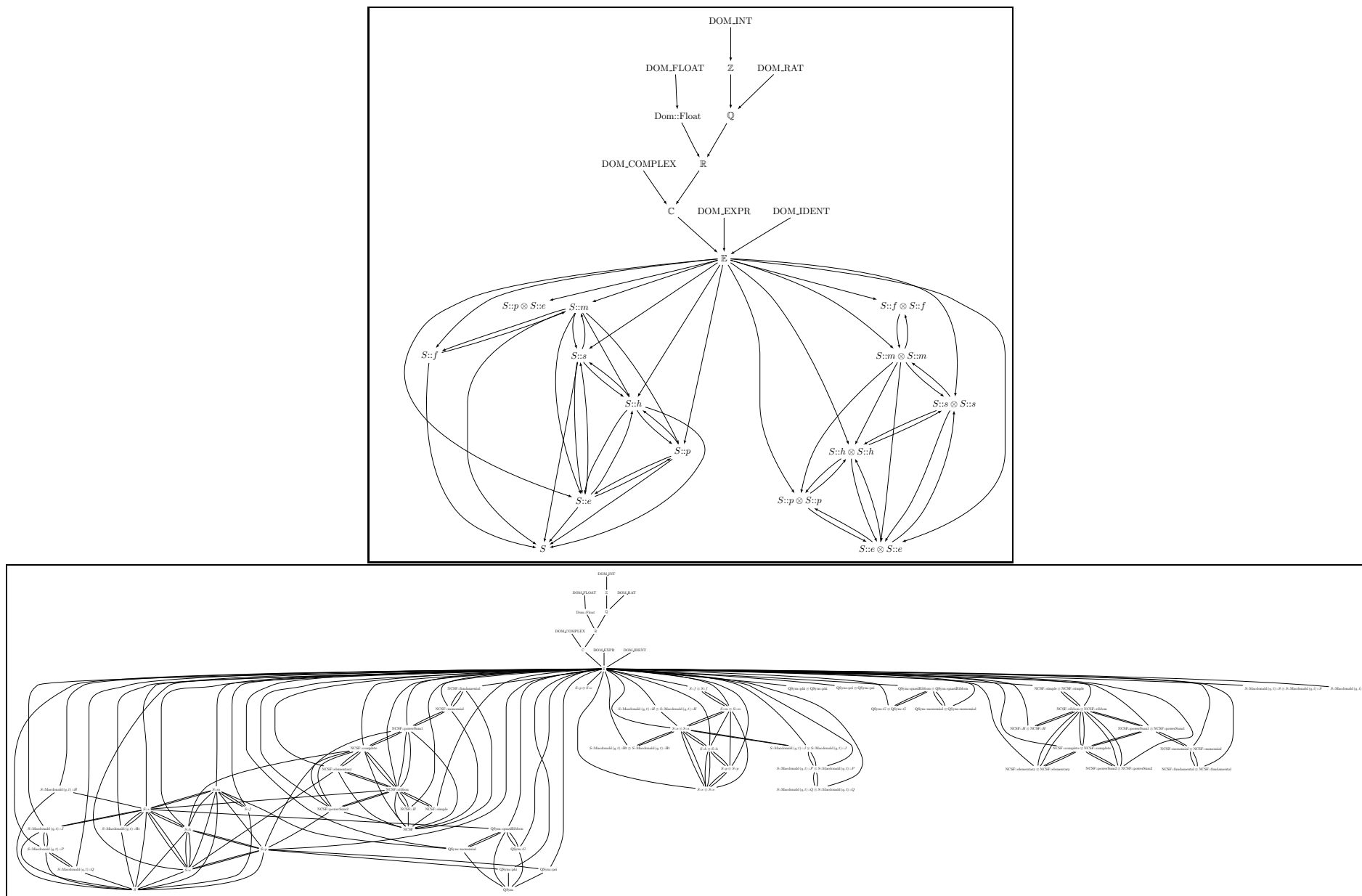


FIGURE 14. Graphes des conversions dans deux sessions MuPAD-Combinat typiques; en haut : avec juste les fonctions symétriques usuelles; en bas : avec en sus les polynômes de Maconald, les fonctions symétriques non commutatives et les fonctions quasismétriques

Je me suis chargé des deux premiers points, que je décris ci-dessous, et j'ai secondé Florent Hivert sur le troisième. Nous aurions bien préféré pouvoir transposer directement un des modèles existant dans des langages comme C++, Java, Python, Lisp, GAP, Axiom; cependant l'ampleur de nos besoins spécifiques nous a amené à des solutions originales, quoique largement inspirées de ces modèles.

5.2. Résolution de surcharge multiparamètres et conversions. Tous les langages orientés objet, y compris MuPAD, implantent la surcharge simple des méthodes : lors d'un appel $f(x, y)$, le choix de la fonction appelée sera déterminé par la classe de x . Usuellement cela est mis en valeur par la syntaxe, l'appel s'écrivant plutôt sous une forme comme $x.f(y)$.

Ce système a ses limites. Prenons un exemple dans le contexte du calcul formel (voir [Sin05] pour des exemples dans d'autres contextes). Supposons que x soit un nombre rationnel, et y un vecteur de \mathbb{Q}^3 . Quelle méthode est effectivement appelée par $x * y$? Ce devrait être celle de multiplication des vecteurs de \mathbb{Q}^3 par les scalaires dans \mathbb{Q} , dont l'implantation est rattachée à \mathbb{Q}^3 . Cependant, dans MuPAD, mais aussi dans Sage jusqu'à très récemment, c'était la méthode `mult` des rationnels qui était appelée, à charge pour celle-ci de renvoyer l'appel au bon endroit (l'exemple est simplifié pour l'exposé, en fait il y avait un traitement spécifique pour les rationnels, mais le principe reste). Cela est contraire au principe de localisation : la méthode de multiplication des rationnels ne devrait avoir à traiter que de la multiplication des rationnels (ou à la rigueur des rationnels par des entiers), sans avoir à gérer et donc dans une certaine mesure à connaître, tout ce qui peut et pourra être construit au dessus des rationnels. En particulier, il est difficile de s'assurer que tous les objets planteront le renvoi de manière consistante (ce n'était pas le cas en MuPAD).

Ce n'est certainement pas un problème original. Tous les langages de programmation implantent au moins un système minimal de surcharge permettant d'écrire $x * y$ pour x et y entiers, flottants, etc. Le plus souvent, cela se fait avec un traitement au cas par cas. Ce qui est original dans notre cas, c'est l'ampleur du problème. Il n'est pas inusuel dans nos calculs de créer temporairement plusieurs milliers de domaines, modélisant autant de structures algébriques dans lesquelles ont lieu les calculs. J'avais réalisé un premier prototype, inspiré par celui de C++ (quoique adapté à un langage où le typage est dynamique) ou de GAP. Ce prototype a été essentiel pour le développement de l'infrastructure pour les algèbres (de Hopf) et en particulier les outils de théorie des représentations (Florent Hivert considère par exemple que, sans ce prototype, la publication [HNT06] n'aurait simplement pas existé). Il a permis de simplifier considérablement et donc de rendre robuste et cohérentes les conversions et les surcharges dans MuPAD.

Les spécifications suivantes ont été conditionnées par le retour d'expérience lors d'utilisations intensives. En particulier, les considérations de complexité en temps et en mémoire proviennent toutes de bogues rencontrés lors de gros calculs (jusqu'à 20% du temps perdu dans la résolution des surcharges) ou lors de l'intégration dans MuPAD (fuites de mémoire dans les tests de factorisation, ces derniers nécessitant la construction de multiples tours d'extensions algébriques).

- Les résolutions de surcharge ou de conversions sont déterminées par le domaine du ou des arguments.
- Le système maintient un graphe des conversions implicites entre domaines existants. Chacune de ces conversions correspond à l'implantation d'un morphisme naturel canonique (le plongement du corps de base dans une algèbre, les isomorphismes entre fonctions symétriques dans différentes bases, etc.).

L'aspect canonique garantit que, s'il existe deux chemins de conversions implicites entre deux domaines, alors les conversions doivent donner le même résultat (diagramme commutatif).

L'aspect naturel garantit que la conversion est un morphisme pour toutes les opérations surchargées. Ce dernier point est flou : quelle catégorie mathématique ce graphe modélise-t-il ?

Deux domaines dans la même composante fortement connexe sont considérés isomorphes ; c'est-à-dire qu'ils implantent deux représentations pour la même structure mathématique. Un résultat peut alors être exprimé dans l'une quelconque des représentations.

Noter que ce graphe (voir figure 14) a une structure très spécifique : l'ensemble des prédécesseurs d'un domaine est relativement petit (<20), tandis que l'ensemble des successeurs peut représenter presque tout le graphe (pour \mathbb{N}).

- La résolution de surcharge, lorsqu'il n'y a pas coïncidence exacte des signatures, recherche la solution la plus approchée en autorisant l'utilisation de conversions implicites. Il n'est pas souhaitable de remonter dans la hiérarchie de classe, comme dans le modèle des multiméthodes : deux \mathbb{Q} algèbres A et B de dimension 4 hériteront typiquement de la même structure sous-jacente d'espace vectoriel \mathbb{Q}^4 ; mais cela ne donne pas pour autant un sens à l'addition d'un élément de A et d'un de B .

Lorsqu'il y a plusieurs solutions possibles, les hypothèses garantissent que les résultats sont identiques (éventuellement *via* un isomorphisme). Le rôle du choix est donc uniquement de minimiser le nombre de conversions pour gagner en efficacité. Une erreur dans ce choix n'aura donc pas de conséquence sémantique.

- Dans une résolution approchée, tous les arguments peuvent simultanément subir une conversion. Mais au moins l'une de ces conversions doit être entre deux domaines isomorphes.
- Le mécanisme doit supporter la création au vol, éventuellement temporaire, de plusieurs milliers de domaines. En particulier :
 - La résolution de conversions ou de surcharge ne doit avoir lieu que si elle est nécessaire (évaluation paresseuse). La première résolution ne doit considérer que la portion du graphe strictement nécessaire (typiquement de l'ordre de la dizaine de domaines). Les résolutions suivantes doivent être en temps constant (petit), *via* l'utilisation d'un cache.
 - Le cache mémoire ne doit pas dépasser un espace essentiellement linéaire vis-à-vis du nombre de domaines et d'opérations effectivement utilisés.
 - La représentation en mémoire du graphe et du cache ne doivent pas empêcher la libération de la mémoire occupée par un domaine lorsque celui-ci n'est plus utilisé (que ce soit par comptage de référence ou par glanage de cellules lorsqu'il fait partie d'une composante connexe non triviale).
 - Il n'est pas essentiel que le rajout (ou la suppression) a posteriori d'une conversion invalide le cache pour les conversions et surcharges indirectes déjà résolues.

L'implantation actuelle dans MuPAD vérifie ces spécifications grâce à une structure de données répartie, pour le graphe de conversion comme pour le cache, et une algorithmique appropriée. Dans le même temps le modèle suit suffisamment les mathématiques pour que son utilisation reste intuitive, voire transparente.

6. Étude de cas : objets décomposables et espèces combinatoires

Les classes combinatoires décomposables (ou espèces combinatoires) sont un des moteurs de calcul essentiels pour une bibliothèque de combinatoire (algébrique). Elles permettent de traiter génériquement toutes les classes combinatoires que l'on peut définir en composant des classes plus simples, typiquement récursivement. Comme nous l'avons vu dans la section 2 cela inclut bien entendu les arbres sous toutes leur formes (plus d'une vingtaine dans MuPAD-Combinat),

mais aussi les partitions, les langages définis par automate ou grammaire, etc. Pour mentionner une application plus originale, cet outil a fait de `MuPAD-Combinat` un maillon important d'une chaîne logicielle pour faire du test statistique de programmes [**Gou04**, **DGG04**, **DGG+06a**, **DGG+06b**, **Oud07**, **DGG+08**].

La première implantation, par Paul Zimmermann, remonte à Gaia [**Zim94**, **FZVC94**], devenue par la suite la bibliothèque `combstruct` de `Maple`. Cette bibliothèque permet le comptage et le tirage aléatoire pour les classes décomposables non étiquetées. Par la suite, Paul Zimmermann, Alain Denise et Isabelle Dutour ont porté cette bibliothèque vers `MuPAD 1.4.2` sous le nom de `CS` [**DDZ98**], et surtout l'ont étendue : tirage aléatoire de structures de grande taille grâce au comptage approché en flottant, calcul automatique de relations de récurrence, utilisation du produit de Karatsuba sur les séries génératrices, génération de code C autonome. Sébastien Cellier s'est chargé sous ma direction de son adaptation à `MuPAD 2.0.0`.

Ma première contribution a été d'utiliser l'architecture objet de `MuPAD-Combinat` pour réduire considérablement la taille du code et surtout le rendre réentrant : il est maintenant possible de construire plusieurs classes décomposables simultanément. En utilisant les fermetures et des générateurs, j'ai rajouté les fonctionnalités de listage et d'itération (qui, pour nous, sont les plus importantes). Enfin, j'ai intégré cette bibliothèque dans `MuPAD-Combinat`, lui permettant d'une part de prendre une classe combinatoire quelconque de `MuPAD-Combinat` comme brique de base, et d'autre part d'être utilisé comme moteur de calcul interne. M'inspirant des travaux de Xavier Molinero et Conrado Martínez [**MM03**, **MM05**], j'ai commencé à étendre la bibliothèque pour traiter les objets étiquetés. Florent Hivert s'est chargé de l'implantation des constructeurs `Cycle` et `Set` en étiqueté ; suite à un *quiproquo*, il a en fait introduit pour cela une technique nouvelle (`Div/MultX`) qui permet en fait de traiter des exemples nouveaux par rapport au « boxed product » de Xavier Molinero et Conrado Martínez [**MM03**]. Lors de deux séjours croisés, Xavier Molinero a rajouté les opérations de *unrank*, tandis que j'implantais l'itération pour les constructeurs de cycles, de mots de Lyndon et d'ensembles pour les objets non étiquetés [**MMT06**], en m'appuyant sur un algorithme de Joe Sawada [**MM04**, **Saw03**]. À l'automne 2005, j'étais dans le jury de thèse de Xavier Molinero.

Il est clairement apparu à ce moment là que la conception monolithique de la bibliothèque atteignait ses limites, rendant fastidieuse sa maintenance et son extension. Aussi, lorsque invité à l'atelier `Axiom 2006` j'ai décrit les grandes étapes pour établir une bibliothèque de combinatoire, j'ai insisté sur l'importance des objets décomposables et de leur implantation de façon très modulaire. Pour illustrer mon propos, j'avais écrit un petit prototype en `Aldor`, proche d'un autre prototype en `C++` que j'avais écrit quelques mois auparavant après une discussion avec Conrado Martínez.

Martin Rubey et Ralf Hemmecke ont tout de suite pris le projet en main, marquant la naissance d'`aldor-Combinat` [**HR06**]. Martin Rubey a introduit un leitmotiv qui s'est avéré brillant : suivre au plus près la théorie des espèces [**Joy81**, **BLL94**] pour unifier le cas étiqueté et non étiqueté. Cela l'a amené à traiter aussi les cas intermédiaires (étiquetage semi-standard à contenu fixé) afin de pouvoir générer les structures non-étiquetées pour la substitution.

Ralf Hemmecke a, de son côté, apporté une solution logicielle originale : modéliser la définition récursive de la classe combinatoire par une définition récursive des classes par foncteurs paramétrés. Cela permet de construire simultanément la structure de données récursive des objets, et l'algorithmique récursive pour les opérations combinatoires sur la classe. Je ne connais pas d'autre langage de programmation permettant cela ; en fait, lorsque nous en avons discuté, il n'était même pas clair pour nous que le compilateur d'`Aldor` le permette. Ce deuxième aspect, quoique fort esthétique, est à mon sens loin d'être essentiel, et ne doit pas être vu comme un frein pour une adaptation à d'autres langages.

Ensemble, ils ont implanté une bibliothèque extrêmement bien documentée [**HR06**], que l'on peut voir comme une version effective du livre [**BLL94**].

Cela a servi de base inestimable de travail pour Mike Hansen qui, sur ma suggestion, s'est chargé d'une bonne partie de l'adaptation à **Sage** lors de l'été 2008, sur un financement de **Google**⁶. Il est encore trop tôt pour juger de son apport personnel pour l'algorithmique et la conception. Mais son travail est prometteur, en particulier du point de vue de l'intégration avec la génération incrémentale d'objets combinatoires à un isomorphisme près — telle qu'utilisée par Brendan McKay pour les graphes dans **Nauty** — implantée dans **Sage** par Robert Miller.

En résumé, l'implantation actuelle dans **Sage-Combinat** des objets décomposables (ou plus généralement des espèces combinatoires) est le fruit de l'intervention de très nombreux contributeurs, sur le long terme, chacun apportant sa pierre à l'édifice. C'est une illustration exemplaire de l'importance pour ce type d'outil d'être développés sur un modèle libre. Dans ce cadre, et grâce à la pollinisation croisée, même le développement sur plusieurs plateformes en parallèle a été bénéfique, chaque portage ayant été utilisé à bon escient pour essayer de nouvelles techniques logicielles.

6. Voir <http://blog.phasing.org/>.

7. Liste de publications utilisant ou concernant MuPAD-Combinat

- [1] Jason Bandlow, Anne Schilling, and Nicolas M. Thiéry. On the uniqueness of promotion operators on tensor products of type a crystals. Submitted, arXiv :0806.3131 [math.CO], June 2008.
- [2] N. Bergeron, F. Descouens, and M. Zabrocki. A filtration of (q,t) -catalan numbers, 2008.
- [3] N. Bergeron, F. Descouens, and M. Zabrocki. A non-commutative generalization of k -schur functions, 2008.
- [4] N. Bergeron, F. Hivert, and J.-Y. Thibon. The peak algebra and the Hecke-Clifford algebras at $q = 0$. *Journal of Combinatorial Theory Series A*, 107(1) :1–19, 2004.
- [5] Marie-Claude David and Nicolas M. Thiéry. Exploration of finite dimensional kac algebras and lattices of irreducible intermediate subfactors. Submitted, 83 pages, arXiv :0812.3044 [math.QA], December 2008.
- [6] Lipika Deka. *Fermionic Formulas For Unrestricted Kostka Polynomials And Superconformal Characters*. PhD thesis, UC-Davis, 2005. arXiv :math/0512536v2 [math.CO].
- [7] Lipika Deka and Anne Schilling. New fermionic formula for unrestricted Kostka polynomials. *J. Combin. Theory Ser. A*, 113(7) :1435–1461, 2006.
- [8] A. Denise, M.-C. Gaudel, S.-D. Gouraud, R. Lassaigne, and S. Peyronnet. Uniform random sampling of traces in very large models. In *1st ACM International Workshop on Random Testing (RT)*, pages 10–19, 2006.
- [9] Alain Denise, Marie-Claude Gaudel, and S.-D. Gouraud. A generic method for statistical testing. In *15th IEEE International Symposium on Software Reliability In Engineering*, November 2004.
- [10] Alain Denise, Marie-Claude Gaudel, Sandrine-Dominique Gouraud, Richard Lassaigne, Johan Oudinet, and Sylvain Peyronnet. Coverage-biased random exploration of large models and application to testing. Technical Report 1494, LRI, Université Paris-Sud XI, June 2008.
- [11] Alain Denise, Marie-Claude Gaudel, Sandrine-Dominique Gouraud, Richard Lassaigne, and Sylvain Peyronnet. Uniform random sampling of traces in very large models. *CoRR*, abs/cs/0606086, 2006.
- [12] F. Descouens. Un algorithme de génération des tableaux de rubans et de calcul de polynôme de spin. In *10èmes Journées Montoises*, Liège, 2004.
- [13] F. Descouens. Ribbon tableaux, rigged configurations and Hall-Littlewood functions at roots of unity. In *Poster session of 17th Formal Power Series and Algebraic Combinatorics (FPSAC'05)*, pages 579–586, Taormina, Italia, 2005.
- [14] F. Descouens. Making research on symmetric functions using MuPAD-Combinat. In Andrés Iglesias and Nobuki Takayama, editors, *2nd International Congress on Mathematical Software (ICMS'06)*, volume 4151 of *LNCS*, pages 407–418, Castro Urdiales, Spain, September 2006. Springer-Verlag.
- [15] F. Descouens. An algorithm for generating ribbon tableaux and spin polynomials. *Discrete Mathematics and Theoretical Computer Science Proceedings*, 9(2) :51–58, 2007. arXiv :math.CO/0611824.
- [16] F. Descouens. *Combinatoire des tableaux de rubans et des polynômes de Kostka généralisés*. Thèse de doctorat, I.G.M., Université de Marne-la-Vallée, June 2007. Jury : B. Leclerc, C. Lecouvey, J.-C. Novelli, G. Roussel, A. Schilling, J.-Y. Thibon, M. Zabrocki.
- [17] F. Descouens. Ribbon tableaux, ribbon rigged configurations and Hall-Littlewood functions at roots of unity. *Journal of Combinatorial Theory, Series A*, 115(3) :361–375, 2008. arXiv :math.CO/0701221.
- [18] F. Descouens and A. Lascoux. Non symmetric Hall-Littlewood polynomials. *Séminaire Lotharingien de Combinatoire*, 54, 2006. electronic version (14 pp.) slc :B54Ar, arXiv :math.CO/0609512.
- [19] F. Descouens and H. Morita. Macdonald polynomials at roots of unity. In *Poster session of Formal Power Series and Algebraic Combinatorics (FPSAC'06)*, pages 423–434, San Diego, USA, 2006.
- [20] F. Descouens and H. Morita. Factorization formulas for Macdonald polynomials. *European Journal of Combinatorics*, 29(2) :395–410, 2008. arXiv :math/0611522.
- [21] Gerard Duchamp, Hatem Hadj Kacem, and Eric Laugerotte. An unexpected application of minimization theory to module decompositions, 2004.
- [22] Sandrine-Dominique Gouraud. *Utilisation des Structures Combinatoires pour le Test Statistique*. PhD thesis, Université Paris XI, June 2004.
- [23] F. Hivert, J.-C. Novelli, L. Tevlin, and J.-Y. Thibon. Permutation statistics related to a class of noncommutative symmetric functions and generalizations of the Genocchi numbers . *Selecta Mathematica*, 2008. electronic version (13 pp.) Submitted arXiv :0710.0447.

- [24] F. Hivert, J.-C. Novelli, and J.-Y. Thibon. The algebra of binary search trees. *Theoretical Computer Science*, 339(1) :129–165, 2005.
- [25] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as degenerate affine Hecke algebras. *Discrete Math. Theor. Comput. Sci.*, September 2008. In print.
- [26] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as quotients of affine Hecke algebras at level 0. *J. Combin. Theory Ser. A*, November 2009. doi :10.1016/j.jcta.2008.11.010, arXiv :0804.3781 [math.RT].
- [27] Florent Hivert and Nicolas M. Thiéry. Deformation of symmetric functions and the rational Steenrod algebra. In *Invariant theory in all characteristics*, volume 35 of *CRM Proc. Lecture Notes*, pages 91–125. Amer. Math. Soc., Providence, RI, 2004. arXiv :0812.3056v1 [math.CO].
- [28] Florent Hivert and Nicolas M. Thiéry. MuPAD-Combinat, an open-source package for research in algebraic combinatorics. *Sém. Lothar. Combin.*, 51 :Art. B51z, 70 pp. (electronic), 2004. <http://mupad-combinat.sf.net/>.
- [29] Florent Hivert and Nicolas M. Thiéry. Representation theories of some towers of algebras related to the symmetric groups and their hecke algebras. In *Proceedings of FPSAC'06 San Diego*, 2006. arXiv :math/0607391v2 [math.RT].
- [30] Florent Hivert and Nicolas M. Thiéry. The Hecke group algebra of a Coxeter group and its representation theory. *Journal of Algebra*, November 2008. doi :10.1016/j.jalgebra.2008.09.039, arXiv :0711.1561 [math.RT].
- [31] Thomas Lam, Anne Schilling, and Mark Shimozono. Schubert polynomials for the affine grassmannian of the symplectic group. Submitted, arXiv :0710.2720[math.CO], October 2007.
- [32] Conrado Martínez, Xavier Molinero, and Nicolas M. Thiéry. Generació ordenada de classes d'estructures combinatòries (ordered generation of combinatorial structures). In *In Jornada de Recerca EPSEM 2006*, pages 83–84. EPSEM (Technical College of Manresa <<http://www.eupm.upc.es/>>), Remsa S.L. Manresa, April 2006. ISBN : 84-86784-05-0. ISBN : 978-84-86784-05-8.
- [33] X. Molinero. *Ordered Generation of Classes of Combinatorial Structures*. PhD thesis, Universitat Politècnica de Catalunya, November 2005.
- [34] Sergey Mozgovoy. A computational criterion for the Kac conjecture. *J. Algebra*, 318(2) :669–679, 2007.
- [35] J.-C. Novelli and J.-Y. Thibon. Construction of dendriform trialgebras. *Comptes Rendus Mathématique. Académie des Sciences. Paris. Série I*, 342(6) :365–369, 2006.
- [36] J.-C. Novelli and J.-Y. Thibon. Hopf algebras and dendriform structures arising from parking functions. *Fundamenta Mathematicae*, 193 :189–241, 2007.
- [37] J.-C. Novelli and J.-Y. Thibon. A one-parameter family of dendriform identities . *European Journal of Combinatorics*, 2008. electronic version (12 pp.) Submitted arXiv :0709.3235.
- [38] J.-C. Novelli and J.-Y. Thibon. Noncommutative Symmetric Functions and Lagrange Inversion. *Advances in Applied Mathematics*, 40 :8–35, 2008. arXiv :math/0512570v3.
- [39] J-C. Novelli and J.-Y. Thibon. Superization and (q,t) -specialization in combinatorial Hopf algebras. Technical report, March 2008. electronic version (30 pp.) arXiv :0803.1816.
- [40] J-C. Novelli, J.-Y. Thibon, and L. K. Williams. Combinatorial Hopf algebras, noncommutative Hall-Littlewood functions, and permutation tableaux . Technical report, April 2008. electronic version (36 pp.) arXiv :0804.0995.
- [41] Jean-Christophe Novelli and Jean-Yves Thibon. Free quasi-symmetric functions and descent algebras for wreath products, and noncommutative multi-symmetric functions, 2008.
- [42] Janvier Nzeutchap. Dual Graded Graphs and Fomin's r-correspondences associated to the Hopf Algebras of Planar Binary Trees, Quasi-symmetric Functions and Noncommutative Symmetric Functions. In *Poster session of 18th Formal Power Series and Algebraic Combinatorics(FPSAC'06)*, pages 632–644, San Diego, USA, 2006. Best poster award.
- [43] Janvier Nzeutchap. Finding polynomials to count lattice points. Computer explorations with MuPAD-Combinat. In *Colloque Africain sur la Recherche en Informatique (CARI'06)*, pages 59–66, Cotonou, 2006.
- [44] Janvier Nzeutchap. Young-fibonacci insertion, tableauhedron and kostka numbers. *J. Combin. Theory Ser. A*, 2008. in press.

- [45] Johan Oudinet. Uniform random walks in very large models. In *RT '07 : Proceedings of the 2nd international workshop on Random testing*, pages 26–29, Atlanta, GA, USA, November 2007. ACM Press.
- [46] Maurice Pouzet and Nicolas M. Thiéry. Invariants algébriques de graphes et reconstruction. *C. R. Acad. Sci. Paris Sér. I Math.*, 333(9) :821–826, 2001. arXiv :0812.3079v1 [math.CO].
- [47] Anne Schilling. Combinatorial structure of Kirillov-Reshetikhin crystals of type $D_n^{(1)}, B_n^{(1)}, A_{2n-1}^{(2)}$. *J. Algebra*, 319(7) :2938–2962, 2008.
- [48] Nicolas M. Thiéry. Computing minimal generating sets of invariant rings of permutation groups with SAGBI-Gröbner basis. In *Discrete models : combinatorics, computation, and geometry (Paris, 2001)*, Discrete Math. Theor. Comput. Sci. Proc., AA, pages 315–328 (electronic). Maison Inform. Math. Discrèt. (MIMD), Paris, 2001.
- [49] Nicolas M. Thiéry. Algebraic invariants of graphs ; an experimental study. Poster, ISSAC'99, Vancouver, CANADA, 1999. Awarded best poster prize.
- [50] Nicolas M. Thiéry. *Invariants algébriques de graphes et reconstruction ; une étude expérimentale*. PhD thesis, Université Lyon I, June 1999. N° d'ordre : 167-99.
- [51] Nicolas M. Thiéry. Algebraic invariants of graphs : a study based on computer exploration. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 34(3) :9–20, September 2000. arXiv :0812.3082v1 [math.CO].
- [52] Nicolas M. Thiéry. **PerMuVAR**, a library for computing in invariant rings of permutation groups. Software demonstration, MEGA 2000, Bath, UK, 2000.
- [53] J. Vives. *Generació ordenada d'estructures combinatòries en "MuPAD-Combinat (Ordered Generation of Combinatorial Structures in MuPAD-Combinat)*. PhD thesis, Universitat Politècnica de Catalunya, January 2008. Bachelor Thesis.

Bibliographie

- [AB03] J.-C. Aval and N. Bergeron. Catalan paths and quasi-symmetric functions. *Proc. Amer. Math. Soc.*, 131(4) :1053–1062 (electronic), 2003.
- [ABB04] J.-C. Aval, F. Bergeron, and N. Bergeron. Ideals of quasi-symmetric functions and super-covariant polynomials for S_n . *Adv. Math.*, 181(2) :353–367, 2004.
- [ABS06] Marcelo Aguiar, Nantel Bergeron, and Frank Sottile. Combinatorial Hopf algebras and generalized Dehn-Sommerville relations. *Compos. Math.*, 142(1) :1–30, 2006. arXiv :math/0310016v1 [math.CO].
- [ACG96] Helmer Aslaksen, Shih-Ping Chan, and Tor Gulliksen. Invariants of S_4 and the shape of sets of vectors. *Appl. Algebra Engrg. Comm. Comput.*, 7(1) :53–57, 1996.
- [BD08] Jason Bandlow and Michele D’Adderio. A new proof of a theorem of Littlewood. *European Journal of Combinatorics*, 2008.
- [BHT04] Nantel Bergeron, Florent Hivert, and Jean-Yves Thibon. The peak algebra and the Hecke-Clifford algebras at $q = 0$. *J. Combin. Theory Ser. A*, 107(1) :1–19, 2004.
- [BLL94] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Théorie des espèces et combinatoire des structures arborescentes*. Number 19. Publication du LACIM, 1994. 400 pages.
- [Bon91] J. A. Bondy. A graph reconstructor’s manual. In *Surveys in combinatorics, 1991 (Guildford, 1991)*, volume 166 of *London Math. Soc. Lecture Note Ser.*, pages 221–252. Cambridge Univ. Press, Cambridge, 1991.
- [BST08] Jason Bandlow, Anne Schilling, and Nicolas M. Thiéry. On the uniqueness of promotion operators on tensor products of type a crystals. Submitted, arXiv :0806.3131 [math.CO], June 2008.
- [Cam96] Peter J. Cameron. Stories from the age of reconstruction. *Congr. Numer.*, 113 :31–41, 1996. Festschrift for C. St. J. A. Nash-Williams.
- [Cam97] Peter J. Cameron. The algebra of an age. In *Model theory of groups and automorphism groups (Blaubeuren, 1995)*, volume 244 of *London Math. Soc. Lecture Note Ser.*, pages 126–133. Cambridge Univ. Press, Cambridge, 1997.
- [Col97] Antoine Colin. *Théorie des invariants effective ; Applications à la théorie de Galois et à la résolution de systèmes algébriques ; Implantation en AXIOM*. PhD thesis, École polytechnique, 1997.
- [CP95] Vyjayanthi Chari and Andrew Pressley. Quantum affine algebras and their representations. In *Representations of groups (Banff, AB, 1994)*, volume 16 of *CMS Conf. Proc.*, pages 59–78. Amer. Math. Soc., Providence, RI, 1995.
- [DDZ98] Alain Denise, Isabelle Dutour, and Paul Zimmermann. CS : a MuPAD package for counting and randomly generating combinatorial structures. *Proceedings of FPSAC’98*, pages 195–204, 1998.
- [DG04] Vesselin Drensky and Lothar Gerritzen. Nonassociative exponential and logarithm. *J. Algebra*, 272(1) :311–320, 2004.
- [DGG04] Alain Denise, Marie-Claude Gaudel, and S.-D. Gouraud. A generic method for statistical testing. In *15th IEEE International Symposium on Software Reliability In Engineering*, pages 25–34, November 2004.
- [DGG⁺06a] A. Denise, M.-C. Gaudel, S.-D. Gouraud, R. Lassaigne, and S. Peyronnet. Uniform random sampling of traces in very large models. In *1st ACM International Workshop on Random Testing (RT)*, pages 10–19, 2006.
- [DGG⁺06b] Alain Denise, Marie-Claude Gaudel, Sandrine-Dominique Gouraud, Richard Lassaigne, and Sylvain Peyronnet. Uniform random sampling of traces in very large models. *CoRR*, abs/cs/0606086, 2006.
- [DGG⁺08] Alain Denise, Marie-Claude Gaudel, Sandrine-Dominique Gouraud, Richard Lassaigne, Johan Oudinet, and Sylvain Peyronnet. Coverage-biased random exploration of large models and application to testing. Technical Report 1494, LRI, Université Paris-Sud XI, June 2008.
- [DK02] Harm Derksen and Gregor Kemper. *Computational invariant theory*. Invariant Theory and Algebraic Transformation Groups, I. Springer-Verlag, Berlin, 2002. Encyclopaedia of Mathematical Sciences, 130.

- [DSW03] A. Duran, D. Saunders, and Z. Wan. Hybrid Algorithms for Rank of Sparse Matrices. In *Proceedings of the SIAM International Conference on Applied Linear Algebra*, 2003.
- [DSW08] Xavier Dahan, Éric Schost, and Jie Wu. Evaluation properties of invariant polynomials. Submitted, Mar 2008.
- [DT08] Marie-Claude David and Nicolas M. Thiéry. Exploration of finite dimensional kac algebras and lattices of irreducible intermediate subfactors. Submitted, 83 pages, arXiv :0812.3044 [math.QA], December 2008.
- [DV02] Jean-Guillaume Dumas and Gilles Villard. Computing the rank of large sparse matrices over finite fields. In *CASC'2002 : Computer Algebra in Scientific Computing, Big Yalta, Crimea, Ukraine.*, pages 22 – 27, Septembre 2002.
- [ES92] Michel Enock and Jean-Marie Schwartz. *Kac algebras and duality of locally compact groups*. Springer-Verlag, Berlin, 1992. With a preface by Alain Connes, With a postface by Adrian Ocneanu.
- [Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3) :61–88, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).
- [Fra71] Roland Fraïssé. *Cours de logique mathématique. Tome 1 : Relation et formule logique*. Gauthier-Villars Éditeur, Paris, 1971. Deuxième édition revue et modifiée, Collection de “Logique Mathématique”, Série A, No. 23.
- [Fra00] Roland Fraïssé. *Theory of relations*, volume 145 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, revised edition, 2000. With an appendix by Norbert Sauer.
- [FSS07] Ghislain Fourier, Anne Schilling, and Mark Shimozono. Demazure structure inside Kirillov-Reshetikhin crystals. *J. Algebra*, 309(1) :386–404, 2007.
- [FZVC94] Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoret. Comput. Sci.*, 132(1-2) :1–35, 1994.
- [Gau08] Philippe Gaucher. Combinatorics of labelling in higher dimensional automata. arXiv :0808.1667v1 [math.AT], August 2008.
- [Ger04a] L. Gerritzen. Automorphisms of the planar tree power series algebra and the non-associative logarithm. *Serdica Math. J.*, 30(2-3) :135–158, 2004.
- [Ger04b] L. Gerritzen. Planar rooted trees and non-associative exponential series. *Adv. in Appl. Math.*, 33(2) :342–365, 2004.
- [Ges84] Ira M. Gessel. Multipartite P -partitions and inner products of skew Schur functions. In *Combinatorics and algebra (Boulder, Colo., 1983)*, volume 34 of *Contemp. Math.*, pages 289–317. Amer. Math. Soc., Providence, RI, 1984.
- [GH94] A. M. Garsia and M. Hamain. Orbit harmonics and graded representations. UCSD lecture notes, UCSD, 1994.
- [Gou04] Sandrine-Dominique Gouraud. *Utilisation des Structures Combinatoires pour le Test Statistique*. PhD thesis, Université Paris XI, June 2004.
- [Gri79] D. Ju. Grigoriev. Two reductions of the graph isomorphism to problems for polynomials. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 88 :56–61, 237–238, 1979. Studies in constructive mathematics and mathematical logic, VIII.
- [GS84] A. M. Garsia and D. Stanton. Group actions of Stanley - Reisner rings and invariants of permutation groups. *Adv. in Math.*, 51(2) :107–201, 1984.
- [GS04] Pierrick Gaudry and Éric Schost. Construction of secure random curves of genus 2 over prime fields. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 239–256. Springer, Berlin, 2004.
- [GST06] Pierrick Gaudry, Éric Schost, and Nicolas M. Thiéry. Evaluation properties of symmetric polynomials. *Internat. J. Algebra Comput.*, 16(3) :505–523, 2006. <http://hal.inria.fr/inria-00000629>.
- [GW03] A. M. Garsia and N. Wallach. Qsym over Sym is free. *J. Combin. Theory Ser. A*, 104(2) :217–263, 2003.
- [Hai92] Mark D. Haiman. Dual equivalence with applications, including a conjecture of Proctor. *Discrete Math.*, 99(1-3) :79–113, 1992.

- [Hai01] Mark Haiman. Hilbert schemes, polygraphs, and the macdonald positivity conjecture. *J. Amer. Math. Soc.*, 14 :941–1006, 2001.
- [Hiv04] Florent Hivert. Local actions of the symmetric group and generalisations of quasi-symmetric functions. Preprint, October 2004.
- [HKO⁺02] Goro Hatayama, Atsuo Kuniba, Masato Okado, Taichiro Takagi, and Zengo Tsuboi. Paths, crystals and fermionic formulae. In *MathPhys odyssey, 2001*, volume 23 of *Prog. Math. Phys.*, pages 205–272. Birkhäuser Boston, Boston, MA, 2002. arXiv :math/0102113v1 [math.QA].
- [HNT06] Florent Hivert, Jean-Christophe Novelli, and Jean-Yves Thibon. Yang-Baxter bases of 0-Hecke algebras and representation theory of 0-Ariki-Koike-Shoji algebras. *Adv. Math.*, 205(2) :504–548, 2006. arxiv :math.CO/0506546.
- [HR04] Thérèse Hardin and Renaud Rioboo. Les objets des mathématiques. *RSTI - L’objet*, Octobre 2004.
- [HR06] Ralf Hemmecke and Martin Rubey. Aldor-combinat : An implementation of combinatorial species, 2006. Available at <http://www.risc.uni-linz.ac.at/people/hemmecke/aldor/combinat/>.
- [HST08] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as degenerate affine Hecke algebras. *Discrete Math. Theor. Comput. Sci.*, September 2008. In print.
- [HST09] Florent Hivert, Anne Schilling, and Nicolas M. Thiéry. Hecke group algebras as quotients of affine Hecke algebras at level 0. *J. Combin. Theory Ser. A*, November 2009. doi :10.1016/j.jcta.2008.11.010, arXiv :0804.3781 [math.RT].
- [HT04a] Florent Hivert and Nicolas M. Thiéry. Deformation of symmetric functions and the rational Steenrod algebra. In *Invariant theory in all characteristics*, volume 35 of *CRM Proc. Lecture Notes*, pages 91–125. Amer. Math. Soc., Providence, RI, 2004. arXiv :0812.3056v1 [math.CO].
- [HT04b] Florent Hivert and Nicolas M. Thiéry. MuPAD-Combinat, an open-source package for research in algebraic combinatorics. *Sém. Lothar. Combin.*, 51 :Art. B51z, 70 pp. (electronic), 2004. <http://mupad-combinat.sf.net/>.
- [HT06] Florent Hivert and Nicolas M. Thiéry. Representation theories of some towers of algebras related to the symmetric groups and their Hecke algebras. In *Proceedings of FPSAC’06 San Diego*, 2006. arXiv :math/0607391v2 [math.RT].
- [HT08] Florent Hivert and Nicolas M. Thiéry. The Hecke group algebra of a Coxeter group and its representation theory. *Journal of Algebra*, November 2008. doi :10.1016/j.jalgebra.2008.09.039, arXiv :0711.1561 [math.RT].
- [Jon83] V. F. R. Jones. Index for subfactors. *Invent. Math.*, 72(1) :1–25, 1983.
- [Jon85] Vaughan F. R. Jones. A polynomial invariant for knots via von Neumann algebras. *Bull. Amer. Math. Soc. (N.S.)*, 12(1) :103–111, 1985.
- [Joy81] André Joyal. Une théorie combinatoire des séries formelles. *Adv. in Math.*, 42(1) :1–82, 1981.
- [JS97] V. Jones and V. S. Sunder. *Introduction to subfactors*, volume 234 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1997.
- [Kas05] Masaki Kashiwara. Level zero fundamental representations over quantized affine algebras and Demazure modules. *Publ. Res. Inst. Math. Sci.*, 41(1) :223–250, 2005.
- [Kas08] Masaki Kashiwara. Private notes on finite-dimensional representations of quantized affine algebras. Private communication, 2008.
- [Kem93] Gregor Kemper. The *invar* package for calculating rings of invariants. IWR Preprint 93-94, University of Heidelberg, 1993.
- [Kem99] Gregor Kemper. Computational invariant theory. *Queen’s Papers in Pure and Applied Math.*, 1999.
- [Kin07] Simon King. Minimal generating sets of non-modular invariant rings of finite groups, 2007.
- [KM89] Deepak Kapur and Klaus Madlener. A completion procedure for computing a canonical basis for a k -subalgebra. In *Computers and mathematics (Cambridge, MA, 1989)*, pages 1–11. Springer, New York, 1989.
- [Koc82] W. L. Kocay. Some new methods in reconstruction theory. In *Combinatorial mathematics, IX (Brisbane, 1981)*, pages 89–114. Springer, Berlin, 1982.
- [KP66] G. I. Kac and V. G. Paljutkin. Finite ring groups. *Trudy Moskov. Mat. Obšč.*, 15 :224–261, 1966.

- [KT97] Daniel Krob and Jean-Yves Thibon. Noncommutative symmetric functions. IV. Quantum linear groups and Hecke algebras at $q = 0$. *J. Algebraic Combin.*, 6(4) :339–376, 1997.
- [Las03] Alain Lascoux. *Symmetric functions and combinatorial operators on polynomials*, volume 99 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 2003.
- [Las07] Alain Lascoux. *Macdonald and Schubert for dummies*. Phalanstère books, 2007.
- [Lit06] Dudley E. Littlewood. *The theory of group characters and matrix representations of groups*. AMS Chelsea Publishing, Providence, RI, 2006. Reprint of the second (1950) edition.
- [LS85] Alain Lascoux and Marcel-Paul Schützenberger. Formulaire raisonné des fonctions symétriques. *Publications Université Paris 7*, 1985.
- [Mac95] I. G. Macdonald. *Symmetric functions and Hall polynomials*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, second edition, 1995. With contributions by A. Zelevinsky, Oxford Science Publications.
- [MB07] Tomi Mikkonen and Xavier Buchwalder. The algebra of graph invariants - lower and upper bounds for minimal generators, 2007. arXiv :0712.0142v2 [math.CO].
- [McK97] Brendan D. McKay. Small graphs are reconstructible. *Australas. J. Combin.*, 15 :123–126, 1997.
- [Mil98] J. Lyn Miller. Effective algorithms for intrinsically computing SAGBI-Gröbner bases in a polynomial ring over a field. In *Gröbner bases and applications (Linz, 1998)*, pages 421–433. Cambridge Univ. Press, Cambridge, 1998.
- [MM03] Conrado Martínez and Xavier Molinero. Generic algorithms for the generation of combinatorial objects. In *Mathematical foundations of computer science 2003*, volume 2747 of *Lecture Notes in Comput. Sci.*, pages 572–581. Springer, Berlin, 2003.
- [MM04] Conrado Martínez and Xavier Molinero. An efficient generic algorithm for the generation of unlabelled cycles. In *Mathematics and computer science. III*, Trends Math., pages 187–197. Birkhäuser, Basel, 2004.
- [MM05] Conrado Martínez and Xavier Molinero. Efficient iteration in admissible combinatorial classes. *Theoret. Comput. Sci.*, 346(2-3) :388–417, 2005.
- [MMT06] Conrado Martínez, Xavier Molinero, and Nicolas M. Thiéry. Generació ordenada de classes d'estructures combinatòries (ordered generation of combinatorial structures). In *In Jornada de Recerca EPSEM 2006*, pages 83–84. EPSEM (Technical College of Manresa <<http://www.eupm.upc.es/>>), Remsa S.L. Manresa, April 2006. ISBN : 84-86784-05-0. ISBN : 978-84-86784-05-8.
- [Mnu92] V. B. Mnukhin. The k -orbit reconstruction and the orbit algebra. *Acta Appl. Math.*, 29(1-2) :83–117, 1992. Interactions between algebra and combinatorics.
- [NB07] Huilan Li Nantel Bergeron, Thomas Lam. Combinatorial hopf algebras and towers of algebras. arXiv :0710.3744v1 [math.CO], October 2007.
- [NTT04] Jean-Christophe Novelli, Jean-Yves Thibon, and Nicolas M. Thiéry. Algèbres de Hopf de graphes. *C. R. Math. Acad. Sci. Paris*, 339(9) :607–610, 2004. doi :10.1016/j.crma.2004.09.012, arXiv :0812.3407v1 [math.CO].
- [NV00] Dmitri Nikshych and Leonid Vainerman. A Galois correspondence for II_1 factors and quantum groupoids. *J. Funct. Anal.*, 178(1) :113–142, 2000.
- [OS08] Masato Okado and Anne Schilling. Existence of Kirillov-Reshetikhin crystals for nonexceptional types. *Represent. Theory*, 12 :186–207, 2008. arXiv :0706.2224v2 [math.QA].
- [Oud07] Johan Oudinet. Uniform random walks in very large models. In *RT '07 : Proceedings of the 2nd international workshop on Random testing*, pages 26–29, Atlanta, GA, USA, November 2007. ACM Press.
- [Pou77] Maurice Pouzet. Quelques remarques sur les résultats de Tutte concernant le problème de Ulam. *Publ. Dép. Math. (Lyon)*, 14(2) :1–8, 1977.
- [Pou78] Maurice Pouzet. *Sur la théorie des relations*. PhD thesis, Thèse d'état, Université Claude-Bernard, Lyon 1, 1978.
- [Pou79] Maurice Pouzet. Note sur le problème de Ulam. *J. Combin. Theory Ser. B*, 27(3) :231–236, 1979.
- [Pou06] Maurice Pouzet. The profile of relations. *Glob. J. Pure Appl. Math.*, 2(3) :237–272, 2006.

- [Pou08] Maurice Pouzet. When is the orbit algebra of a group an integral domain? Proof of a conjecture of P. J. Cameron. *Theor. Inform. Appl.*, 42(1) :83–103, 2008.
- [PT01] Maurice Pouzet and Nicolas M. Thiéry. Invariants algébriques de graphes et reconstruction. *C. R. Acad. Sci. Paris Sér. I Math.*, 333(9) :821–826, 2001. arXiv :0812.3079v1 [math.CO].
- [PT05] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras. In *Proceedings of FPSAC'05 Taormina*, 2005. arXiv :math/0601256 [math.CO].
- [PT08a] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras I : Quasi-polynomiality of the profile. 2008. In preparation, 21 pages.
- [PT08b] Maurice Pouzet and Nicolas M. Thiéry. Some relational structures with polynomial growth and their associated algebras II : Finite generation. 2008. In preparation, 20 pages.
- [PT08c] Maurice Pouzet and Nicolas M. Thiéry. Algebraic invariants of graphs and reconstruction, 2008. Preprint, 35 pages.
- [Ram03] Arun Ram. Affine Hecke algebras and generalized standard Young tableaux. *J. Algebra*, 260(1) :367–415, 2003. Special issue celebrating the 80th birthday of Robert Steinberg.
- [RS90] Lorenzo Robbiano and Moss Sweedler. Subalgebra bases. In *Commutative algebra (Salvador, 1988)*, pages 61–87. Springer, Berlin, 1990.
- [Saw03] Joe Sawada. A fast algorithm to generate necklaces with fixed content. *Theoret. Comput. Sci.*, 301(1-3) :477–489, 2003.
- [Sch72] M. P. Schützenberger. Promotion des morphismes d'ensembles ordonnés. *Discrete Math.*, 2 :73–94, 1972.
- [Sch08] Anne Schilling. Combinatorial structure of Kirillov-Reshetikhin crystals of type $D_n^{(1)}, B_n^{(1)}, A_{2n-1}^{(2)}$. *J. Algebra*, 319(7) :2938–2962, 2008.
- [Se03] N. J. A Sloane (editor). The on-line encyclopedia of integer sequences. Published electronically, 2003.
- [Shi02] Mark Shimozono. Affine type A crystal structure on tensor products of rectangles, Demazure characters, and nilpotent varieties. *J. Algebraic Combin.*, 15(2) :151–187, 2002.
- [Sin05] Asim Anand Sinha. Multiple dispatch and roles in oo languages : Fickle mr. Master's thesis, Department of Computing, Imperial College, London, 2005. <http://wwwhomes.doc.ic.ac.uk/~aas04/project/MSc/thesis.pdf>.
- [Ste03] John R. Stembridge. A local characterization of simply-laced crystals. *Trans. Amer. Math. Soc.*, 355(12) :4807–4823 (electronic), 2003.
- [Stu93] Bernd Sturmfels. *Algorithms in invariant theory*. Springer-Verlag, Vienna, 1993.
- [Thi99] Nicolas M. Thiéry. *Invariants algébriques de graphes et reconstruction ; une étude expérimentale*. PhD thesis, Université Lyon I, June 1999. N° d'ordre : 167-99.
- [Thi00a] Nicolas M. Thiéry. Algebraic invariants of graphs : a study based on computer exploration. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 34(3) :9–20, September 2000. arXiv :0812.3082v1 [math.CO].
- [Thi00b] Nicolas M. Thiéry. **PerMuVAR**, a library for computing in invariant rings of permutation groups. Software demonstration, MEGA 2000, Bath, UK, 2000.
- [Thi01] Nicolas M. Thiéry. Computing minimal generating sets of invariant rings of permutation groups with SAGBI-Gröbner basis. In *Discrete models : combinatorics, computation, and geometry (Paris, 2001)*, Discrete Math. Theor. Comput. Sci. Proc., AA, pages 315–328 (electronic). Maison Inform. Math. Discrèt. (MIMD), Paris, 2001.
- [TT04] Nicolas M. Thiéry and Stéphan Thomassé. Convex cones and SAGBI bases of permutation invariants. In *Invariant theory in all characteristics*, volume 35 of *CRM Proc. Lecture Notes*, pages 259–263. Amer. Math. Soc., Providence, RI, 2004. arXiv :math/0607380 [math.AC].
- [Tut79] W. T. Tutte. All the king's horses. A guide to reconstruction. In *Graph theory and related topics (Proc. Conf., Univ. Waterloo, Waterloo, Ont., 1977)*, pages 15–33. Academic Press, New York, 1979.
- [Ula60] S. M. Ulam. *A collection of mathematical problems*. Interscience Publishers, New York-London, 1960. Interscience Tracts in Pure and Applied Mathematics, no. 8.

- [Vai98] Leonid Vainerman. 2-cocycles and twisting of Kac algebras. *Comm. Math. Phys.*, 191(3) :697–721, 1998.
- [Wik] Wikipedia française.
- [Woo97] R. M. W. Wood. Differential operators and the Steenrod algebra. *Proc. London Math. Soc. (3)*, 75(1) :194–220, 1997.
- [Woo98] R. M. W. Wood. Problems in the Steenrod algebra. *Bull. London Math. Soc.*, 30(5) :449–517, 1998.
- [Woo01] R. M. W. Wood. Hit problems and the Steenrod algebra. In *Proceedings of the summer school Interactions between Algebraic topology and invariant theory, University of Ioannina, Greece, June 2000*. University of Ioannina reports, jun 2001.
- [Zel80] A. V. Zelevinsky. Induced representations of reductive p -adic groups. II. On irreducible representations of $GL(n)$. *Ann. Sci. École Norm. Sup. (4)*, 13(2) :165–210, 1980.
- [Zel81] Andrey V. Zelevinsky. *Representations of finite classical groups*, volume 869 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1981. A Hopf algebra approach.
- [Zim94] Paul Zimmermann. Gaia : A package for the random generation of combinatorial structures. *The Maple Technical Newsletter*, 1(1), 1994.

Abstract :

This manuscript synthesizes almost fifteen years of research in algebraic combinatorics, in order to highlight, theme by theme, its perspectives.

In part one, building on my thesis work, I use tools from commutative algebra, and in particular from invariant theory, to study isomorphism problems in combinatorics. I first consider algebras of graph invariants in relation with Ulam's reconstruction conjecture, and then, more generally, the age algebras of relational structures. This raises in return structural and algorithmic problems in the invariant theory of permutation groups.

In part two, the leitmotiv is the quest for simple yet rich combinatorial models to describe algebraic structures and their representations. This includes the Hecke group algebras of Coxeter groups which I introduced and which relate to the affine Hecke algebras, but also some finite dimensional Kac algebras in relation with inclusions of factors, and the rational Steenrod algebras. Beside being concrete and constructive, such combinatorial models shed light on certain algebraic phenomena and can lead to elegant and elementary proofs.

My favorite tool is computer exploration, and the algorithmic and effective aspects play a major role in this manuscript. In particular, I describe the international open source project *-Combinat which I founded back in 2000, and whose mission is to provide an extensible toolbox for computer exploration in algebraic combinatorics and to foster code sharing among researchers in this area. I present specific challenges that the development of this project raised, and the original algorithmic, design, and development model solutions I was led to develop.

Keywords :

Algebraic Combinatorics – Computer Algebra – Computer exploration

Graphs and Isomorphisms – Groups – Invariants – Representations

Commutative Algebras – Hopf and Kac Algebras – Iwahori-Hecke algebras

Résumé :

Ce mémoire fait la synthèse de presque quinze années de recherche en combinatoire algébrique afin d'en dégager, thème par thème, les perspectives.

Dans un premier volet, issu de ma thèse, j'utilise des outils d'algèbre commutative, et notamment de théorie des invariants, pour étudier des problèmes d'isomorphisme en combinatoire. Je m'intéresse tout d'abord aux algèbres d'invariants de graphes en lien avec la conjecture de reconstruction de Ulam puis, plus généralement, aux algèbres d'âges des structures relationnelles. Cela pose en retour des problèmes algorithmiques et structurels en théorie des invariants des groupes de permutations.

Dans un deuxième volet, le leitmotiv est la recherche de modèles combinatoires simples, mais riches, pour décrire des structures algébriques et leurs représentations. Cela inclut notamment les algèbres de Hecke groupe que j'ai associées aux groupes de Coxeter (en lien avec les algèbres de Hecke affine), mais aussi les algèbres de Kac de dimension finie (en lien avec les inclusions de facteurs) et les algèbres de Steenrod. Outre un aspect concret et effectif, de tels modèles apportent un éclairage sur certains phénomènes algébriques, et en particulier des démonstrations élégantes et élémentaires.

Mon outil principal est l'exploration informatique. Aussi, les aspects algorithmiques et effectifs tiennent une place particulière dans ce mémoire. En effet, je coordonne le développement du projet logiciel international **-Combinat* depuis sa création en 2000. Sa mission est de fournir une boîte à outils extensible pour l'exploration informatique en combinatoire algébrique et de promouvoir la mutualisation de code entre les chercheurs de ce domaine. Je détaille notamment les défis particuliers rencontrés lors de son développement, et les solutions originales que ceux-ci m'ont amené à mettre au point, tant du point de vue de l'algorithmique que de la conception ou du choix du modèle de développement.

Mots clefs :

Combinatoire Algébrique – Calcul Formel – Exploration Informatique

Graphes et Isomorphisme – Groupes – Invariants – Représentations

Algèbres commutatives – Algèbres de Hopf et de Kac – Algèbres de Iwahori-Hecke

This figure "logo_UFR.png" is available in "png" format from:

<http://arxiv.org/ps/0912.2619v1>

This figure "logo_u-psud.png" is available in "png" format from:

<http://arxiv.org/ps/0912.2619v1>