Calcul des invariants de groupes de permutations par transformée de Fourier

Nicolas Borie

UNIVERSITÉ PARIS-SUD 11

ÉCOLE DOCTORALE DE MATHÉMATIQUES DE LA RÉGION PARIS-SUD

Laboratoire de mathématiques d'Orsay

THÈSE DE DOCTORAT

Spécialité mathématiques

soutenue le 07/12/2011

par

Nicolas BORIE

Calcul des invariants de groupes de permutations par transformée de Fourier

Directeur de thèse :

Nicolas M. THIÉRY	Maître de conférences habilité, Université Paris-Sud

Rapporteurs :

Harm DERKSEN	Professeur associé, University of Michigan at Ann Arbor
Victor REINER	Professeur, University of Minnesota at Minneapolis
Éric SCHOST	Professeur associé, University of Western Ontario, London

Composition du Jury :

Jean-Benoît BOST	Professeur, Université Paris-Sud
Jean-Charles FAUGÈRE	Directeur de Recherche, Université Pierre et Marie Curie
Marc GIUSTI	Directeur de Recherche, École Polytechnique
Yannis MANOUSSAKIS	Professeur, Université Paris-Sud
Jean-Yves THIBON	Professeur, Université Paris-Est Marne-la-Vallée
Annick VALIBOUZE	Professeur, Université Pierre et Marie Curie

Remerciements

Je remercie tout d'abord Nicolas Thiéry, grâce à qui, j'ai pu vivre cette formidable aventure durant ces trois dernières années. Outre l'expérience scientifique, l'expérience humaine que constitue la préparation d'une thèse est très enrichissante. Le soin qu'il a apporté à l'encadrement de mon entrée dans le monde de la recherche est allé au delà du travail du directeur de thèse. Il a su aussi lire en moi mes aspirations mathématiques pour guider mon intuition vers des tâches et problèmes auxquelles je suis sensible.

Je remercie très chaleureusement Harm Derksen, Vic Reiner et Éric Schost pour avoir accepté de rapporter cette thèse. Située à cheval entre la théorie usuelle des invariants, la combinatoire algébrique et le calcul formel, j'espère que l'originale connexion de domaines que propose cette thèse suscitera leur intérêt.

Je remercie Jean-Benoît Bost, Jean-Charles Faugère, Marc Giusti, Yannis Manoussakis, Jean-Yves Thibon et Annick Valibouze d'avoir accepté de participer en tant que jury à la soutenance de cette thèse.

Le laboratoire de mathématiques d'Orsay et l'école doctorale de la région Paris-sud ont étés d'un support exemplaire. J'ai pu ainsi voyager suffisamment pour faire les rencontres décisives mais aussi défendre mes propres travaux dans différents meetings. La rencontre avec François Bergeron, lorsque ce dernier fût professeur invité au laboratoire d'Orsay, fût particulièrement fructueuse. Je remercie aussi Odile Brandière et son équipe pédagogique pour m'avoir si bien accueilli. Mes premiers enseignements ont ainsi pu être dispensés dans un très bon climat.

Je remercie avec amitié toute l'équipe de combinatoire du laboratoire Gaspard-Monge. En particulier Adrien Boussicault, Hayat Cheballah, Valentin Féray, Samuele Giraudo, Florent Hivert, Alain Lascoux, Jean-Christophe Novelli, Viviane Pons et Jean-Yves Thibon. J'ai vraiment apprécié le temps qu'ont pris les vétérans pour me guider, me conseiller ou m'imprimer en vitesse un article que je devais lire ; mais aussi les moments passés ensemble à réfléchir avec mes homologues plus jeunes, à résoudre un problème de recherche ou d'implantation informatique.

Je remercie Jason Bandlow, Tom Denton, Brant Jones, Steve Pon et Anne Schilling que j'ai côtoyés lors de deux longs séjours à l'University of California at Davis. L'accueil là-bas fût exceptionnel et je suis sûrement revenu grandi de cette expérience de début de thèse. Ce fût aussi l'occasion de travailler ma maîtrise de la langue anglaise durant presque six mois sur ces quatre dernières années.

Je remercie les contributeurs au logiciel `Sage` pour toutes les rencontres fructueuses que j'ai pu avoir avec eux. Notamment Karl-Dieter Crisman, Nathann Cohen, Vincent Delecroix, Simon King, Sébastion Labbé, Robert Miller, Franco Saliola, Mike Hansen et Paul Zimmermann.

Je remercie ma famille et tous mes proches pour leur bienveillant soutien.

Table des matières

I Introduction, rappels et notations **13**

1 Rappels et Notations **19**

1.1 Groupe de permutations, de matrices et de réflexions complexes 19

1.2 Algèbres polynomiales, algèbres graduées 22

2 Généralités sur les anneaux d'invariants **25**

2.1 Anneau des polynômes invariants sous l'action d'un groupe de matrices 25

2.2 Polynômes symétriques 27

2.3 Invariants Primitifs 28

2.4 Graduation et séries de Hilbert 29

2.5 Système de générateurs de l'algèbre des invariants 31

2.6 L'algèbre des invariants est de Cohen-Macaulay 33

II Génération efficace de représentants d'orbites **37**

2.7 Action sur les vecteurs d'entiers, vecteurs canoniques 39

2.8 Arbre de génération des vecteurs d'entiers 42

2.9 Tests efficaces du caractère canonique 45

III Une approche par évaluation pour le calcul des invariants **53**

3 Calcul des invariants secondaires des groupes de permutations par évaluation **55**

3.1 Enjeux et objectifs : confiner dans un «petit» quotient 55

3.2 Démarche et cas tests 57

3.3 Un morphisme d'évaluation 58

3.4 Algorithme 62

3.5 Optimisations 63

4 Complexité et bancs d'essais **65**

4.1 Complexité théorique 65

4.2 Protocole de tests 66

4.3 Bancs d'essais 70

5 Polynômes de Schubert, candidats alternatifs pour les invariants **73**

5.1 Polynômes de Schubert 73

5.2 Choix alternatifs pour les invariants d'un groupe de permutation 84

IV Quotient de l'algèbre de Hecke affine au niveau 0 **87**

5.3 Introduction 89

5.4 Préliminaires 89

5.5 La groupe algèbre de Hecke comme quotient de l'algèbre de Hecke affine 94

5.6 Méthodologie 95

5.7 Exploration informatique 98

Annexe **105**

6 Polynômes harmoniques diagonaux déformés **105**

6.1 Introduction 105

6.2 Deformed harmonic polynomials for $G(m, p, n)$ 109

6.3 Inflating q-harmonic polynomials from $\mathfrak{S}_n$ to $G(m, n)$ and $G(m, m, n)$ 110

6.4 Singular values 111

6.5 Complete study for $n = 2$ 112

7 Énumération modulo l'action d'un groupe de permutation **117**

7.1 `SearchForest`, un outil pour le parcours paresseux d'arbres de recherche 118

7.2 Construction des canoniques sous l'action d'un groupe de permutations 121

8 L'anneau des invariants d'un groupe de permutation dans `Sage` **125**

8.1 Esprit de l'implantation, «building the car» 125

8.2 L'anneau abstrait des invariants et ses différentes représentations 126

8.3 Calcul des invariants secondaires 130

Index des notations **138**

Index **138**

Table des figures

2.1 Graphe de l'arbre de génération des vecteurs d'entiers 43

2.2 Arbre de génération des canoniques pour le groupe cyclique d'ordre 3 44

2.3 Nombre de canoniques sous l'escalier en fonction de $n!/|G|$ 48

2.4 Nombre de canoniques sous l'escalier en fonction de $n!$ 48

2.5 Nombre de canoniques non décroissants sous l'escalier en fonction de $n!/|G|$ . . . 49

2.6 Temps de calcul des canoniques sous l'escalier en fonction de $n!/|G|$ 50

2.7 Temps de calcul des canoniques sous l'escalier en fonction de leur nombre. 50

2.8 Temps pour générer 1000 canoniques en fonction du degré 51

4.1 Bancs d'essais comparatif entre `Sage` et `Singular` 71

4.2 Complexité suivant le nombre de points d'évaluation 72

5.1 Polynômes de Schubert du groupe $\mathfrak{S}_3$ 76

5.2 Graphe de Cayley du groupe $\mathfrak{S}_3$ 90

6.1 Structure of q-Harmonic polynomials of $G(4, 2)$. 114

8.1 Schéma d'implantation du module sur les invariants 126

Liste des tableaux

5.1 Familles remarquables de polynômes pour l'anneau des invariants 85
5.2 Réflections et projections simples pour le type de Cartan A_4 93
5.3 Introduction de s_0 et π_0 pour le type de Cartan A_4 93
5.4 Complexité de la taille de la réponse au pire en nombre de rationnels à calculer . 98
5.5 Dimension progressive suivant la longueur des éléments engendrant 99
5.6 Dimensions aux racines de l'unité 99
5.7 Borne théorique et plus grande hauteur k telle que la racine k-ième est mauvaise. 100

Liste des algorithmes

1 Énumération des canoniques sous l'escalier 44
2 Test de canonicité des vecteurs d'entiers 46
3 Algorithme classique de calcul des invariants secondaires 56
4 Calcul des invariants secondaires par évaluation 62
5 algorithm by length 98

Première partie

Introduction, rappels et notations

Le fil rouge de cette thèse, inscrite dans la combinatoire algébrique, est la théorie des invariants effective, et notamment, une nouvelle approche par évaluation pour calculer les invariants d'un groupe de permutations dans le cas non modulaire. Comme préliminaire, j'ai été amené à développer et implanter des algorithmes efficaces pour engendrer des objets à un isomorphisme près. Enfin, diverses rencontres ont motivé des travaux connexes ; tout d'abord deux visites à l'University of California at Davis se sont soldées par un travail à propos des systèmes de racines et des algèbres de Hecke. Ensuite, lors d'un séjour à Orsay de François Bergeron, nous avons collaboré sur une conjecture de Wood sur les polynômes harmoniques déformés. Dans chacun de ces sujets, l'implantation et l'exploration informatique, en particulier avec `Sage`, ont joué un rôle majeur.

Théorie des invariants effective

La théorie des invariants a été, depuis un siècle et demi, un axe riche et central de recherche en algèbre, avec des applications pratiques [DK02, § 5] en théorie de Galois(see e.g. [Col97b], [Abd00], [GK00]), dans l'exploitation du groupe des symétries dans la résolution de systèmes d'équations (voir e.g. [Col97a], [Gat90], [Stu93, § 2.6], [FR09]) ou encore en mathématiques discrètes (see e.g. [Thi00, PT01] pour la motivation première du second auteur). La littérature contient des résultats profonds et explicites pour certaines classes de groupes comme les groupes de réflexions complexes ou les groupes réductifs classiques mais aussi quelques résultats généraux valables pour n'importe quel groupe. Étant donné le niveau de généralité, on ne peut guère espérer obtenir en général des résultats à la fois explicites et fins pour tous les groupes de permutations. Ainsi ce sujet a développé un côté effectif très rapidement : étant donné un groupe, l'on veut calculer les propriétés de son anneau des invariants. Sous l'impulsion du calcul symbolique, des méthodes effectives, avec leur différentes implantations informatiques, ont fleuri durant les vingt dernières années [Kem93, Stu93, MN08, Her03, Thi01, DK02, Kin07b, Kin07a]. Néanmoins, des progrès sont nécessaires pour atteindre des exemples de taille intéressante et ainsi élargir le spectre des applications. Les programmes actuels arrivent à traiter le problème pour tous les sous-groupes du groupe symétrique d'ordre 7. Au delà de cette limite, le problème, pour un sous-groupe de $\mathfrak{S}_n$, nécessite typiquement de faire de l'algèbre linéaire sur un espace dont la dimension est de l'ordre de $n!$; toute approche frontale est veine.

Une obstruction importante vient du fait que les algorithmes dépendent principalement de l'efficacité des calculs dans certains quotients de l'anneau des invariants ; ceux-ci sont usuellement traités avec des techniques d'élimination (bases de Gröbner ou bases de SAGBI-Gröbner) mais ces méthodes se comportent mal avec les symétries. Aussi, une approche alternative prometteuse est l'utilisation de techniques d'évaluations. Cela a par exemple été utilisé avec succès pour réécrire les invariants en termes de générateurs déjà connus de l'anneau des invariants [GST06, DSW09].

L'objectif principal de cette thèse est d'explorer l'utilisation de techniques d'évaluations pour le calcul de l'algèbre des invariants elle-même (générateurs, ...).

Dans cette étude, et comme test récurrent, nous nous focalisons sur le calcul des invariants secondaires d'un groupe de permutations dans le cas non modulaire. Pour cela, nous montrons comment mener les calculs dans un quotient adapté en utilisant des évaluations sur des points bien choisis. De cela dérive notre contribution principale : **un nouvel algorithme pour calculer les invariants secondaires par évaluation, et l'analyse de sa complexité théorique et pratique**. Ce travail a été présenté lors de la conférence internationale MEGA 2011 [BT11], et fait l'objet d'une publication soumise [BT11].

Présentation du contexte

Dans un premier chapitre 1 de cette thèse, nous exposons des rappels et des notations sur les objets algébriques et combinatoires intervenant dans cette étude comme les groupes de permutations, les groupes finis de matrices ainsi que les groupes de réflexions complexes. Nous redéfinissons les structures d'algèbres graduées ainsi que leurs premières propriétés. Nous exposons dans un chapitre suivant 2 la théorie classique des invariants de groupes finis de matrices dans le cas non-modulaire. Nous rappelons les résultats classiques ainsi que les algorithmes associés quand il y a lieu.

Génération d'objets à isomorphie près

Les groupes de permutations agissent naturellement sur les vecteurs en permutant leur différentes entrées. Dans la seconde partie II, nous présentons une approche pour énumérer les vecteurs d'entiers modulo l'action d'un groupe de permutation G, sous-groupe du groupe symétrique $\mathfrak{S}_n$. L'objectif est d'établir une algorithmique efficace pour construire tous les vecteurs d'entiers en n'en gardant qu'un seul par orbite sous l'action de G. Ce problème, souvent passé sous silence dans le contexte des invariants, est fondamental pour le calcul pratique des invariants. Un problème connexe est abordé dans une implantation des changements de bases pour les fonctions symétriques dans [Val89].Nous proposons une structure arborescente inspirée de l'«orderly generation» [Ser03] pour traiter ce problème. Puis nous présentons des bancs d'essais sur notre implantation en `Sage` pour donner au lecteur une idée de la complexité pratique d'une telle algorithmique.

Calcul d'invariants secondaires par évaluation

Dans la troisième partie III, nous aborderons l'enjeu principal de cette thèse : construire explicitement les invariants secondaires donnant une base de l'anneau des invariants comme module libre sur les polynômes symétriques. Ainsi on obtient aussi une décomposition de Hironaka de l'algèbre des invariants. Dans un premier chapitre 3, nous présentons les prérequis théoriques pour le calcul par évaluation d'invariants secondaires associés aux polynômes symétriques. Nous montrons qu'en déformant légèrement l'idéal $\langle e_1, e_2, \ldots, e_n\rangle$ engendré par les polynômes symétriques, les calculs modulo cet idéal peuvent être confinés dans un espace de dimension coïncidant avec celle du quotient voulu $n!/|G|$ (c'est-à-dire le nombre invariants secondaires). On peut ensuite exploiter la graduation pour retransférer les résultats dans le quotient original sans introduire de complication majeure.

Nous en déduisons au chapitre 3 un algorithme 4 pour calculer des invariants secondaires relatifs au système de primaires formé par les polynômes symétriques élémentaires. Dans le chapitre 4 suivant, nous obtenons une borne de complexité théorique pour cet algorithme $\mathcal{O}(n!^2 + \frac{n!^3}{|G|^2})$ pour G un sous-groupe de $\mathfrak{S}_n$, et présentons des bancs d'essais systématiques sur les groupes de permutations transitifs. Nous traitons en particulier tous les groupes de permutations transitif G, sous-groupe de S_n pour lesquels $\frac{n!}{|G|} \leqslant 1000$. Nos principales conclusions sont que l'on obtient un meilleur contrôle de la complexité via une bonne exploitation des symétries (gain d'un facteur d'au moins $1/|G|$). De ce fait, l'approche est d'autant plus intéressante lorsque le groupe est gros (i.e. tels que le nombre $n!/|G|$ d'invariants secondaires est petit). Ainsi nous arrivons à traiter un exemple pour $n = 14$, pour lequel le groupe est de cardinal 50.803.200 et avec les polynômes symétriques comme invariants primaires, il a fallu calculer 1716 invariants secondaires. Un calcul d'algèbre linéaire en dimension 14! n'aurait pu construire cette famille en un temps raisonnable.

Évaluation des polynômes de Schubert

Les points d'évaluations utilisés ont un fort parfum combinatoire. Ainsi l'évaluation des polynômes symétriques de Schur sur ces points est-elle bien connue [Las08]. De même, les polynômes de Schubert, objets centraux dans la combinatoire des polynômes multivariés, sont *définis* par leur propriétés d'évaluations sur certains de ces points.

Aussi pourrait-on espérer exploiter l'approche par évaluation non seulement pour les calculs effectifs, mais aussi comme **outil théorique**. Cela pourrait par exemple fournir un angle d'attaque pour un problème ancien : la description combinatoire des invariants secondaires pour un groupe de permutations. Ce problème a été uniquement résolu dans le cas des sous-groupes de Young du groupe symétrique $\mathfrak{S}_n$ par Garsia et Stanton [GS84]. Aussi, à titre de premier pas dans cette direction comme pour le calcul, nous étudions en section 5.1.2 les propriétés d'évaluation des polynômes de Schubert [Pro00].

Un des points de départ de notre approche est le choix d'une base des polynômes multivariés comme module sur les polynômes symétriques ; ainsi, dans l'objectif de raffiner notre approche, nous utilisons le fait que les polynômes de Schubert ont aussi cette propriété et qu'ils peuvent être définis comme polynômes d'interpolation en des points corrélés avec l'approche par évaluation. Nous discutons de l'apport et de la faisabilité d'un calcul des invariants secondaires construit à partir de polynômes de Schubert.

Quotients au niveau 0 de l'algèbre de Hecke affine aux racines de l'unités

La troisième partie contient un travail original, dont le point de départ est un article de Hivert, Schilling et Thiéry intitulé «Hecke group algebras as quotients of affine Heckes algebras at level 0» [HST09]. L'algèbre de Hecke Groupe $H\mathring{W}$ d'un groupe de Coxeter fini $\mathring{W}$, définie par Florent Hivert et Nicolas M. Thiéry, est obtenue de $\mathring{W}$ en recollant de manière appropriée sa 0-Hecke algèbre et son algèbre de groupe. Avec la contribution d'Anne Schilling, ils donnèrent une construction alternative et équivalente lorsque $\mathring{W}$ est le Weyl fini associé au groupe de Weyl affine W. Ils prouvèrent que pour q non racine de l'unité de petit degré, $H\mathring{W}$ est un quotient naturel de l'algèbre de Hecke affine $H(W)(q)$ à travers sa représentation au niveau 0. Dans ce chapitre, nous explorons le comportement de ce quotient pour les racines de l'unité de petit ordre. Nous décrivons le contexte mathématique pour approcher ce problème, notre méthodologie pour l'exploration informatique et nos résultats. Nous énonçons aussi quelques conjectures.

Polynômes harmoniques diagonaux déformés pour les groupes de réflections complexes

En première annexe 6 à cette thèse, nous plaçons un travail effectué en collaboration avec François Bergeron et Nicolas M. Thiéry, intitulé «Deformed diagonal harmonic polynomials for complex reflection groups». Nous y introduisons une déformation de l'espace des polynômes harmoniques (multi-diagonaux) pour tout groupe de réflexions complexes de la forme $W = G(m, p, n)$, et soutenons l'hypothèse que cet espace est toujours isomorphe, en tant que W-module gradué, à l'espace d'origine. Ce travail accepté dans les Proceeding de la conférence internationale FPSAC 2011 a été présenté lors de cette dernière [BBT11].

Exploration informatique et implantation

L'implantation et l'exploration informatique ont été des piliers fondamentaux de cette thèse. La littérature existante sur le domaine des invariants présente de nombreux algorithmes mais ceux-ci sont rarement accompagnés d'analyse de complexité. Comme la complexité théorique

reste difficile à contrôler, l'efficacité en pratique est un aspect essentiel de la validation. Ainsi, une implantation rigoureuse et largement testée de nos algorithmes fût un des points principaux du cahier des charges de nos investigations.

Cette thèse aboutissant à des méthodes automatiques et effectives de calcul des invariants, nous avons pris parti de décrire complètement tous les procédés mathématiques et informatiques apparaissant dans cette étude. En effet, une implantation efficace du calcul des invariants requiert de nombreux outils préliminaires, certains portant sur les groupes comme la génération de vecteurs d'entiers modulo l'action d'un groupe de permutations, l'expression d'une transversale d'un quotient de groupe G/H, les classes de conjugaisons d'un groupe, etc. Mais aussi des fonctionnalités sur les séries, de l'arithmétique sur les corps cyclotomiques, de l'algèbre linéaire et bien d'autres choses aussi précisées ici [BT11]. Cela nous a orienté vers la plateforme libre de calcul mathématiques `Sage`, où la plupart de ces fonctionnalités étaient déjà présentes, et où nous avons pu rajouter les pièces manquantes. L'apport de `GAP`, interfacé depuis `Sage`, nous permet de récupérer à peu près toute la technologie algorithmique connue sur les groupes et en particulier les groupes de permutations. Les corps de nombres et l'algèbre linéaire sur ces derniers sont déjà implantés dans `Sage`. Il en est de même pour la petite arithmétique sur les séries.

En cet Octobre 2011, le code pour le calcul des invariants est toujours en développement. Il est disponible sur internet via le serveur de «patch» expérimentaux `Sage-Combinat`. Parmi les fonctionnalités, certaines sont déjà intégrées dans `Sage` comme le module `SearchForest`, la génération modulo l'action d'un groupe de permutations est soumise, en arbitrage par les autres contributeurs et le module principal sur les invariants nécessite une finalisation avant d'être proposé à l'intégration. Durant ces trois dernières années, l'auteur de cette thèse a rapporté 23 rapports de bogues ou suggestions d'améliorations sur la plateforme de contribution de `Sage`, il a été auteur de 14 contributions correctives ou de développement de nouvelles fonctionnalités et a participé à l'arbitrage de 14 contributions. Cette contribution se chiffre aujourd'hui à environ 6000 lignes de code intégrées, avec documentation et tests systématiques.

En annexe de cette thèse, nous présentons trois modules du logiciel `Sage`. Les deux premiers 7.1 7.2, un peu techniques, permettent respectivement l'implantation de structure arborescente et l'énumération de vecteurs d'entier modulo l'action d'un groupe de permutations. Le dernier module 8 est une implantation de l'anneau des invariants d'un groupe de permutations avec les algorithmes présentés précédemment pour récupérer un système d'invariants secondaires.

Ce document est illustré par de nombreux exemples de calculs qui sont présentés tels qu'ils apparaissent dans la console de sage. On s'autorisera, dans certains cas, à réorganiser le texte issu de la console pour en faciliter sa lisibilité.

```
sage: 1+1
2
```

Les algorithmes apparaissant dans la thèse sont écrits en pseudo code avec une syntaxe voisine du code du langage `Python`.

Chapitre 1

Rappels et Notations

1.1 Groupe de permutations, de matrices et de réflexions complexes

1.1.1 Permutations, groupes de permutations

Définition 1.1.1. *Soit n un entier naturel, nous désignerons par $\mathfrak{S}_n$* le groupe symétrique *de degré n, c'est à dire le groupe formé par les bijections de l'ensemble $\{1, 2, \ldots, n\}$ munies de la loi de composition.*

Le cardinal de ce groupe est la factorielle de n, noté $n! = n \cdot (n-1) \cdot \cdots \cdot 2 \cdot 1$. Dans `Sage`, on peut construire un tel groupe via le constructeur `SymmetricGroup`.

```
sage: S3 = SymmetricGroup(3); S3
Symmetric group of order 3! as a permutation group
sage: S3.cardinality()
6
```

Rappelons que les éléments de ce groupe, appelés permutations, peuvent être représentés de différentes manières. La notation par mot d'une permutation p consiste à ne garder que le n-uplet des images $(p(1), p(2), \ldots, p(n))$ (ce qui décrit complètement et de manière unique p). Une autre notation classique est la représentation par produit de cycles disjoints. Il s'agit d'une liste de cycles $(c_1, c_2, \ldots, c_r)$, où chaque cycle $c_i = (a_1, a_2, \ldots, a_l)$ détermine les images suivantes $p(a_1) = a_2, \ldots, p(a_k) = a_{k+1}, \ldots, p(a_l) = a_1$. Chaque cycle étant composé d'éléments disjoints, chaque nombre de $\{1, \ldots, n\}$ apparaît au plus une fois. Si une valeur a n'apparaît dans aucun des cycles, c'est que $p(a) = a$ (tous les cycles (a) avec une seule valeur ainsi que le cycle vide $()$ décrivent l'identité).

La méthode `list` appliquée au groupe symétrique dans `Sage` permet de déployer les éléments du groupe en tant que produits de cycles à supports disjoints.

```
sage: [Permutation(p) for p in S3]
```

```
[[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
sage: S3.list()
[(), (2,3), (1,2), (1,2,3), (1,3,2), (1,3)]
```

Définition 1.1.2 (Groupe de permutations)**.** *Un* groupe de permutations G *est un sous-groupe de* $\mathfrak{S}_n$ *pour un certain entier* n*.*

Nous appellerons *degré* du groupe de permutations G l'entier n.

La méthode la plus aisée et la plus courante pour décrire un groupe de permutations consiste à fournir une liste de générateurs, éléments vivants dans $\mathfrak{S}_n$ pour un certain n. Voici, par exemple, le groupe cyclique C_3 d'ordre 3, engendré par le cycle $(1, 2, 3)$. Ainsi, on notera $C_3 := \langle (1, 2, 3) \rangle$.

```
sage: G = PermutationGroup([[(1,2,3)]]); G
Permutation Group with generators [(1,2,3)]
sage: G.cardinality()
3
sage: G.list()
[(), (1,2,3), (1,3,2)]
```

Soit $V = \langle e_1, \ldots, e_n \rangle_{\mathbb{K}}$ un espace vectoriel de dimension n. Les éléments du groupe symétrique $\mathfrak{S}_n$ agissent linéairement sur V par

$$\sigma \in \mathfrak{S}_n, (a_1, \ldots, a_n) \in \mathbb{K}^n : \sigma \cdot (a_1 e_1 + \cdots + a_n e_n) = a_1 e_{\sigma(1)} + \cdots + a_n e_{\sigma(n)}$$

Lorsque nous parlerons de la matrice de la permutation σ, nous désignerons la matrice de l'opérateur linéaire de $End(V)$ correspondant à l'action définie ci-dessus.

1.1.2 Code de Lehmer

Nous rappelons ici une autre manière de représenter les permutations.

Définition 1.1.3. *Soit* σ *une permutation du groupe symétrique* $\mathfrak{S}_n$*, son* code de Lehmer $\mathcal{C}(\sigma)$ *est le vecteur* $v = (v_1, v_2, \ldots, v_n)$ *tel que*

$$v_i = \text{cardinal}(\{j : j > i \text{ et } \sigma(i) > \sigma(j)\})$$

Nous ne reviendrons pas sur les détails décrivant la bijection entre les permutations du groupe symétrique $\mathfrak{S}_n$ et les codes de Lehmer sous l'escalier (i.e. $\forall i : v_i \leqslant n - i - 1$). Il existe aussi de nombreux algorithmes pour construire un code de Lehmer ou récupérer une permutation à partir de son code notamment dans [Knu73].

Comme les permutations, les codes de Lehmer s'adaptent aux règles de branchement du groupe symétrique $\mathfrak{S}_{n-1} \hookrightarrow \mathfrak{S}_n$. Lorsque l'on identifie une permutation σ avec $[\sigma, n{+}1, n{+}2, \ldots]$; côté code de Lehmer, l'opération correspondante consiste à rajouter les zéros finaux sur le code $v = \mathcal{C}(\sigma)$.

1.1.3 Groupes finis de matrices

Soit $\mathbb{K}$ un corps, nous noterons $GL_n(\mathbb{K})$ le *groupe linéaire* de taille n sur le corps $\mathbb{K}$, c'est à dire le groupe des matrices inversibles de taille n dont les coefficients sont à valeur dans $\mathbb{K}$.

Définition 1.1.4. *On désignera par* groupe fini de matrices *tout sous-groupe fini de* $GL_n(\mathbb{K})$

Par exemple, les groupes de permutations sont des groupes finis de matrices dont la représentation classique donne des matrices ayant pour coefficients 0 ou 1. On verra aussi que les groupes finis de Weyl, les groupes de Coxeter finis et les groupes de réflexions complexes finis admettent des représentations matricielles et forment eux aussi des groupes finis de matrices.

1.1.4 Groupes de réflexions complexes

Définition 1.1.5 (Réflexion complexe)**.** *Soit* $V = \mathbb{C}^n$ *un espace vectoriel complexe de dimension finie. Un élément de* $g \in GL(V)$ *est une* réflexion complexe *(ou pseudo-réflexion) si c'est un élément d'ordre fini qui stabilise un hyperplan de* V*.*

Toutes les matrices des permutations issues du groupe symétrique sont aussi des réflexions complexes.

Soit g une réflexion complexe et d l'entier minimal tel que $g^d = Id$. Soit ρ une racine primitive d-ième de l'unité, il existe une base $\mathcal{B}$ de V telle que la matrice de g soit diagonale :

$$mat_{\mathcal{B}}(g) = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 0 & \dots & 0 & \rho \end{pmatrix}.$$

Définition 1.1.6 (Groupe de réflexions complexes)**.** *Un* groupe de réflexions complexes G *est un groupe agissant sur un espace vectoriel complexe de dimension finie et qui est engendré par des réflexions complexes.*

Pour la suite, il n'est pas nécessaire de s'étendre plus largement sur la théorie des groupes de réflexions complexes. Toutefois, nous manipulerons deux grandes classes de ces groupes : les groupes symétriques colorés $G(m, n)$ ainsi que leurs sous-groupes $G(m, p, n)$.

Soit n et m deux entiers positifs. Soit S_n le sous-groupe de $GL_n(\mathbb{C})$ formé des matrices issues des permutations du groupe symétrique $\mathfrak{S}_n$. Soit ρ une racine primitive m-ième de l'unité et $S = \{1, \rho, \dots, \rho^{m-1}\}$. Le groupe de réflexions complexes $G(m, n)$ est le sous-groupe de $GL_n(\mathbb{C})$ formé de toutes les matrices de S_n à l'intérieur desquelles les entrées non nulles ont été remplacées par un des éléments de S sans restriction. Pour chaque matrice, il y a n entrées non nulles et pour chacune, il y a m choix possibles. Ainsi le cardinal de $G(m, n)$ est $m^n n!$.

Soit p un entier diviseur de m, le groupe $G(m, p, n)$ est le sous groupe de $G(m, n)$

$$G(m, p, n) := \{g \in G(m, n) \mid \det g^{m/p} = 1\}.$$

Définition 1.1.7 (Transversale d'un sous-groupe)**.** *Soient G un groupe et H un sous-groupe de G. Une partie T de G est appelée une* transversale à droite *de H dans G si toute classe à droite de G modulo H contient un et un seul élément de T. Cela revient à dire que tout élément de G s'écrit d'une et une seule façon sous la forme ht, avec h dans H et t dans T.*

En gardant les mêmes notations, on dira aussi que T est un *ensemble de représentants des classes à droite* du quotient G/H.

1.2 Algèbres polynomiales, algèbres graduées

1.2.1 Algèbres polynomiales, dimension de Krull

Définition 1.2.1 (Anneau des polynômes)**.** *Soit $\mathbb{K}$ un corps commutatif et $n \geqslant 2$ un entier. On appelle* anneau des polynômes multivariés en n variables *la $\mathbb{K}$-algèbre commutative libre engendrée par les variables formelles $x_1, x_2, \ldots, x_n = \boldsymbol{x}$. On la notera $\mathbb{K}[x_1, x_2, \ldots, x_n]$ ou $\mathbb{K}[\boldsymbol{x}]$.*

Définition 1.2.2. *Soit A une $\mathbb{K}$-algèbre commutative et I un ensemble d'élément de A. On dira que I est une partie de A* algébriquement indépendante *lorsque*

$$\forall P \in \mathbb{K}[(x_i)_{i\in I}] : \quad (P((x_i)_{i\in I}) = 0) \Rightarrow (P = 0)$$

Autrement dit, il n'existe pas de relation algébrique non triviale entre les éléments de I.

L'exemple zéro d'une telle définition se retrouve dans l'algèbre des polynômes en n indéterminées. En effet, la partie $\{x_1, x_2, \ldots, x_n\}$ est une partie algébriquement libre de $\mathbb{K}[x_1, x_2, \ldots, x_n]$ par définition.

Définition 1.2.3 (dimension de Krull)**.** *Soit A une algèbre commutative. La* dimension de Krull *de A est le cardinal maximum d'une famille algébriquement libre de A.*

Exemple 1.2.4. *$\mathbb{K}[x_1, \ldots, x_n]$ a pour dimension de Krull n.*

La dimension de Krull est un outil important pour l'étude des structures algébriques. On remarque que lorsque A est de dimension de Krull n, cela signifie en particulier qu'il existe un morphisme injectif d'algèbre de $\mathbb{K}[x_1, \ldots, x_n]$ dans A mais qu'il n'existe pas de morphisme d'algèbre injectif de $\mathbb{K}[x_1, \ldots, x_n, x_{n+1}]$ dans A. Coincé entre deux algèbres polynomiales, cela donne une idée de la taille de l'algèbre A.

Définition 1.2.5 (Système de paramètres homogènes)**.** *On appelle* système de paramètres homogènes *d'une algèbre graduée A de dimension de Krull n, une famille $\Theta_1, \ldots, \Theta_n$ formé de n éléments homogènes de A algébriquement indépendants.*

1.2.2 Algèbres graduées et séries de Hilbert

Définition 1.2.6 (Algèbre graduée)**.** *Soit A une algèbre sur un corps $\mathbb{K}$. A est une* algèbre graduée *si l'anneau A admet une décomposition en somme directe de groupes abéliens.*

$$A = \bigoplus_{n\in\mathbb{N}} A_n = A_0 \oplus A_1 \oplus A_2 \oplus \ldots$$

telle que la multiplication satisfait

$$x \in A_s, y \in A_r \Rightarrow xy \in A_{s+r}$$

Ainsi $A_s A_r \subset A_{s+r}$.

L'exemple canonique d'algèbre graduée est l'algèbre des polynômes en une indéterminée et par extension, toutes les algèbres polynomiales. Solution d'un problème universel, l'algèbre polynomiale en n indéterminées $(x_i)_{i\in I}$ est l'unique plus petite algèbre commutative engendrée par n éléments formels libres de relations. La graduation naturelle sur ces algèbres consiste à attribuer le degré 1 à chaque générateur x_i ; ce qui permet ensuite de déterminer le degré de tout polynôme par propagation.

Définition 1.2.7 (Algèbre graduée connexe)**.** *Soit A une algèbre graduée sur un corps $\mathbb{K}$; on suppose de plus que la graduation prend des valeurs dans $\mathbb{N}$. On dira que A est* connexe *lorsque le sous espace formé des éléments de A de degré 0 est égal au corps de base (i.e. $A_0 = \mathbb{K}$ avec les notations précédentes).*

Définition 1.2.8 (Série de Hilbert)**.** *Soit A une $\mathbb{K}$-algèbre graduée telle que pour tout entier n, le sous espace vectoriel A_n des éléments de A de degré n soit de dimension finie. On définit alors formellement une série entière $H(A, z)$*

$$H(A, z) := \sum_{i\in\mathbb{Z}} \dim(a_i) z^i$$

appelée série de Hilbert *de A.*

Dans cette thèse, nous manipulerons des sous-algèbres graduées de l'algèbre des polynômes en plusieurs indéterminées. Les graduations seront ainsi à valeurs dans $\mathbb{N}$ ainsi que l'indexation des séries de Hilbert.

L'exemple zéro est donc l'algèbre des polynômes multivariés qui donne

$$H(\mathbb{K}[\mathbf{x}], z) = \frac{1}{(1-z)^n}$$

Plus généralement, pour une algèbre graduée A librement engendrée par une famille finie de r éléments dont les degrés sont $e_1, e_2, \ldots, e_r$, on a

$$H(A, z) = \frac{1}{(1-z^{e_1})(1-z^{e_2})\ldots(1-z^{e_r})}$$

Les deux propositions techniques suivantes portent sur les algèbres graduées. Elles seront fondamentales dans la suite pour montrer que tous les systèmes d'invariants primaires présentent des propriétés intéressantes.

Proposition 1.2.9. *Soit A une algèbre graduée connexe de dimension de Krull n et soit $\{\Theta_1, \ldots, \Theta_n\}$ un système de paramètres homogènes pour A. Les deux assertions suivantes sont équivalentes*

(i) A est un module libre (nécessairement de rang fini) sur la sous-algèbre $\mathbb{K}[\Theta_1, \ldots, \Theta_n]$. Ainsi, il existe $\eta_i, \ldots, \eta_t \in A$ (que l'on peut choisir homogène) tel que

$$A = \bigoplus_{i=1}^{t} \eta_i \mathbb{K}[\Theta_1, \ldots, \Theta_n]$$

(ii) Pour tout système de paramètres homogènes $\Psi_1, \ldots, \Psi_n$ de A, A est un $\mathbb{K}[\Psi_1, \ldots, \Psi_n]$-module libre.

Cette première proposition apparaît dans les travaux de Stanley [Sta79], qui, lui-même, se réfère aux cours de Jean-Pierre Serre [Ser65] dans lesquels une preuve est proposée.

La proposition suivante donne une première caractérisation de cette famille $\eta_i, \ldots, \eta_t$ en vue d'une construction explicite.

Proposition 1.2.10 (Lemme de Nakayama gradué)**.** *Soit A une $\mathbb{K}$-algèbre graduée connexe de dimension de Krull n et soit $\{\Theta_1, \ldots, \Theta_n\}$ un système de paramètres homogènes pour A. Soit $\eta_i, \ldots, \eta_t$ une famille d'éléments homogène de A. Les deux assertions suivantes sont équivalentes :*

(i)
$$A = \bigoplus_{i=1}^{t} \eta_i \mathbb{K}[\Theta_1, \ldots, \Theta_n]$$

(ii) Les images des polynômes $\eta_i, \ldots, \eta_t$ dans le quotient $A/(\Theta_1, \ldots, \Theta_n)$ forment une base du $\mathbb{K}$-espace vectoriel $A/(\Theta_1, \ldots, \Theta_n)$.

Chapitre 2

Généralités sur les anneaux d'invariants

2.1 Anneau des polynômes invariants sous l'action d'un groupe fini de matrices

Définition 2.1.1. *Soit G un groupe de matrices, sous-groupe de $GL_n(\mathbb{K})$. On définit une* action des éléments M de G sur les polynômes *de $\mathbb{K}[\boldsymbol{x}]$ par*

$$P \in \mathbb{K}[\boldsymbol{x}], \quad \mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad (M \cdot P)(\mathbf{X}) := P(M\mathbf{X})$$

Par exemple, en 3 variables, pour $\mathbb{K} = \mathbb{C}$,

$$M = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & i \end{pmatrix} \quad \text{et} \quad P(x_1, x_2, x_3) = x_1^2 - x_2^2 + x_3$$

On obtient

$$M\mathbf{x} = \frac{1}{\sqrt{2}} \begin{pmatrix} x_1 + x_2 \\ -x_1 + x_2 \\ ix_3 \end{pmatrix}$$

$$M \cdot P(x_1, x_2, x_3) = (\frac{1}{\sqrt{2}}(x_1 + x_2))^2 + (\frac{1}{\sqrt{2}}(-x_1 + x_2))^2 + \frac{i}{\sqrt{2}}x_3 = x_1^2 + 2x_1x_2 + \frac{i}{\sqrt{2}}x_3$$

Dans le cas des groupes de permutations, l'action se réécrit plus simplement via l'action par position sur les vecteurs d'entiers.

Définition 2.1.2 (Action du Groupe symétrique sur les polynômes)**.** *Le groupe symétrique agit de manière naturelle sur $\mathbb{K}[\boldsymbol{x}]$ en permutant les variables des polynômes par*

$$\sigma \cdot P = P(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}) \quad \textit{pour } \sigma \in \mathfrak{S}_n \textit{et} P \in \mathbb{K}[\boldsymbol{x}]\,.$$

Par exemple, prenons la transposition simple $s_1 = (1, 2)$ et le polynôme $P = x_1^2 + x_1x_2 + x_2x_3 + x_4$, s_1 permute x_1 et x_2 et stabilise les autres variables. Ce qui donne :

$$(1, 2) \cdot P = x_2^2 + x_2x_1 + x_1x_3 + x_4$$

On remarque, que pour tout groupe de permutations G, les actions définies directement par permutation des arguments ou par action des matrices coïncident. Même si la théorie générale se focalise plus souvent sur les groupes finis ou réductifs de matrices, nous prendrons le temps de reformuler voire de raffiner les résultats pour les groupes de permutations.

Définition 2.1.3 (Polynôme invariant)**.** *Soit G un groupe de matrices, sous-groupe de $GL_n(\mathbb{K})$. Un polynôme P de $\mathbb{K}[\boldsymbol{x}]$ est dit* invariant *sous l'action de G si*

$$\forall M \in G : M \cdot P = P$$

On remarque immédiatement que la somme ou le produit de deux polynômes invariants sous l'action d'un groupe G est aussi un invariant sous l'action du même groupe. On introduit la sous-algèbre suivante :

Définition 2.1.4. *Soit G un groupe de matrices, sous-groupe de $GL_n(\mathbb{K})$. On appelle* anneau des invariants*, noté $\mathbb{K}[\boldsymbol{x}]^G$, la sous-algèbre de $\mathbb{K}[\boldsymbol{x}]$ formé par les polynômes invariants sous l'action de G.*

Cette algèbre, introduite ci-dessus, est l'objet central de cette thèse. Pour construire des éléments à l'intérieur de cette dernière, on utilise l'opération suivant :

Définition 2.1.5 (Opérateur de Reynolds)**.** *Soit G un groupe fini de matrices, lorsque $|G|$ est inversible dans $\mathbb{K}$, on définit un opérateur sur les polynômes multivariés à valeur dans l'algèbre des invariants sous l'action de G. Cette application R est appelée* opérateur de Reynolds*.*

$$\begin{array}{rccc} R: & \mathbb{K}[\boldsymbol{x}] & \longrightarrow & \mathbb{K}[\boldsymbol{x}]^G \\ & P & \longmapsto & \frac{1}{|G|}\displaystyle\sum_{M\in G} M\cdot P \end{array}$$

Proposition 2.1.6. *L'opérateur de Reynolds R est une projection graduée surjective sur $\mathbb{K}[\boldsymbol{x}]^G$ l'algèbre des invariants sous l'action de G. R est aussi un morphisme de $\mathbb{K}[\boldsymbol{x}]^G$-module.*

Démonstration. Comme pour tout groupe, un élément M agit bijectivement dans G. Ainsi, pour tout polynôme P :

$$M \cdot (\frac{1}{|G|}\sum_{N\in G} N\cdot P) = \frac{1}{|G|}\sum_{N\in G} MN\cdot P = \frac{1}{|G|}\sum_{N\in G} N\cdot P$$

$R(P)$ est donc un polynôme invariant. L'action des matrices préservant le degré, il est ainsi de même pour R. Un polynôme invariant P est invariant sous l'action de chaque élément de G, ainsi $R(P) = P$ lorsque $P \in \mathbb{K}[\mathbf{x}]^G$. Prenant valeur dans les invariants et les fixant point par point, on a $R^2 = R$. Enfin, l'opérateur de Reynolds vérifie :

$$\begin{aligned} \forall P \in \mathbb{K}[\mathbf{x}], \forall Q \in \mathbb{K}[\mathbf{x}]^G : R(P\cdot Q) &= \tfrac{1}{|G|}\sum_{M\in G} M(P\cdot Q) \\ &= \tfrac{1}{|G|}\sum_{M\in G} M(P)\cdot M(Q) \\ &= \tfrac{1}{|G|}\sum_{M\in G} M(P)\cdot Q \\ &= (\tfrac{1}{|G|}\sum_{M\in G} M(P))\cdot Q \\ &= R(P)\cdot Q; \end{aligned}$$

ce qui en fait un morphisme de module sur l'anneau $\mathbb{K}[\mathbf{x}]^G$ des invariants sous G. □

Pour tout entier $d \geqslant 0$, on peut former une base des polynômes en n variables de degrés d. Étant un morphisme gradué et surjectif, appliquer l'opérateur de Reynolds sur chaque élément de cette base fournit une famille génératrice du sous espace des invariants de degré d sous l'action G.

Définition 2.1.7 (Somme sur orbite)**.** *Soit G un groupe de permutations, sous-groupe de $\mathfrak{S}_n$, et soit $m = \boldsymbol{x}^{\alpha}$ un monôme où α est le vecteur d'entiers naturels $\alpha_1, \ldots, \alpha_n$ de ces exposants. On définit un opérateur appelé* somme sur orbite *et noté $\sum_{orb(G)}$ comme il suit :*

$$\sum_{orb(G)} (\boldsymbol{x}^{(\alpha_1,\ldots,\alpha_n)}) = \sum_{\sigma \in orb_G(\alpha_1,\ldots,\alpha_n)} \boldsymbol{x}^{\sigma\cdot(\alpha_1,\ldots,\alpha_n)}$$

On étend par linéarité cet opérateur à tout polynôme de $\mathbb{K}[\boldsymbol{x}]$.

L'intérêt des sommes sur orbite est de diminuer le nombre de calculs pour construire un invariant en partant d'un monôme. Aussi, la somme sur orbite ne génère pas de nouveau dénominateur. En particulier pour les petits degrés pour lesquels les vecteurs d'entiers encodant les monômes présente des groupes d'automorphismes importants, les sommes sur orbite présentent un nombre de termes négligeable devant $|G|$, le cardinal du groupe.

Exemple 2.1.8. *Soit le groupe cyclique $G = \langle (1,2,3,4) \rangle$, sous-groupe de $\mathfrak{S}_4$ et soit le polynôme $P = 6 * x_1^2 x_2 + x_1 x_3 = 6 * \boldsymbol{x}^{(2,1,0,0)} + \boldsymbol{x}^{(1,0,1,0)}$. L'opérateur de Reynolds appliqué à P donne*

$$R(P) = \frac{3}{2}(\boldsymbol{x}^{(2,1,0,0)} + \boldsymbol{x}^{(0,2,1,0)} + \boldsymbol{x}^{(0,0,2,1)} + \boldsymbol{x}^{(1,0,0,2)}) + \frac{1}{2}(\boldsymbol{x}^{(1,0,1,0)} + \boldsymbol{x}^{(0,1,0,1)}).$$

La somme sur orbite ne génère pas de nouveau dénominateur :

$$\sum_{orb(G)} (P) = 6 * (\boldsymbol{x}^{(2,1,0,0)} + \boldsymbol{x}^{(0,2,1,0)} + \boldsymbol{x}^{(0,0,2,1)} + \boldsymbol{x}^{(1,0,0,2)}) + \boldsymbol{x}^{(1,0,1,0)} + \boldsymbol{x}^{(0,1,0,1)}.$$

2.2 Polynômes symétriques

Commençons par un cas particulier bien connu et pour lequel il existe de nombreux résultats mathématiques. C'est le cas où le groupe de permutations choisi est le groupe symétrique $\mathfrak{S}_n$ tout entier.

Pour n un entier naturel non nul, les *polynômes symétriques* d'ordre n sont les éléments de l'algèbre des invariants sous l'action du groupe symétrique $\mathfrak{S}_n$. On dénote l'algèbre des polynômes symétriques en n variables $\mathrm{Sym}(\mathbf{x})$.

Définition 2.2.1 (Polynômes symétriques élémentaires)**.** *Soit n un entier naturel non nul et i tel que $1 \leqslant i \leqslant n$. Le i^{ime}* polynôme symétrique élémentaire *e_i en n variables est défini par :*

$$e_i := \sum_{\substack{F \subset \{1,\ldots,n\} \\ |F| = i}} (\prod_{j \in F} x_j)$$

Le polynôme e_i est la somme de tous les monômes produits de i variables différentes parmi les n variables. Par exemple, pour $n = 4$, on obtient :

$$\begin{cases} e_1 &= x_1 + x_2 + x_3 + x_4 \\ e_2 &= x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \\ e_3 &= x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \\ e_4 &= x_1x_2x_3x_4 \end{cases}$$

Étant une somme sur les parties d'un ensemble fini, on remarque que le nombre de monômes apparaissant dans le polynôme e_i en n variables est le nombre de combinaison $\binom{n}{i}$.

Theorème 2.2.2 (Théorème fondamental des polynômes symétriques)**.** *Soit n un entier naturel non nul et $\mathbb{K}$ un corps commutatif, l'algèbre des polynômes symétriques est une algèbre polynomiale sur $\mathbb{K}$ engendrée par les polynômes symétriques élémentaires.*

$$\mathrm{Sym}(\mathbf{x}) := \mathbb{K}[x_1, x_2, \ldots, x_n]^{\mathfrak{S}_n} \simeq \mathbb{K}[e_1, e_2, \ldots, e_n]$$

Un tel résultat caractérise complètement l'algèbre des invariants pour le groupe symétrique. Il donne directement les générateurs, la série de Hilbert, et les bases degré par degré des composantes homogènes exprimées à partir des générateurs. La série de Hilbert est construite à partir des degrés des générateurs, elle est donnée par

$$H(\mathrm{Sym}(\mathbf{x}), z) = \frac{1}{(1-z)(1-z^2)\ldots(1-z^n)}$$

Définition 2.2.3 (Monôme sous l'escalier)**.** *Soit n un entier naturel, un monôme $m = \boldsymbol{x}^{\alpha}$ en n variables sera dit* sous l'escalier *dans $\mathbb{K}[\boldsymbol{x}]$ lorsque le vecteur d'entiers α de ces exposants est inférieur, terme à terme, au vecteur $(n-1, n-2, \ldots, 1, 0)$.*

Pour tout n, il y a $n!$ monômes sous l'escalier différents. On remarque par ailleurs que cette famille peut indexer les éléments du groupe symétrique $\mathfrak{S}_n$; ce sont les codes de Lehmer.

Theorème 2.2.4. *Soit n un entier naturel. L'algèbre des polynômes multivariés $\mathbb{K}[\boldsymbol{x}]$ est un module libre de rang fini sur l'algèbre des polynômes symétriques. Le rang de ce module est $n!$ et une base est donnée, par exemple, par les monômes sous l'escalier :*

$$\mathbb{K}[\boldsymbol{x}] \simeq \bigoplus_{\forall i: a_i \leqslant n-i} x_1^{a_1} \ldots x_n^{a_n} \cdot \mathbb{K}[e_1, e_2, \ldots, e_n]$$

2.3 Invariants Primitifs

Soit H un groupe de permutations et P un polynôme de $\mathbb{K}[\mathbf{x}]$. Le *stabilisateur* de P sous l'action de H est

$$Stab_H(P) = \{\sigma \in H | \sigma \cdot P = P\}.$$

Soit G un groupe de permutations, sous-groupe de $\mathfrak{S}_n$, on remarque l'anneau des invariants sous l'action de G peut être alors défini de la manière suivante :

$$\mathbb{K}[\mathbf{x}]^G := \{P \in \mathbb{K}[\mathbf{x}] | G \subset Stab_{\mathfrak{S}_n}(P)\}$$

Comme énoncé dans [Abd99], nous rappelons la définition d'un invariant primitif.

Définition 2.3.1 (G-invariant H-primitif)**.** *Soit G et H deux groupes de permutations tels que $G \subset H$, un polynôme $P \in \mathbb{K}[\boldsymbol{x}]$ est dit* G-invariant H-primitif *lorsque*

$$Stab_H(P) = G$$

Si $H = \mathfrak{S}_n$, alors P est appelé G-invariant primitif (absolu)*.*

Une première remarque est que de tels polynômes existent toujours et sont faciles à exhiber. Pour tout sous-groupe G de $\mathfrak{S}_n$, prenant un second alphabet $y_1, \ldots, y_n$ la somme sur orbite de l'interpolateur de Lagrange :

$$P := \prod_{1 \leqslant i < j \leqslant 1} (x_i - y_j)$$

est un G-invariant primitif (en spécialisant les y_i sur des éléments du corps de base, on obtient un polynôme dans $\mathbb{K}[\mathbf{x}]$). Le problème est qu'ici, ce polynôme est de degré très important $\binom{n}{2}$ et son expansion dans la base des monômes déploie $n!$ termes. Le problème devient beaucoup plus difficile lorsque l'on recherche des invariants primitifs de petit degré et possédant un minimum de termes dans son expansion dans la base des monômes.

2.4 Graduation et séries de Hilbert

Un outil pour borner, quantifier et stopper les calculs au bon moment est l'utilisation d'une graduation. La graduation est une information contenue dans chaque élément, cette information est plus "légère" que la connaissance complète de l'élément considéré, mais permet d'effectuer des calculs clés rapidement pour lesquels seule la connaissance du degré est cruciale.

Pour tout groupe de permutations G, L'algèbre des invariants $\mathbb{K}[\mathbf{x}]^G$ est une algèbre graduée connexe.

Le théorème suivant, fondamental, permet de construire la série de Hilbert des algèbres d'invariants sous l'action d'un groupe de matrices (et donc en particulier, sous l'action d'un groupe de permutations).

Theorème 2.4.1 (Molien 1897)**.** *Soit G un sous-groupe fini de $GL_n(\mathbb{K})$. La série de Hilbert de l'anneau des invariants $\mathbb{K}[\boldsymbol{x}]^G$ peut être obtenue via la formule*

$$H(\mathbb{K}[\boldsymbol{x}]^G, z) = \frac{1}{|G|} \sum_{M \in G} \frac{1}{det(Id - zM)}$$

Interfacées depuis `Gap`, les séries de Molien peuvent être appelées en `Sage` pour tout groupe de permutations. On remarque le côté adaptatif de la méthode `molien_series` ; peu importe la manière dont le groupe est construit, il suffit qu'il appartienne à la catégorie `Sage` des groupes de permutations.

```
sage: G = PermutationGroup([[(1,2,3,4,5)],[(1,2,4,3)]]); G
Permutation Group with generators [(1,2,3,4,5), (1,2,4,3)]
sage: G.molien_series()
(x^6 + x^3 - x + 1)/(-x^13 + 2*x^12 - 2*x^10 + 2*x^9 - x^8 - 2*x^7
                          + 2*x^6 + x^5 - 2*x^4 + 2*x^3 - 2*x + 1)
```

```
sage: S5 = SymmetricGroup(5); S5
Symmetric group of order 5! as a permutation group
sage: S5.molien_series()
1/(-x^15 + x^14 + x^13 - x^10 - x^9 - x^8 + x^7 + x^6 + x^5 - x^2 - x + 1)
sage: A3 = AlternatingGroup(3); A3
Alternating group of order 3!/2 as a permutation group
sage: A3.molien_series()
(x^2 - x + 1)/(-x^5 + 2*x^4 - x^3 + x^2 - 2*x + 1)
sage: H = TransitiveGroup(4,3); H
Transitive group number 3 of degree 4
sage: H.molien_series()
(x^2 - x + 1)/(x^8 - 2*x^7 + 2*x^5 - 2*x^4 + 2*x^3 - 2*x + 1)
```

Ce résultat très général présente des simplifications de calcul lorsqu'on l'applique au cas des groupes de permutations. En effet, Le calcul de $det(Id - zM)$ lorsque M représente la matrice d'une permutation σ ne dépend que du *type cyclique* de σ. Aussi, au lieu de sommer sur tous les éléments de G, on peut regrouper les permutations par type cyclique et ainsi utiliser le *polynôme énumérateur des cycles* du groupe G.

Proposition 2.4.2. *Soit σ une permutation, soit M la matrice représentant σ et (l_i) le type cyclique de cette permutation, on a :*

$$det(Id - zM) = \prod_i (1 - z^i)^{l_i}$$

Maintenant pour regrouper par paquets la somme sur le groupe G, on a besoin d'un représentant et du cardinal de chaque classe de conjugaison du groupe G. Notons C_j la classe de conjugaison de G dont un représentant est μ_j et dont le cardinal est d_j. On réécrit la formule de Molien comme il suit.

Corollaire 2.4.3. *Soit G un groupe de permutations. La série de Hilbert de l'algèbre des invariants $\mathbb{K}[\boldsymbol{x}]^G$ s'exprime sous la forme :*

$$H(\mathbb{K}[\boldsymbol{x}]^G, z) = \frac{1}{|G|} \sum_{C_j \subset G} \frac{d_j}{\prod (1 - z^i)^{l_i(\mu_j)}}$$

On rappelle que pour le groupe symétrique $\mathfrak{S}_n$, les classes de conjugaisons sont paramétrées par les partitions de n. La classe de conjugaison correspondant à la partition $p = (c_1, \ldots, c_k)$ contient les $\dfrac{n!}{c_1 \cdots c_k |Aut(p)|}$ permutations ayant des cycles de longueur $c_1, \ldots, c_k$. $Aut(p)$ désigne le produit de groupes symétriques permutants les cycles de mêmes tailles, son cardinal est un produit de factorielles calculé en comptant les occurrences apparaissant dans p. Le gain est considérable, car le nombre de partitions croît relativement lentement, de l'ordre de $\exp(\sqrt{(}n))$, comparé au $n!$ permutations de $\mathfrak{S}_n$.

Nous avons implanté un *polynôme énumérateur des cycles* dans `Sage` en récupérant dans `Gap` les classes de conjugaisons. Le polynôme retourné est une fonction symétrique exprimée dans la base des sommes de puissances. Pour tout groupe de permutations G, son polynôme énumérateur des cycles est ainsi défini par :

$$P := \frac{1}{|G|} \sum_{g \in G} p_{\text{cycle type}(g)}$$

```
sage: G = PermutationGroup([[(1,2,3,4,5)],[(1,2,4,3)]]); G
Permutation Group with generators [(1,2,3,4,5), (1,2,4,3)]
sage: G.cycle_index()
1/20*p[1, 1, 1, 1, 1] + 1/4*p[2, 2, 1] + 1/2*p[4, 1] + 1/5*p[5]
sage: G.cardinality()*G.cycle_index()
p[1, 1, 1, 1, 1] + 5*p[2, 2, 1] + 10*p[4, 1] + 4*p[5]
sage: S5 = SymmetricGroup(5); S5
Symmetric group of order 5! as a permutation group
sage: S5.cycle_index()
1/120*p[1, 1, 1, 1, 1] + 1/12*p[2, 1, 1, 1] + 1/8*p[2, 2, 1]
+ 1/6*p[3, 1, 1] + 1/6*p[3, 2] + 1/4*p[4, 1] + 1/5*p[5]
sage: factorial(5)*S5.cycle_index()
p[1, 1, 1, 1, 1] + 10*p[2, 1, 1, 1] + 15*p[2, 2, 1] + 20*p[3, 1, 1]
+ 20*p[3, 2] + 30*p[4, 1] + 24*p[5]
```

Le groupe symétrique de degré 5 comporte ainsi 30 permutations qui sont en fait des 4-cycles.

2.5 Système de générateurs de l'algèbre des invariants

Theorème 2.5.1 (Hilbert)**.** *Soit G un groupe fini de matrices, sous-groupe de $GL_n(\mathbb{K})$. L'algèbre $\mathbb{K}[\boldsymbol{x}]^G$ des invariants sous l'action de G a pour dimension de Krull n.*

Définition 2.5.2 (système d'invariants primaires)**.** *Soit G un sous-groupe fini de $GL_n(\mathbb{K})$. Un ensemble de polynôme $\{\Theta_1, \ldots, \Theta_n\}$ est appelé* système d'invariants primaires *lorsque chaque polynôme Θ_i est un invariant homogène sous l'action de G et lorsque la partie $\{\Theta_1, \ldots, \Theta_n\}$ est algébriquement indépendante sur $\mathbb{K}$.*

Remarque 2.5.3. *Lorsque $n!$ est inversible dans le corps $\mathbb{K}$, pour tout groupe de permutations $G \subset \mathfrak{S}_n$; les polynômes invariants sommes de puissances forment un système d'invariants primaires.*

L'ensemble

$$\{\Theta_1, \Theta_2, \ldots, \Theta_n\} = \{\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i^2, \ldots, \sum_{i=1}^{n} x_i^n\}$$

est un système d'invariants primaires pour tout sous-groupe du groupe symétrique $\mathfrak{S}_n$.

Theorème 2.5.4 (Noether)**.** *Soit G un sous-groupe fini de $GL_n(\mathbb{K})$ où $\mathbb{K}$ est un corps tel que $|G|!$ est inversible. L'algèbre $\mathbb{K}[\boldsymbol{x}]^G$ des invariants est engendrée sur $\mathbb{K}$ par au plus $\binom{|G|+n}{n}$ invariants homogènes de degré au plus $|G|$.*

Peter Fleischman montra ensuite que ce résultat d'Emmy Noether reste valable dans le cas non-modulaire, c'est à dire qu'il est suffisant que $|G|$ seulement soit inversible.

Définition 2.5.5. *On appelle* borne sur le degré *des générateurs de l'algèbre des invariants $\mathbb{K}[\boldsymbol{x}]^G$ l'entier :*

$$\beta(\mathbb{C}[\mathbf{x}]^G) := min\{d, \mathbb{C}[x]^G \text{ est engendré par des polynômes invariants de degré } \leqslant d\}$$

Il s'agit du plus grand degré apparaissant dans un ensemble minimal de générateurs. Travaillant avec les groupes de permutations, nous noterons cette borne $\beta(G)$.

A défaut de connaître $\beta(G)$ exactement, il est très utile en pratique d'en avoir une majoration fine, afin de donner un critère d'arrêt dans les algorithmes de recherche des générateurs. La borne fournie par Noether nous donne $\beta(G) \leqslant |G|$ et cette inégalité se révèle être une égalité pour les groupes cycliques. Pour les groupes «proches» du groupe symétrique (lorsque $|\mathfrak{S}_n/G|$ est petit devant $n!$, on dira que G est «proche» du groupe $\mathfrak{S}_n$.), la borne de Noether est malheureusement fort grossière. Par exemple, pour le groupe symétrique, la borne de Noether donne $n!$ alors que nous avons vu que $\beta(\mathfrak{S}_n) = n << n!$.

Pour les groupes de permutations, Garsia et Stanton ont établi une autre borne. Lorsque $G \subset \mathfrak{S}_n$, on a aussi $\beta(G) \leqslant \binom{n}{2}$. Cette borne est optimale pour les groupes alternés mais se révèle fort grossière pour les groupes de permutations G de petit cardinal $|G| << \binom{n}{2}$. Ainsi, suivant le groupe de permutations étudié, on sélectionne la limite la plus avantageuse.

Définition 2.5.6 (ensemble minimal de générateurs)**.** *Soit M un ensemble de polynômes invariants. M est un système minimal de générateurs si $\mathbb{K}[\boldsymbol{x}]^G$ est engendré par M, mais aucun sous-ensemble strict de M.*

Cette famille de générateurs n'est pas nécessairement homogène. Toutefois, la définition est suffisamment restrictive pour contraindre le nombre et les degrés de ces générateurs. Cependant, il n'existe pas de méthode générale pour connaître ces degrés *a priori*, sans calculer explicitement un système de générateurs minimal.

Proposition 2.5.7. *Soient $\{p_1, \ldots, p_k\}$ et $\{q_1, \ldots, q_l\}$ deux systèmes minimaux de générateurs. On suppose, de plus, qu'ils sont triés par degré croissant. Alors $k = l$ et pour tout i, les polynômes p_i et q_i sont de même degré.*

La borne maximale donnée par Noether permet la mise en place d'un algorithme glouton pour la recherche d'un ensemble minimal de générateurs. On initialise l'algorithme avec une famille génératrice $\langle 1 \rangle$. Ensuite, commençant par le degré 1, puis en incrémentant progressivement ce degré, on définit une base de $\mathbb{K}[\mathbf{x}]_d^G$ de l'espace des invariants du degré courant sous l'action de G. Pour chacun de ces invariants, on regarde s'il s'obtient comme un polynôme en la famille génératrice courante. Si oui, on passe au vecteur suivant, sinon, on ajoute ce polynôme à la famille génératrice courante. Une fois la borne maximale atteinte, on retourne la famille génératrice courante. Cela constitue un algorithme qui termine, mais il est inutilisable en pratique dû à sa complexité.

Mettons l'accent sur le fait qu'un tel algorithme implique de nombreux calculs dans $\mathbb{K}[\mathbf{x}]^G$; c'est dans cet espace ambiant que sont menés les calculs d'algèbre linéaire. Notamment, la difficulté principale vient de la question suivante.

Problème 2.5.8. *Soit $\{f_1, \ldots, f_p\}$ un ensemble fini d'éléments de l'algèbre des invariants $\mathbb{K}[\boldsymbol{x}]^G$ et soit f un polynôme de $\mathbb{K}[\boldsymbol{x}]$. On note $\langle f_1, \ldots, f_p \rangle_{\mathbb{K}}$ la sous algèbre de $\mathbb{K}[\boldsymbol{x}]^G$ engendrée par cette famille finie. Fournir un algorithme efficace pour répondre à la question : f appartient-il à $\langle f_1, \ldots, f_p \rangle_{\mathbb{K}}$?*

On peut aussi caractériser un ensemble de générateurs minimaux comme suit :

Proposition 2.5.9. *Soit G un sous-groupe fini de $GL_n(\mathbb{K})$ et M un ensemble d'invariants homogènes sous l'action de G, les propriétés suivantes sont équivalentes :*

(i) M est un ensemble de générateur minimal de $\mathbb{K}[\boldsymbol{x}]^G$;

(ii) Les images des éléments de M dans le quotient $\mathbb{K}[\boldsymbol{x}]^G/(\mathbb{K}[\boldsymbol{x}]^G_+)^2$ forment une base de ce dernier comme $\mathbb{K}$-espace vectoriel.

Bien que les générateurs dans M n'aient pas de caractère canonique, le nombre de générateurs de degré d dans M est donné par la dimension de la composante correspondante dans le quotient gradué $\mathbb{K}[\mathbf{x}]^G/(\mathbb{K}[\mathbf{x}]^G_+)^2$. Toutefois, il n'existe aucun algorithme pour calculer ces dimensions, voire même juste obtenir $\beta(\mathbb{K}[\mathbf{x}]^G)$ sans calculer explicitement un système de générateurs minimaux.

2.6 L'algèbre des invariants est de Cohen-Macaulay

Voici la description algébrique la plus fine que l'on connaisse de l'anneau des invariants des groupes finis de matrices. On souhaite naturellement avoir une description algébrique précise de cette algèbre comme le nombre de générateurs, leurs relations, lesquels ne comportent pas de dépendance algébrique, comment les choisir dans l'algèbre ambiante des polynômes multivariés ou encore comment définir une relative unicité de ces générateurs.

Lemme 2.6.1. *Soit G un groupe fini de matrices, sous-groupe de $GL_n(\mathbb{K})$. L'algèbre $\mathbb{K}[\boldsymbol{x}]$ des polynômes est* entière *sur l'algèbre $\mathbb{K}[\boldsymbol{x}]^G$ des polynômes invariants.*

Démonstration. Soit p un polynôme quelconque de $\mathbb{K}[\mathbf{x}]$. On considère le polynôme P en la variable t.

$$P := \prod_{\sigma \in G} (t - \sigma \cdot p)$$

On développe P par rapport à t pour écrire P sous la forme :

$$P = t^{|G|} + \alpha_1 t^{|G|-1} + \cdots + \alpha_{|G|}$$

P est un polynôme invariant sous l'action de G par construction ; ainsi, une fois développé, chaque coefficient α_i est un polynôme invariant de $\mathbb{K}[\mathbf{x}]^G$. Par construction de p, on remarque de plus que $P(p) = 0$. p est entier sur $\mathbb{K}[\mathbf{x}]^G$. □

Une description des algorithmes pour construire un système d'invariants primaires avec degré minimal est disponible dans [DK02]. Ils utilisent les bases de Gröbner, exploitant la propriété que n polynômes homogènes $\Theta_1, \ldots, \Theta_n$ forment un système d'invariants primaires si et seulement si $\mathbf{x} = 0$ est l'unique racine du système d'équations $\Theta_1(\mathbf{x}) = \cdots = \Theta_n(\mathbf{x}) = 0$.

Définition 2.6.2 (Décomposition de Hironaka)**.** *Soit $\mathbb{K}$ un corps de caractéristique nulle et G un sous groupe fini de $GL_n(\mathbb{K})$. On appelle décomposition de Hironaka de l'algèbre des invariants $\mathbb{K}[\boldsymbol{x}]^G$ une écriture structurale de cette algèbre comme il suit :*

$$\mathbb{K}[\boldsymbol{x}]^G = \bigoplus_{i=1}^{t} \eta_i \cdot \mathbb{K}[\theta_1, \ldots, \theta_m]$$

où l'ensemble $\{\theta_1, \ldots, \theta_m\}$ forme un système d'invariants primaires et les polynômes $\{\eta_1, \ldots, \eta_t\}$ sont homogènes.

En gardant les mêmes notations, et suivant la décomposition de Hironaka, la famille formée par les polynômes η_i est dénommée comme il suit :

Définition 2.6.3 (Système d'invariants secondaires)**.** *On appelle* système d'invariants secondaires *associé aux invariants primaires* $\{\Theta_1, \ldots, \Theta_n\}$ *l'ensemble des polynômes homogènes* $\{\eta_1, \ldots, \eta_t\}$.

Nous appellerons chacun des polynômes η_i un invariant secondaire.

On remarque qu'une algèbre A admettant une décomposition de Hironaka est tout d'abord un module libre de rang fini sur une algèbre polynomiale comportant un nombre fini de générateurs. Ainsi l'algèbre A est finiment engendrée. Une famille possible de générateurs est $\{\theta_1, \ldots, \theta_m, \eta_1, \ldots, \eta_t\}$; cette famille de $m+t$ générateurs n'est pas minimale dans le cas général (minimale en nombre de générateurs, minimale pour les degrés s'il y a lieu). Une telle algèbre A est aussi de dimension de Krull m, une famille maximale algébriquement indépendante est formée par les $\{\theta_1, \ldots, \theta_m\}$.

Bien que la notion soit plus fine dans toute sa généralité, les anneaux d'invariants forment des *anneaux de Cohen-Macaulay* lorsqu'ils admettent une décomposition de Hironaka.

Theorème 2.6.4. *Soit G un groupe fini de matrices, dans le cas non-modulaire ($|G|$ est un inversible du corps $\mathbb{K}$) l'algèbre $\mathbb{K}[\boldsymbol{x}]^G$ des polynômes invariants sous l'action de G est une algèbre de Cohen-Macaulay.*

Des preuves de ce dernier théorème sont disponibles dans [HE71, Sta79, Stu93].

Dans le cadre de la théorie des invariants, cette propriété d'être Cohen-Macaulay garantit l'existence d'un système d'invariants secondaires pour tout système d'invariants primaires donné. Il n'y a pas d'unicité des éléments (des invariants secondaires eux-mêmes) mais leurs degrés sont contraints par la géométrie de l'algèbre des invariants. Notons $(d_1, \ldots, d_n)$ les degrés des invariants primaires $(\Theta_1, \ldots, \Theta_n)$ et $(d'_1, \ldots, d'_t)$ les degrés des invariants secondaires $(\eta_1, \ldots, \eta_t)$. Les séries de Hilbert et la décomposition de Hironaka relient ces entiers via

$$z^{d'_1} + \cdots + z^{d'_t} = (1 - z^{d_1}) \ldots (1 - z^{d_n}) H(\mathbb{K}[\mathbf{x}]^G, z)$$

Définition 2.6.5 (Série des invariants secondaires)**.** *On appelle* série des invariants secondaires *(ou polynôme énumérateur des invariants secondaires) le polynôme*

$$S(\mathbb{K}[\boldsymbol{x}]^G, z) := z^{d'_1} + \cdots + z^{d'_t}$$

où les d'_i sont les degrés respectifs des invariants secondaires η_i.

Attention, un tel polynôme est complètement conditionné par les différents degrés des invariants primaires, aussi nous utiliserons cette notion lorsqu'il n'y aura aucune ambiguïté sur le choix du système d'invariants primaires.

Conservant les mêmes notations pour les invariants primaires et les secondaires ainsi que leurs degrés et supposant de plus que $d_1 \leqslant \cdots \leqslant d_n$ et $d'_1 \leqslant \cdots \leqslant d'_t$; on peut alors prouver que :

$$t = \frac{d_1 \ldots d_n}{|G|}, \quad d'_t = d_1 + \cdots + d_n - n - \mu, \quad \beta(\mathbb{K}[\mathbf{x}]^G) \leqslant max(d_n, d'_t),$$

où μ est le plus petit degré d'un polynôme p tel que $\sigma \cdot p = det(\sigma)p$ pour tout $\sigma \in G$ [Sta79]

De manière générale, évaluer μ n'est pas une chose évidente. Pour le groupe symétrique, c'est un résultat bien connu que le polynôme antisymétrique de degré minimal est le Vandermonde. Ainsi, pour $G = \mathfrak{S}_n$, $\mu = \binom{n}{2}$ et le plus petit sigma-invariant est

$$\Delta(x_1, \ldots, x_n) := \prod_{1 \leqslant i < j \leqslant n} (x_j - x_i)$$

Lorsque G est le groupe symétrique $\mathfrak{S}_n$, les n premières sommes de puissances forment un système d'invariants primaires, mais aussi engendrent complètement l'algèbre des invariants, ainsi $t = 1$, $d'_t = 0$ et $\eta_1 = 1$. Ce qui est consistant avec le théorème fondamental des polynômes symétriques 2.2.

Plus généralement, pour G un groupe de permutations, les sommes de puissances forment toujours un ensemble de paramètres homogènes. Ainsi, $\mathbb{K}[\mathbf{x}]^G$ est un module libre sur l'algèbre $\mathrm{Sym}(\mathbf{x}) = \mathbb{K}[\mathbf{x}]^{\mathfrak{S}_n}$ des polynômes symétriques. Il suit alors :

$$t = \frac{n!}{|G|}, \quad d'_t = \binom{n}{2} - \mu, \quad \beta(\mathbb{K}[\mathbf{x}]^G) \leqslant \binom{n}{2}$$

Remarque 2.6.6. *Soit G un groupe de permutations. Alors $\mu = 0$ (i.e. les constantes sont sigma invariantes) si et seulement si G est un sous-groupe du groupe alterné A_n. Lorsque c'est le cas, le polynôme :*

$$\sum_{orb(G)} \boldsymbol{x}^{(n-1,n-2,\ldots,2,1,0)}$$

est un invariant secondaire. Cet invariant secondaire est irréductible lorsque $G = A_n$.

Exemple 2.6.7. *Pour G un groupe de permutations cyclique. Si G est d'ordre pair, alors $\mu = 1$, et sinon $\mu = 0$.*

Démonstration. Si G est d'ordre impair, c'est un sous-groupe de A_n et $\mu = 0$. Sinon, sans perte de généralité, on peut supposer que $G = \langle (1, 2, 3, \ldots, 2n) \rangle$, et alors le polynôme suivant est sigma-invariant de degré 1 :

$$P = x_1 - x_2 + x_3 - x_4 + \cdots + x_{2n-1} - x_{2n}\,. \qquad \square$$

Deuxième partie

Génération efficace de représentants d'orbites

Génération efficace de représentants d'orbites

Un pré-requis important pour le calcul des invariants de groupes de permutations est l'énumération *efficace* des vecteurs d'entiers modulo l'action d'un groupe de permutations. C'est un problème difficile : il contient, par exemple, comme cas particulier la génération des graphes à isomorphie près [McK98, Rea78], même si les cas génériques en pratique, et donc les choix des compromis dans l'algorithmique, diffèrent quelque peu. Dans cette partie nous décrivons l'algorithmique que nous avons mise au point et implantée dans `Sage`, en nous basant sur la méthode d'énumération ordonnée [Ser03] et l'algorithmique des groupes de permutations. Cette implantation, en cours d'intégration dans la version officielle de `Sage`, trouvera rapidement d'autres applications comme pour le calcul sur les espèces combinatoires.

2.7 Action sur les vecteurs d'entiers, vecteurs canoniques

Rappelons rapidement l'action par position d'un groupe de permutations sur les n-uplets d'entiers.

Définition 2.7.1 (action sur les vecteurs d'entiers)**.** *Soit n un entier strictement positif et g un élément du groupe symétrique $\mathfrak{S}_n$. On définit l'*action de g sur un vecteur *à n composantes par :*

$$g \cdot (x_1, x_2, \ldots, x_n) := (x_{g(1)}, x_{g(2)}, \ldots, x_{g(n)})$$

`Sage` autorise à développer des fonctions bas niveau en langage `Cython`. Ce langage consiste essentiellement à compiler des fonctions écrites en un style `Python` mais avec possibilité de typage fort. Pour des fonctions basiques manipulant des types primitifs, le recours à `Cython` peut réduire jusqu'à 20 fois le temps d'exécution.

Pour implanter l'action d'une permutation sur un n-uplet, nous avons choisi `Cython`.

```
sage: S5 = SymmetricGroup(5)
sage: p = S5.an_element(); p
(1,2,3,4,5)
sage: p._act_on_list_on_position([1,2,3,4,5])
[2, 3, 4, 5, 1]
sage: p._act_on_list_on_position(['a','b','c','d','e'])
['b', 'c', 'd', 'e', 'a']
```

```
sage: p._act_on_list_on_position([1,0,1,0,0])
[0, 1, 0, 0, 1]
sage: p._act_on_list_on_position([1,1,0,0,0])
[1, 0, 0, 0, 1]
sage: p = S20.random_element()
sage: L = [randint(0,2) for i in range(20)]
sage: timeit("p._act_on_list_on_position(L)")
625 loops, best of 3: 1.07 * 10^-6 s per loop
```

Définition 2.7.2. *Soient $G \subset \mathfrak{S}_n$ un groupe de permutations et $v = (a_1, \ldots, a_n)$ un vecteur de $\mathbb{N}^n$. L'*orbite de v sous l'action de G *est l'ensemble*

$$orb_G(v) := \{g \cdot (a_1, \ldots, a_n) | g \in G\}.$$

Voici quelques exemples de calculs d'orbites.

```
sage: G = SymmetricGroup(3)
sage: I = IntegerVectorsModPermutationGroup(G)
sage: I.orbit([2,1,0])
[[2, 1, 0], [2, 0, 1], [1, 2, 0], [1, 0, 2], [0, 1, 2], [0, 2, 1]]
sage: I.orbit([1,1,0])
[[1, 1, 0], [1, 0, 1], [0, 1, 1]]
sage: I.orbit([0,0,0])
[[0, 0, 0]]
sage: G = PermutationGroup([[(1,2,3,4)]])
sage: I = IntegerVectorsModPermutationGroup(G)
sage: I.orbit([1,1,1,1])
[[1, 1, 1, 1]]
sage: I.orbit([1,0,0,0])
[[1, 0, 0, 0], [0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0]]
sage: I.orbit([3,2,1,0])
[[3, 2, 1, 0], [2, 1, 0, 3], [1, 0, 3, 2], [0, 3, 2, 1]]
```

Problème 2.7.3. *Soit n un entier strictement positif et G un sous groupe de $\mathfrak{S}_n$. Soit M l'ensemble des vecteurs d'entiers sous l'escalier (i.e. $v = (v_1, \ldots, v_n)$ est tel que $v_i \leqslant n-1$ pour tout i). Construire un algorithme pour énumérer les n-uplets de M en ne gardant que les vecteurs maximaux pour l'ordre lexicographique dans leur orbite sous l'action de G.*

Dans l'exemple suivant, nous considérons le groupe alterné A_4, et itérons parmi les représentants des orbites de vecteurs d'entiers sous l'action de A_4. Comme il y en a une infinité, nous ne donnons que les premiers. Pour cela, nous utilisons explicitement un itérateur `it` ; il s'agit d'un objet `Python`tel que l'appel de la méthode `next` renvoie à chaque fois l'objet suivant à énumérer :

```
sage: G = PermutationGroup([[(1,2,3)],[(2,3,4)]])
sage: I = IntegerVectorsModPermutationGroup(G)
sage: it = iter(I)
sage: v = it.next(); print v, " : ",I.orbit(v)
[0, 0, 0, 0]  :  [[0, 0, 0, 0]]
```

```
sage: v = it.next(); print v, " : ",I.orbit(v)
[1, 0, 0, 0]  :  [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 0, 1], [0, 0, 1, 0]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[2, 0, 0, 0]  :  [[2, 0, 0, 0], [0, 2, 0, 0], [0, 0, 0, 2], [0, 0, 2, 0]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[1, 1, 0, 0]  :  [[1, 1, 0, 0], [1, 0, 0, 1], [1, 0, 1, 0], [0, 0, 1, 1],
                  [0, 1, 1, 0], [0, 1, 0, 1]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[3, 0, 0, 0]  :  [[3, 0, 0, 0], [0, 3, 0, 0], [0, 0, 0, 3], [0, 0, 3, 0]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[2, 1, 0, 0]  :  [[2, 1, 0, 0], [2, 0, 0, 1], [2, 0, 1, 0], [1, 2, 0, 0],
                  [1, 0, 0, 2], [1, 0, 2, 0], [0, 0, 2, 1], [0, 2, 1, 0],
                  [0, 1, 0, 2], [0, 0, 1, 2], [0, 1, 2, 0], [0, 2, 0, 1]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[1, 1, 1, 0]  :  [[1, 1, 1, 0], [1, 1, 0, 1], [1, 0, 1, 1], [0, 1, 1, 1]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[4, 0, 0, 0]  :  [[4, 0, 0, 0], [0, 4, 0, 0], [0, 0, 0, 4], [0, 0, 4, 0]]
sage: v = it.next(); print v, " : ",I.orbit(v)
[3, 1, 0, 0]  :  [[3, 1, 0, 0], [3, 0, 0, 1], [3, 0, 1, 0], [1, 3, 0, 0],
                  [1, 0, 0, 3], [1, 0, 3, 0], [0, 0, 3, 1], [0, 3, 1, 0],
                  [0, 1, 0, 3], [0, 0, 1, 3], [0, 1, 3, 0], [0, 3, 0, 1]]
```

Pour traiter le problème 2.7.3, on muni $\mathbb{N}^n$ d'un ordre total, ce qui nous permettra de différencier un unique vecteur d'entiers par orbite et aussi de proposer une méthode pour l'identifier. Nous prendrons l'ordre lexicographique coordonnées par coordonnées et suivant l'ordre total naturel de $\mathbb{N}$.

Définition 2.7.4. *Soit $A = (a_1, a_2, \ldots, a_n)$ et $B = (b_1, b_2, \ldots, b_n)$ deux vecteurs d'entiers de longueur n. On dira que $A \leqslant B$ si :*

$$\begin{cases} n = 1 \text{ et } a_1 \leqslant b_1 \\ \quad ou \\ n \geqslant 2 \text{ et } \begin{cases} a_1 < b_1 \\ \quad ou \\ a_1 = b_1 \text{ et } (a_2, a_3, \ldots, a_n) \leqslant (b_2, b_3, \ldots, b_n) \end{cases} \end{cases}$$

Les n-uplets composés d'entiers constituent une structure de données relativement primitive, sans fonction supplémentaire, `Python` sait déjà comparer ces objets.

```
sage: (4,3,2,1,0) > (4,3,2,0,0)
True
sage: (4,3,2,1,0) > (4,3,2,1,1)
False
sage: (4,3,2,2,0) > (4,3,2,1,100)
True
sage: (4,3,0,2,0) > (4,3,0,1,0)
True
```

Définition 2.7.5 (Vecteur d'entier canonique)**.** *Un vecteur d'entiers naturels $(a_1, a_2, \ldots, a_n)$ est* canonique *s'il est maximal pour l'ordre lexicographique dans son orbite sous G.*

Par exemple, le vecteur $w = (0, 0, \ldots, 0)$ possédant n composantes est canonique pour tout sous groupe de $\mathfrak{S}_n$. De même, pour h un entier naturel, le vecteur $(h, h, \ldots, h)$ est lui aussi canonique pour tout groupe de permutations. Ces vecteurs sont en effet maximaux dans leur orbite, celle-ci étant réduite à un singleton pour tout sous groupe de $\mathfrak{S}_n$.

Remarque 2.7.6. *Pour deux groupes H et G tels que $H \subset G \subset \mathfrak{S}_n$, tout vecteur v canonique sous l'action de G est aussi canonique sous l'action de H.*

Démonstration. Comme $H \subset G$, pour tout v, on a $orb_H(v) \subset orb_G(v)$. Si v est canonique sous l'action de G, il est maximum dans son orbite sous G et ainsi sous H, ce qui donne le résultat. □

Exemple 2.7.7. *Pour le groupe symétrique $\mathfrak{S}_n$, les vecteurs $(a_1, \ldots, a_n)$ canoniques sont les* partitions *: $a_1 \leqslant \cdots \leqslant a_n$. Pour tout sous-groupe G de $\mathfrak{S}_n$, les partitions restent canoniques.*

Les partitions sont les vecteurs canoniques sous l'action du groupe symétrique car $\mathfrak{S}_n$ contient en particulier la permutation qui trie dans l'ordre décroissant un vecteur d'entiers donné.

Problème 2.7.8. *Soit n un entier strictement positif et G un sous groupe de $\mathfrak{S}_n$ et $(a_1, a_2, \ldots, a_n)$ un vecteur de n entiers naturels. Déterminer si ce vecteur est canonique sous l'action de G.*

Dans le cas général, le problème d'énumération n'est pas trivial et construire tous les vecteurs canoniques ne se réduit pas à contrôler la positivité d'un critère sur ces derniers. En effet, l'algorithme glouton consistant à énumérer tous les vecteurs d'entiers pour ne garder que les canoniques présente une complexité prohibitive. Pour être compétitif, il faut absolument avoir recours à un test efficace de canonicité mais aussi ne parcourir qu'un petit sous ensemble des vecteurs d'entiers ; idéalement seulement les canoniques. Nous allons ainsi recourir à une structure arborescente pour limiter les appels aux tests de canonicité dans l'esprit des algorithmiques basées sur l'«orderly generation». [Ser03]

2.8 Arbre de génération des vecteurs d'entiers

On définit un arbre dont la racine est le n-uplet : $(0, 0, \ldots, 0)$. Nous appellerons ce n-uplet la *racine de génération*. A partir de cette racine de génération, nous définissons une fonction *fils*, qui, à partir d'un n-uplet dont la somme des coordonnées est s, donne un ensemble non vide de n-uplets dont la somme est $s + 1$.

Définition 2.8.1. *Soit $(a_1, a_2, \ldots, a_n)$ un n-uplet d'entiers. Soit i la position de la dernière composante non nulle de ce vecteur d'entiers. On définit alors une application* fils *comme il suit :*

$$fils \begin{cases} \mathbb{N}^n & \longrightarrow & \mathbb{P}(\mathbb{N}^n) \\ (a_1, a_2, \ldots, a_i, 0, 0, \ldots, 0) & \longmapsto & \left\{ \begin{array}{c} (a_1, a_2, \ldots, a_i + 1, 0, 0, \ldots, 0) \\ (a_1, a_2, \ldots, a_i, 1, 0, \ldots, 0) \\ (a_1, a_2, \ldots, a_i, 0, 1, \ldots, 0) \\ \ldots \\ (a_1, a_2, \ldots, a_i, 0, 0, \ldots, 1) \end{array} \right\} \end{cases}$$

Cette fonction n'est pas accessible par défaut en `Sage`, On peut toutefois l'importer à la main. Voici quelques exemples :

```
sage: from sage.combinat.enumeration_mod_permgroup import all_children
sage: all_children([0,0,0,0,0])
[[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0],
 [0, 0, 0, 1, 0], [0, 0, 0, 0, 1]]
sage: all_children([0,0,3,0,0])
[[0, 0, 4, 0, 0], [0, 0, 3, 1, 0], [0, 0, 3, 0, 1]]
sage: all_children([1,2,3,4,6])
[[1, 2, 3, 4, 7]]
sage: all_children([1,2,3,1,0])
[[1, 2, 3, 2, 0], [1, 2, 3, 1, 1]]
```

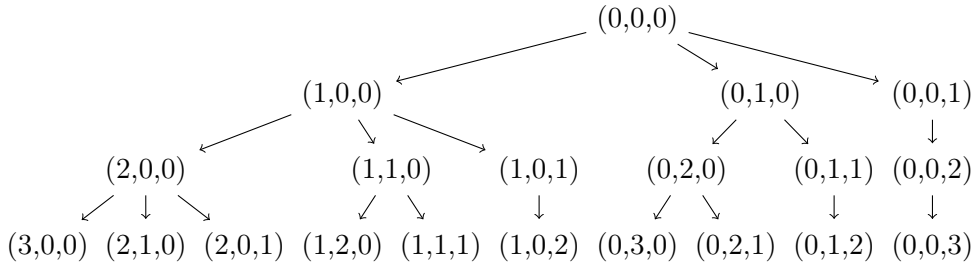
On peut construire une «opération inverse» père.

Définition 2.8.2. *Soit $(a_1, a_2, \ldots, a_n)$ un n-uplet d'entiers. Soit i la position de la dernière entrée non nulle de ce vecteur d'entiers. On définit alors une application* père *comme il suit :*

$$père \begin{cases} \mathbb{N}^n & \longrightarrow & \mathbb{N}^n \\ (0,0,\ldots,0) & \longmapsto & (0,0,\ldots,0) \\ (a_1,a_2,\ldots,a_i,0,0,\ldots,0) & \longmapsto & (a_1,a_2,\ldots,a_i-1,0,0,\ldots,0) \end{cases}$$

On remarque que, pour tout vecteur d'entiers v, le père de chacun des fils est v. Aussi, avec ces deux données, la racine et la fonction *fils*, on peut construire récursivement un arbre.

FIGURE 2.1 – Graphe de l'arbre de génération des vecteurs d'entiers de longueur 3, jusqu'à la profondeur 3



La proposition suivante permet de construire les canoniques comme sous arbre de cet arbre en coupant la plupart des branches.

Proposition 2.8.3. *Soit G un groupe de permutations, le père d'un vecteur d'entiers canonique pour G est aussi canonique. En particulier, les fils d'un vecteur d'entiers non canonique ne sont pas canoniques.*

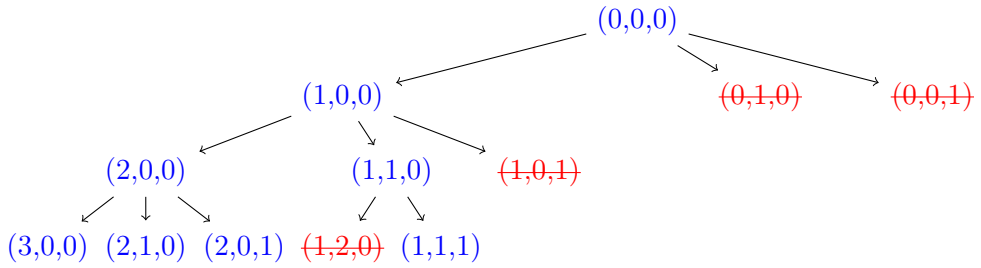
Démonstration. Prenons un couple père-fils où le père n'est pas la racine de génération. Notons le père $P = (a_1, a_2, \ldots, a_j, 0, 0, \ldots, 0)$ où $a_j \neq 0$. Nous allons supposer de plus que ce vecteur n'est pas canonique. Ainsi, il existe un élément g de G et $1 \leqslant k \leqslant n$ tel que :

$$\forall i < k : a_i = a_{g(i)} \quad et \quad a_k < a_{g(k)}$$

En fait, on a nécessairement $k < j$. En effet, si $k \geqslant j$, on aurait $\forall i : a_i \leqslant a_{g(i)}$ et $a_k < a_{g(k)}$, et donc $\sum a_i < \sum a_{g(i)} = \sum a_i$: une contradiction. Prenons $F = (b_1, \ldots, b_n)$ un fils de P. On a, pour $i < k : b_i = a_i = a_{g(i)} \leqslant b_{g(i)}$ mais aussi $b_k = a_k < a_{g(k)} \leqslant b_{g(k)}$. Donc $F < g(F)$, ainsi F n'est pas canonique. □

Appliquons cette dernière proposition à l'arbre de génération apparaissant sur la dernière figure.

FIGURE 2.2 – Arbre de génération des canoniques pour le groupe cyclique d'ordre 3 jusqu'à profondeur 3



La figure 2.2 déploie l'arbre de génération des triplets d'entiers naturels canoniques dont la somme des entrées est inférieur ou égale à 3. Chaque test de canonicité se révélant négatif nous permet de couper une branche de profondeur infinie car chaque noeud possède au moins un fils. Ces vecteurs non canoniques sont ici barrés.

Ceci permet de construire un algorithme pour résoudre le problème 2.7.3.

Algorithm 1 Énumération des vecteurs d'entiers sous l'escalier canoniques pour l'action de G

On suppose que l'on possède déjà les fonctions :

- $fils(v)$: Une fonction, qui à un vecteur d'entiers v, retourne la liste des ces fils comme définie dans 2.8.1 ;
- $EstSousEscalier(v)$: Une fonction qui retourne $True$ si et seulement si le vecteur d'entiers v est sous l'escalier.
- $EstCanonique(v, sgs(G))$: Une fonction qui retourne $True$ si et seulement si le vecteur d'entiers v est canonique sous l'action du groupe de permutations ayant $sgs(G)$ pour système de générateurs forts.

Les arguments sont ici :

- $sgs(G)$: Une liste qui contient les transversales successives des stabilisateurs partiels de G.

```
def CanoniquesSousEscalier(sgs(G)) :
    a_faire ← [(0, 0, ..., 0)]      (Départ de la racine de génération)
    liste_finale = [(0, 0, ..., 0)]
    liste_degre_precedant = [(0, 0, ..., 0)]
    for i ∈ {1, 2, ..., (n 2)} :      (Boucle sur tous les degrés possibles)
        liste_nouveau ← [ ]
        for pere ∈ liste_degre_precedant :
            enfants ← fils(pere)
            for enfant ∈ enfants :
                if EstSousEscalier(enfant) :
                    if EstCanonique(enfant, sgs(G)) :
                        liste_nouveau ← liste_nouveau + [enfant]
                        liste_finale ← liste_finale + [enfant]
        liste_degre_precedant ← liste_nouveau
    return liste_finale
```

2.9 Tests efficaces du caractère canonique

Toujours dans le but de proposer une solution algorithmique efficace au problème 2.7.3, nous allons présenter un algorithme pour le test de canonicité qui nécessite la construction d'un système de générateurs forts du groupe de permutations G. L'algorithme de Schreier–Sims utilisé pour construire cet ensemble de générateurs présente une complexité en $O(n^2 \log^3 |G| + tn \log |G|)$ pour un groupe G décrit par t générateurs. Mais le test de canonicité sera aussi appelé un grand nombre de fois et suivi par des calculs beaucoup plus lourds. Stocker une fois pour toutes un système de générateurs forts se révèle ici une stratégie efficace.

L'algorithme est basé sur la compatibilité entre système de générateurs forts construit sur une base lexicographique des positions et l'ordre lexicographique lui-même. Introduisons clairement les outils à utiliser.

Définition 2.9.1 (Ordre lexicographique partiel)**.** *Soit $A = (a_1, a_2, \ldots, a_n)$ et $B = (b_1, b_2, \ldots, b_n)$ deux vecteurs d'entiers de longueur n. Soit i un entier compris entre 1 et n. On dira que $A \leqslant_i B$ (A est partiellement lexicographiquement inférieur ou égal à B) si $(a_1, a_2, \ldots, a_i) \leqslant (b_1, b_2, \ldots, b_i)$ (pour l'ordre lexicographique usuel).*

Définition 2.9.2 (Système de générateurs forts)**.** *Soit G un groupe de permutations, sous groupe de $\mathfrak{S}_n$. Nous prendrons comme base du groupe G la liste d'entiers ordonnée $\{1, 2, \ldots, n\}$. Pour tout $0 \leqslant i \leqslant n$, soit $G_i = \{g \in G, \forall k \leqslant i : g(k) = k\}$ le sous groupe de G qui stabilise les i premières positions. On obtient ainsi une filtration du groupe G :*

$$\{e\} = G_n \subset G_{n-1} \subset G_2 \subset G_1 \subset G_0 = G$$

Pour i compris entre 1 et n, on construit T_i un ensemble de représentants dans G de l'ensemble quotient G_{i-1}/G_i. On appelle système de générateurs forts *adapté à la base lexicographique de G, la suite de transversales $\{T_1, T_2, \ldots, T_n\}$.*

Voici finalement l'algorithme pour tester la canonicité d'un vecteur d'entiers v sous l'action d'un groupe G ayant pour système de générateurs forts $sgs(G)$.

Algorithm 2 Déterminer si oui ou non, le vecteur d'entiers v est maximal dans son orbite sous l'action du groupe de permutations dont un système de générateurs forts est $sgs(G)$

Les arguments sont ici :

- v : Un vecteur d'entiers de longueur n ;
- $sgs(G)$: Une liste qui contient les transversales successives.

```
def EstCanonique(v, sgs(G)) :
    a_faire ← [v]
    {T_1, T_2, ..., T_n} ← sgs(G)
    for i ∈ {1, 2, ..., n} :        (Boucle sur la base du groupe)
        nouveaux_a_faire ← [ ]
        for v_test ∈ a_faire :
            enfants ← {g(v_test)|g ∈ T_i}
            for enfant ∈ enfants :
                if v <_i enfant :
                    return False
                else :
                    if v =_i enfant and enfant ∉ nouveaux_a_faire :
                        nouveaux_a_faire ← nouveaux_a_faire + [enfant]
        a_faire ← nouveaux_a_faire
    return True
```

Proposition 2.9.3. *Soit $G \subset \mathfrak{S}_n$ un groupe de permutations et v un vecteur d'entiers de longueur n. L'algorithme 2 exécuté avec les arguments $(v, sgs(G))$ retourne $True$ si et seulement si v est un vecteur canonique sous l'action de G.*

Démonstration. On remarque en premier lieu que cet algorithme est terminal pour toutes entrées valides. En effet, les boucles sont limitées et le cas le pire correspond à déplier entièrement le groupe G. Ultimement, l'algorithme retourne donc $True$ ou $False$. Soit v un vecteur non canonique. Il existe une permutation g telle que $g(v)$ soit le représentant canonique de l'orbite de v. La décomposition de cette permutation g suivant le système de générateurs donne :

$$\exists(g_1, \ldots, g_n) \in T_1 \times \cdots \times T_n : g = g_1 \ldots g_n$$

Comme $g(v)$ est canonique, on a

$$\forall(h_1, \ldots, h_n) \in T_1 \times \cdots \times T_n, \forall i : h_1 \ldots h_i(v) \leqslant_i g_1 \ldots g_i(v)$$

Cela a pour conséquence que la branche formée par $(g_1, \ldots, g_n)$ ne peut être coupée. Lorsque l'algorithme atteint la première position j tel que $g_1 \ldots g_j(v) > v$, l'algorithme retourne $False$.

On remarque, par ailleurs, en regardant le code, que l'algorithme retourne $False$ seulement si un permuté du vecteur v se retrouve être strictement plus grand que v. Cela ne peut arriver si v est canonique et donc, l'algorithme retourne $True$ après un parcours complet des boucles. □

Les deux stratégies sont ici l'utilisation d'un S.G.S. (Système de Générateurs Forts) et l'élimination des doublons à chaque étape. L'utilisation d'un S.G.S. construit sur la base du groupe $\{1, 2, \ldots, n\}$ est motivé par le comportement de l'ordre lexicographique et des comparaisons partielles introduites plus haut. L'élimination des doublons est une manière élémentaire d'utiliser le groupe d'automorphismes du vecteur v donné en argument.

Remarque 2.9.4. *Soit v et $sgs(G)$ deux entrées valides pour l'algorithme* EstCanonique. *Soit j tel que $1 \leqslant j \leqslant n$. Si l'algorithme est encore en cours d'exécution lors du passage pour $i = j$ dans la boucle sur la base du groupe, alors la liste de vecteurs contenue dans la variable "a_faire" contient :*

$$a\_faire = \{g_1 \dots g_{j-1}(v) | (g_1, \dots, g_{j-1}) \in T_1 \times \dots \times T_{j-1}; g_1 \dots g_{j-1}(v) =_{j-1} v\}$$

Avec l'élimination des doublons, "a_faire" peut être considéré pleinement comme un ensemble et non pas juste comme une liste informatique. On remarque aussi que "a_faire" est composé de permutés du vecteur v. Ainsi, c'est un sous ensemble de l'orbite de v sous l'action de G.

Définition 2.9.5. *Soit v et $sgs(G)$ deux arguments valides pour l'algorithme* EstCanonique. *La complexité de cet algorithme pour les arguments v et $sgs(V)$ sera donnée par le cardinal du sous ensemble de $Orb_G(v)$ déplié durant l'exécution de l'algorithme.*

On remarque que la complexité du test de canonicité pour les entrées v et $sgs(v)$ est ainsi bornée par $|orb_G(v)|$ le cardinal de l'orbite de v sous l'action de G. Cette dernière remarque est rassurante, mathématiquement parlant, mais elle ne prouve rien sur la complexité moyenne. Pour avoir une idée de la taille des calculs dans le cas général, nous avons opté pour des bancs d'essais calculant systématiquement tous les canoniques sous l'escalier.

La série de graphiques qui suit illustre des statiques sur le problème 2.7.8. Le calcul d'une complexité fine se révèle ici très difficile mais l'expérimentation permet ici de se faire une bonne idée de la taille des calculs. Tous les graphiques seront en échelles logarithmiques pour les abscisses et aussi pour les ordonnées. Les graphes suivant sont des nuages de points où chaque point représente un des groupes transitifs sur n éléments pour $n \in \{1, 2, \dots, 10\}$. Ces groupes, énumérés à isomorphie près, proviennent de la base de données du système `Gap`. La méthode consiste, pour chaque groupe, à collecter les statistiques choisies puis à placer le point correctement dans le repère.

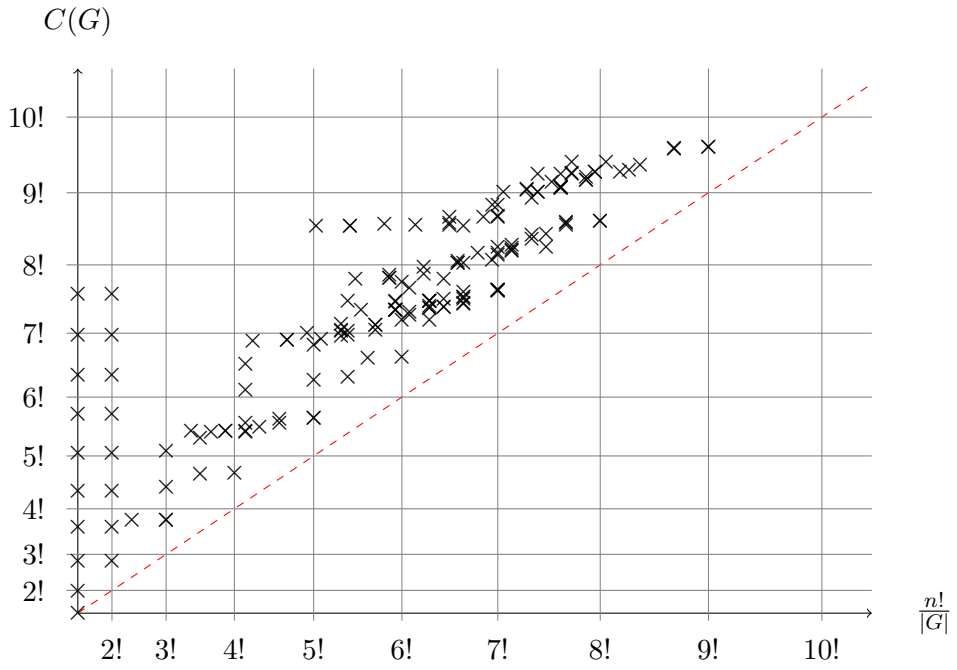


FIGURE 2.3 – Nombre de vecteurs d'entiers canoniques $C(G)$ sous l'escalier en fonction de $n!/|G|$. Le graphe comporte une croix pour chaque sous groupe transitif de $\mathfrak{S}_{10}$.

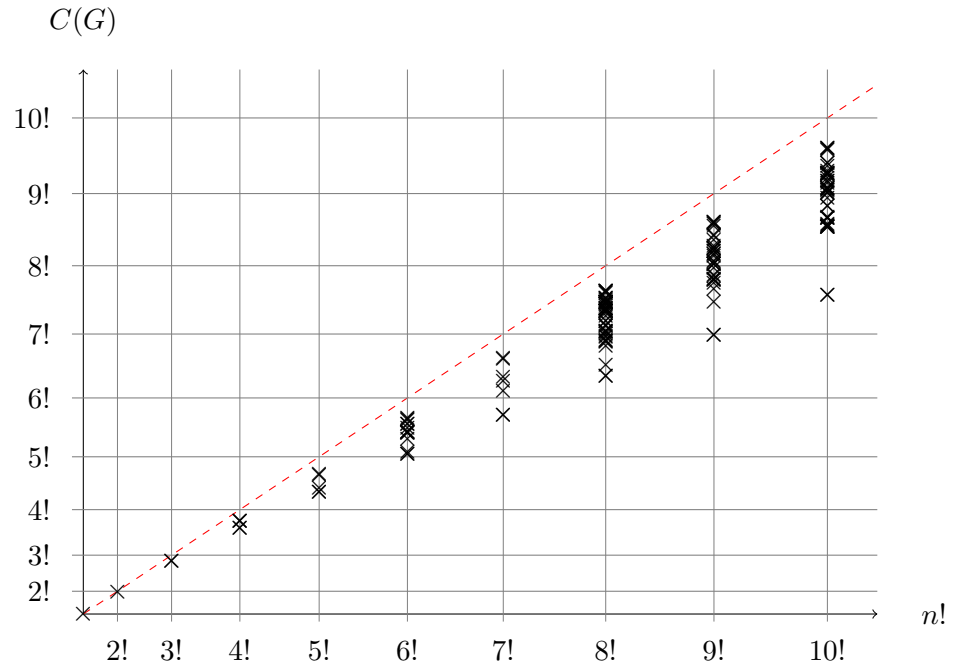


FIGURE 2.4 – Nombre de vecteurs d'entiers canoniques $C(G)$ sous l'escalier en fonction de $n!$. Le graphe comporte une croix pour chaque sous groupe transitif de $\mathfrak{S}_{10}$.

Nous verrons au chapitre 3 que les invariants secondaires associés aux polynômes symétriques sont tout d'abord des invariants sous G mais pas invariants sous $\mathfrak{S}_n$. Un bon candidat pour former un invariant secondaire est donc un polynôme G-invariant H-primitif pour H un sous

groupe strict de $\mathfrak{S}_n$. De même, ici, les canoniques pour $\mathfrak{S}_n$ sont les partitions sous l'escalier, leur cardinal est le nombre de Catalan. Définissons donc C_{raff} un nombre raffiné de canoniques, qui désignera le cardinal de l'ensemble des vecteurs d'entiers v canoniques pour l'action de G mais où v n'est pas un vecteur décroissant mais en gardant la racine de génération. Ainsi :

$$C_{\text{raff}}(G) := C(G) - C(n) + 1$$

où

$$C(n) := C(\mathfrak{S}_n) = \frac{\binom{2n}{n}}{n+1}$$

est le nombre de Catalan. C_{raff} mesure maintenant la complétion des canoniques pour G en partant des canoniques pour $\mathfrak{S}_n$.

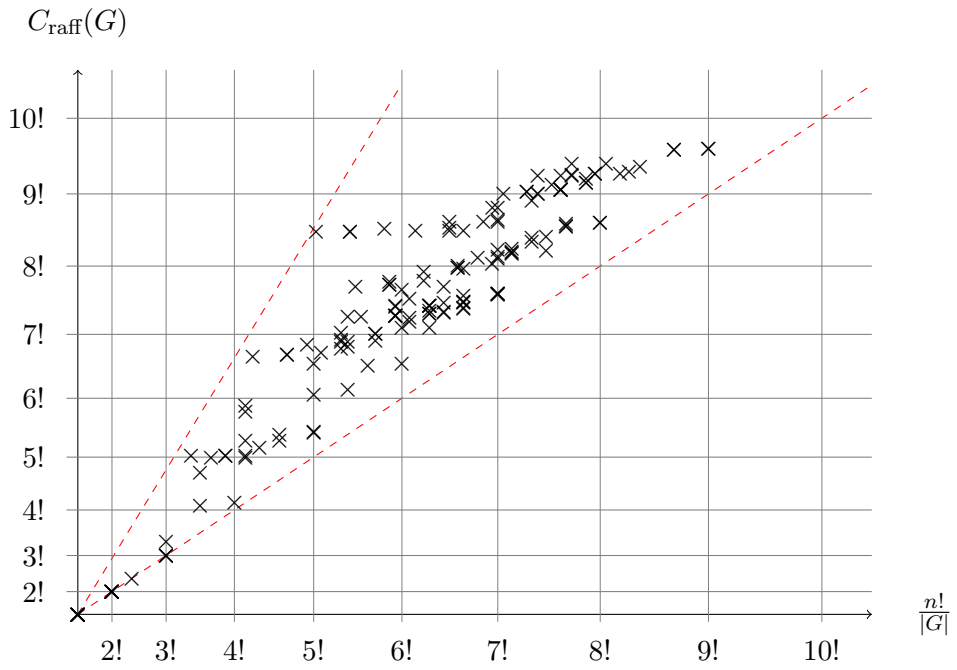


FIGURE 2.5 – Nombre $C_{\text{raff}}(G)$ de vecteurs d'entiers canoniques non décroissants sous l'escalier en fonction de $n!/|G|$. Le graphe comporte une croix pour chaque sous groupe transitif de $\mathfrak{S}_{10}$.

Comme les orbites des monômes sous l'escalier ont une orbite dont le cardinal est compris entre 1 et $|G|$, on a évidement

$$\frac{n!}{|G|} \leqslant C(G) \leqslant n!$$

Mais cet encadrement peut se révéler extrêmement large. Ce dernier graphe montre que pour les groupes praticables informatiquement, le nombre de canoniques sous l'action de G dont on a ôté les canoniques pour $\mathfrak{S}_n$ semble dominé par $(\frac{n!}{|G|})^{2,5}$.

Les monômes canoniques sous l'escalier ne formant pas une partition forment aussi l'ensemble que nous utiliserons pour en tirer les candidats à former un système d'invariants secondaires. Aussi, la complexité du calcul des secondaires via l'algorithme que nous proposerons semble aussi posséder 4.3 une complexité uniquement fonction de $n!/|G|$.

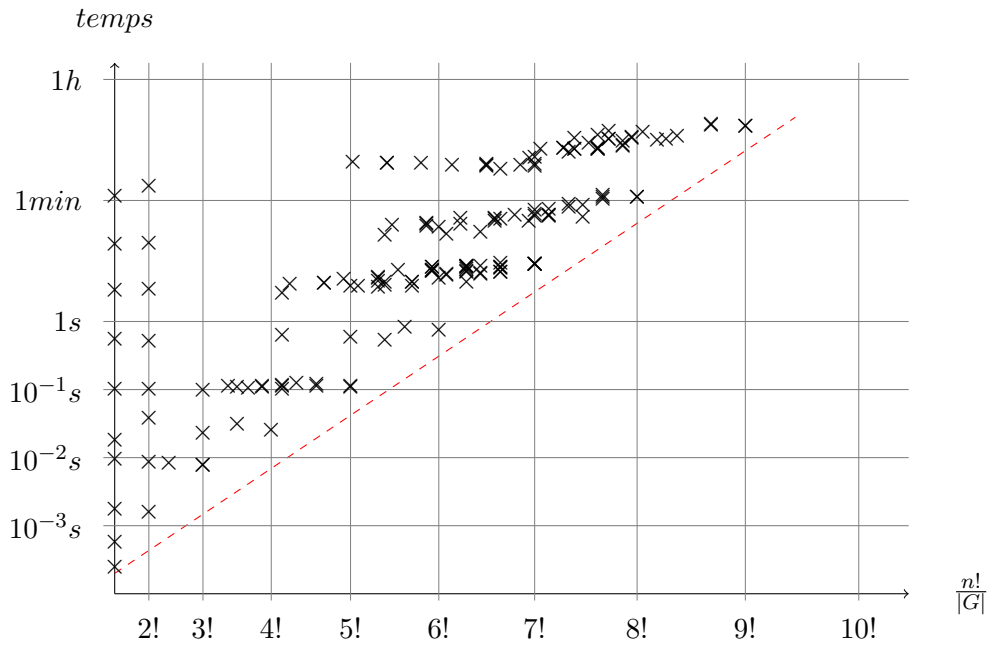


FIGURE 2.6 – Temps de calcul des canoniques pour G sous l'escalier en fonction de $\frac{n!}{|G|}$. Le graphe comporte une croix pour chaque sous groupe transitif de $\mathfrak{S}_{10}$.

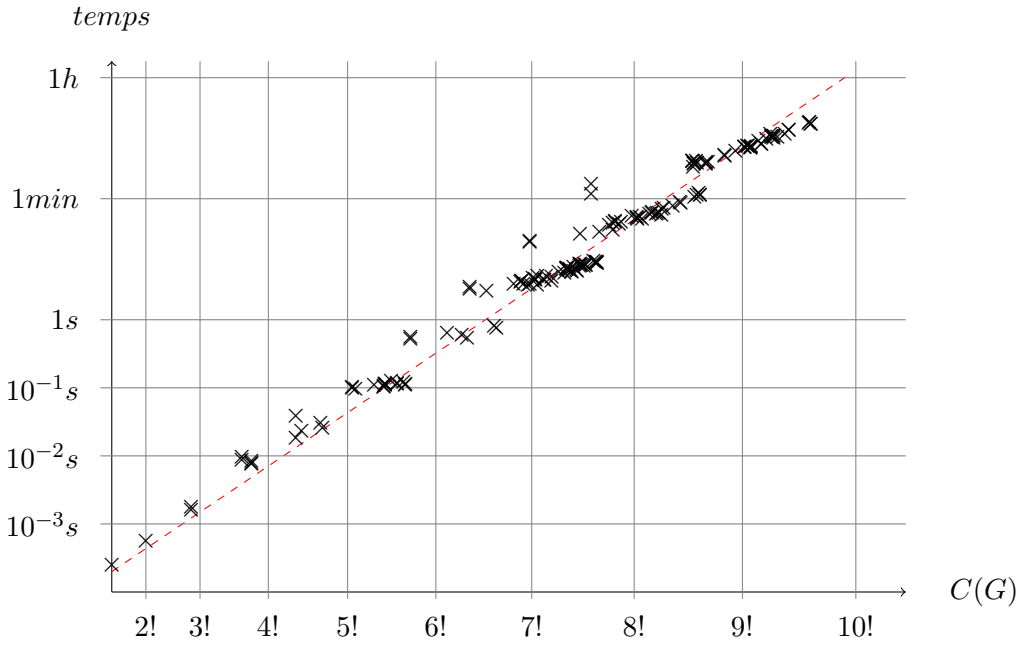


FIGURE 2.7 – Temps de calcul des canoniques en fonction du nombre $C(G)$ de vecteurs sous l'escalier canoniques pour l'action de G. Le graphe comporte une croix pour chaque sous groupe transitif de $\mathfrak{S}_{10}$.

Ce dernier graphe montre que l'algorithmique mise en oeuvre pour énumérer les canoniques sous l'action de G est en pratique de complexité grossièrement linéaire en la taille du résultat. Cette observation nous montre que le calcul des invariants ne sera pas bloqué par ce détail

technique que nous considérons maintenant résolu.

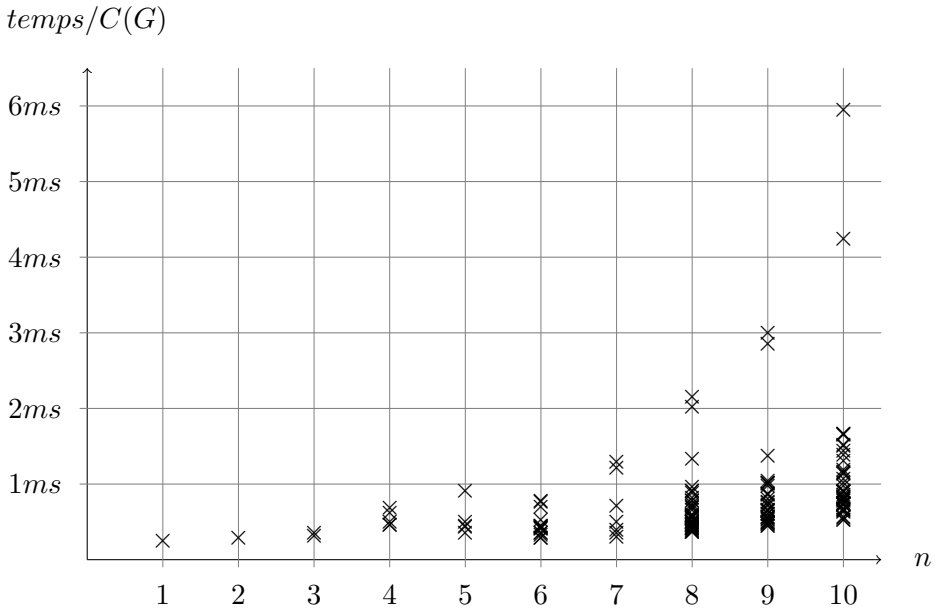


FIGURE 2.8 – Graphe du temps en secondes pour générer 1000 vecteurs en fonction du degré du groupe de permutations. Le graphe comporte une croix pour chaque groupe transitif de degré n avec $n \leqslant 10$.

Ce graphe trace l'efficacité de la méthode de génération suivant le groupe de permutations et en particulier son degré. Il permet de visualiser la vitesse à laquelle les vecteurs sont engendrés suivant n.

Troisième partie

Une approche par évaluation pour le calcul des invariants

Chapitre 3

Calcul des invariants secondaires des groupes de permutations par évaluation

Dans ce chapitre, nous allons nous focaliser sur le problème suivant. On suppose donné un système d'invariants primaires $\Theta_1, \ldots, \Theta_n$ et l'on veux construire des invariants secondaires relatifs à ces primaires. Nous proposons d'installer les calculs dans le quotient $\mathbb{K}[\mathbf{x}]^G/\langle\Theta_1, \ldots, \Theta_n\rangle_{\mathbb{K}[\mathbf{x}]^G}$ en utilisant des méthodes d'évaluation.

3.1 Enjeux et objectifs : confiner dans un «petit» quotient

Usuellement, la construction des secondaires d'un groupe fini de matrices est réduite à un problème d'algèbre linéaire en utilisant la proposition suivante.

Proposition 3.1.1. *Soit $\Theta_1, \ldots, \Theta_n$ un système d'invariants primaires et $S = (\eta_1, \ldots, \eta_t)$ une famille de polynômes homogènes avec les degrés appropriés (i.e. suivant la série des invariants secondaires). Les assertions suivantes sont équivalentes :*

(i) S est un système d'invariants secondaires ;

(ii) S est une base du quotient $\mathbb{K}[\boldsymbol{x}]^G/\langle\Theta_1, \ldots, \Theta_n\rangle_{\mathbb{K}[\boldsymbol{x}]^G}$;

(iii) S est une famille libre dans le quotient $\mathbb{K}[\boldsymbol{x}]/\langle\Theta_1, \ldots, \Theta_n\rangle_{\mathbb{K}[\boldsymbol{x}]}$.

Voici l'algorithme classique pour exhiber un système d'invariants secondaires avec sélection des irréductibles. Les calculs sont menés dans un quotient $\mathcal{Q}$ qui, selon les stratégies usuelles, est soit l'espace $\mathbb{K}[\mathbf{x}]/\langle\Theta_1, \ldots, \Theta_n\rangle$, soit l'espace $\mathbb{K}[\mathbf{x}]^G/\langle\Theta_1, \ldots, \Theta_n\rangle^G$.

Algorithm 3 Calcul des invariants secondaires et invariants secondaires irréductibles d'un groupe de permutations G correspondant au système d'invariants primaires $\Theta_1, \ldots, \Theta_n$

On suppose que ce qui suit a déjà été calculé à partir de la série de Hilbert :

- s_d : le nombre d'invariants secondaires de degré d
  (c'est le coefficient de degré d de $S(\mathbb{K}[\mathbf{x}]^G, z)$)

À chaque itération de la boucle principale, on a :

- S_d est un ensemble d'invariants secondaires de degré d ;
- I_d est un ensemble d'invariants secondaires irréductibles de degré d ;
- E_d engendre un sous espace de la composante de degré d du quotient $\mathcal{Q}$. C'est l'espace engendré par les secondaires de degré d dans le quotient ambiant choisit pour mener les calculs.

```
def SecondaryInvariants(G, (Θ_1, ..., Θ_n)) :
    for d ∈ {0, 1, 2, ..., deg(S(K[x]^G, z))} :        (Boucle principale)
        I_d = {}
        S_d = {}
        E_d = {0}_Q        (Initialisation du quotient)
        # Considération de tous les produits de degré inférieur
        for (η, η') ∈ S_k × I_l  with k + l = d :
            η̄η' = ηη'_Q        (calcul d'une forme normale)
            if η̄η' ∉ E_d :        (calcul dans le quotient)
                S_d = S_d ∪ {ηη'}
                E_d = E_d ⊕ K.Φ(ηη')
        # Complétion avec des sommes sur orbite de monôme canonique sous l'escalier
        for m ∈ CanonicalMonomialsUnderStaircaseOfDegree(d) :
            if dim E_d == s_d :
                break        (Tous les secondaires ont été trouvés)
            η̄ = OrbitSum(m)_Q        (calcul d'une forme normale)
            if η̄ ∉ E_d :        (calcul dans le quotient)
                I_d = I_d ∪ {η}
                S_d = S_d ∪ {η}
                E_d = E_d ⊕ K.Φ(η)
    return ({S_0, S_1, ...}, {I_0, I_1, ...})
```

La principale difficulté est de savoir calculer efficacement dans l'un des deux quotients $\mathbb{K}[\mathbf{x}]^G/\langle\Theta_1, \ldots, \Theta_n\rangle_{\mathbb{K}[\mathbf{x}]^G}$ ou $\mathbb{K}[\mathbf{x}]/\langle\Theta_1, \ldots, \Theta_n\rangle_{\mathbb{K}[\mathbf{x}]}$. La plupart des algorithmes utilisent des réductions sous forme normale suivant la base de Gröbner des invariants primaires $\Theta_1, \ldots, \Theta_n$, cette base ayant usuellement déjà été obtenue comme sous-produit de la construction des invariants primaires. L'obstruction principale pour être efficace est que les bases de Gröbner et les formes normales ne préservent pas les symétries ; en particulier le calcul n'est confiné que dans le quotient du (*iii*) qui est de grande dimension (donnée par le produit des degrés des invariants primaires ; c'est aussi la multiplicité de l'unique racine $\mathbf{x} = 0$ de ce système). De ce fait, les calculs deviennent difficiles en pratique même pour des tailles modérées des entrées : un logiciel comme `Singular` n'arrive à calculer les invariants secondaires que pour quelques sous-groupes de $\mathfrak{S}_8$ qui se trouvent admettre des invariants primaires de petits degrés.

Pour aller plus loin, il est nécessaire de préserver et d'exploiter les symétries. Pour G un groupe de permutations, un invariant peut être représenté de manière compacte par une combinaison linéaire de sommes sur orbite au lieu d'une combinaison de monômes. Ce premier gain permet de diminuer la complexité d'un facteur jusqu'à $|G|$ [Thi01]. De plus, on peut avoir recours

aux bases de SAGBI-Gröbner (un analogue des bases de Gröbner pour les idéaux de sous-algèbres d'algèbres polynomiales) pour mener les calculs dans le quotient du (ii) [Thi01,FR09]. Quoi qu'il en soit, les bases de Gröbner et SAGBI-Gröbner ont tendance à croître très rapidement.

Pour ces deux stratégies basées sur l'élimination, il est difficile d'établir une borne fine pour la complexité des algorithmes dû au manque de contrôle sur le comportement des calculs des bases de (SAGBI-)Gröbner [TT04].

3.2 Démarche et cas tests

3.2.1 Démarche

Rappelons que, dans les cas non singuliers, un moyen efficace de calculer modulo un idéal dans un anneau de polynômes est de procéder par évaluation sur les racines de ce dernier.

Proposition 3.2.1. *Soit P un système d'équations polynomiales dans $\mathbb{K}[\boldsymbol{x}]$ admettant un ensemble fini $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_r$ de racines simples, et soit I l'idéal de dimension 0 qu'elles engendrent. On munit $\mathbb{K}^r$ du produit point par point (produit de Hadamard). Alors, l'application d'évaluation :*

$$\begin{array}{rccc} \Phi : & \mathbb{K}[\boldsymbol{x}] & \longrightarrow & \mathbb{K}^r \\ & p & \longmapsto & (p(\boldsymbol{\rho}_1), \ldots, p(\boldsymbol{\rho}_r)) \end{array}$$

induit un isomorphisme d'algèbre lorsque restreinte à $\mathbb{K}[\boldsymbol{x}]/I$. En particulier, $\mathbb{K}[\boldsymbol{x}]/I$ est une algèbre semi-simple dont une base peut être donnée par les r idempotents $(p_i)_{i=1,\ldots,r}$ satisfaisant $p_i(\boldsymbol{\rho}_j) = \delta_{i,j}$; ces idempotents peuvent être construits par interpolation de Lagrange multivariée.

Cette proposition ne s'applique pas directement à l'idéal $\langle \Theta_1, \ldots, \Theta_n \rangle$ car ce dernier ne possède qu'une seule racine (0) avec une grande multiplicité. L'idée centrale est d'exploser cette racine multiple en autant de racines simples $\boldsymbol{\rho}_i$, en considérant un idéal perturbé tel que $\langle \Theta_1, \ldots, \Theta_n - \epsilon \rangle$, où ϵ est une constante non nulle ; ensuite, on montre que la graduation peut être utilisée pour transférer les résultats dans le quotient originel, au prix de modifications mineures et peu coûteuses.

3.2.2 Cas test : les groupes de permutations

La démarche esquissée ci-dessus est *a priori* générale, quitte à choisir un corps suffisamment gros et une perturbation appropriée de l'idéal. Cependant, le calcul des racines et leur description peut s'avérer difficile. Dans cette thèse, nous avons choisi les groupes de permutations comme cas test. Plus précisément, nous supposerons à partir de maintenant, que G est un groupe de permutations, que $\Theta_1, \ldots, \Theta_n$ sont les polynômes symétriques élémentaires $e_1, \ldots, e_n$ et $\epsilon = (-1)^{n+1}$ et que le corps $\mathbb{K}$ contient les racines n-ièmes de l'unité. Avec toutes ces hypothèses, les racines $\boldsymbol{\rho}_i$ prennent une forme élémentaire, tout en ouvrant de nombreuses connexions avec des objets classiques de la combinatoire algébrique. En filigrane, on peut espérer ainsi gagner un meilleur contrôle combinatoire des invariants, ce qui pourrait mener à terme non seulement à des algorithmes efficaces mais aussi à des résultats théoriques.

L'hypothèse sur le corps est légère, car la théorie des invariants d'un groupe dépend principalement de sa caractéristique. Le choix des invariants primaires est plus restrictif : pour les calculs

effectifs, il est préférable, chaque fois que cela est possible, de prendre des invariants primaires de degrés aussi petits que possible pour réduire le nombre d'invariants secondaires. Notons que, lorsque l'action du groupe n'est pas transitive, on peut prendre comme invariants primaires les polynômes symétriques élémentaires sur chaque orbite sur les variables ; les résultats qui suivent s'appliquent alors *mutatis mutandis*.

Finalement, nous nous somme restreints aux groupes de permutations par commodité mais nous pensons que cette classe de groupes est suffisamment vaste pour contenir tous les germes de la généralité. Par exemple, notre approche par évaluation se généralise simplement aux sous-groupes des groupes de réflexions complexes $W = G(m, p, n)$: il suffit de prendre les polynômes symétriques élémentaires en les puissances des variables comme invariants primaires ; les racines sont alors obtenues comme puissances des racines de l'unité. D'autre part, cette classe permet d'aborder des situations variées (taille du groupe, nombre de variables, etc.), et de nombreux exemples concrets apparaissant dans les applications.

3.3 Un morphisme d'évaluation

3.3.1 Points d'évaluation

Remarque 3.3.1. *Soit ρ une racine n-ième primitive de l'unité et notons $\boldsymbol{\rho} := (1, \rho, \ldots, \rho^{n-1})$. Alors, $e_1(\boldsymbol{\rho}) = \cdots = e_{n-1}(\boldsymbol{\rho}) = 0$ et $e_n(\boldsymbol{\rho}) = \epsilon$ où $\epsilon = (-1)^{n+1}$.*

Démonstration. Une racine primitive de l'unité ainsi que ces puissances définissent exactement les racines du polynôme $X^n - 1$.

$$X^n - 1 = (X - 1)(X - \rho) \ldots (X - \rho^{n-1})$$

Les coefficients étant accessibles à gauche, on écrit les relations coefficients racines du produit à droite de l'égalité.

$$\begin{aligned} X^n - 1 = &\quad X^n - (1 + \rho + \cdots + \rho^{n-1})X^{n-1} + \cdots + (-1)^n \rho \ldots \rho^{n-1} \\ X^n - 1 = &\quad X^n - e_1(\boldsymbol{\rho})X^{n-1} + \cdots + (-1)^k e_k(\boldsymbol{\rho})X^{n-k} + \cdots + (-1)^n e_n(\boldsymbol{\rho}) \end{aligned}$$

Une identification des coefficients à gauche et à droite du signe égal donne le résultat. □

Pour toute permutation σ du groupe symétrique $\mathfrak{S}_n$, introduisons $\boldsymbol{\rho}_\sigma := \sigma \cdot \boldsymbol{\rho}$ le vecteur dont σ a permuté les entrées. Les entrées de $\boldsymbol{\rho}$ étant toutes distinctes deux à deux, l'orbite $(\boldsymbol{\rho}_\sigma)_{\sigma \in \mathfrak{S}_n}$ a pour cardinal $n!$; avec la remarque précédente, il s'ensuit que cette orbite donne toutes les racines du système :

$$e_1(\mathbf{x}) = \ldots e_{n-1}(\mathbf{x}) = e_n(\mathbf{x}) - \epsilon = 0$$

Soit $\mathcal{I}$ l'idéal engendré par $e_1, \ldots, e_{n-1}, e_n - \epsilon$ dans $\mathbb{K}[\mathbf{x}]$. On définit alors un morphisme d'évaluation $\Phi : p \in \mathbb{K}[\mathbf{x}] \mapsto (p(\boldsymbol{\rho}_\sigma))_{\sigma \in \mathfrak{S}_n}$ comme dans la proposition 3.2.1 pour réaliser un isomorphisme entre $\mathbb{K}[\mathbf{x}]/\mathcal{I}$ et $\mathcal{E} = \mathbb{K}^{\mathfrak{S}_n}$.

Évidemment, les polynômes invariants sous G ont une évaluation constante sur les G-orbites. Cette simple remarque va nous permettre de confiner les calculs dans un petit sous espace de dimension $\frac{n!}{|G|}$; notons par ailleurs que cette dimension coïncide avec le nombre d'invariants secondaires à construire. Soit $\mathcal{E}^G$ la sous-algèbre formée des fonctions de $\mathcal{E}$ constantes sur les

orbites de G. $\mathcal{E}^G$ est isomorphe à $\mathbb{K}^L$ où L est une transversale à droite du quotient $\mathfrak{S}_n/G$. Soit $\mathcal{I}^G$ l'idéal engendré par $e_1, \ldots, e_{n-1}, e_n - \epsilon$ dans $\mathbb{K}[\mathbf{x}]^G$; comme la notation le suggère, il s'agit du sous ensemble des polynômes invariants dans $\mathcal{I}$.

Définition 3.3.2 (morphisme clé)**.** *Soit G un groupe de permutations, sous groupe de $\mathfrak{S}_n$, et soit L un ensemble de représentants des classes d'équivalence à droite de $\mathfrak{S}_n/G$. On définit le* morphisme clé Φ *comme il suit :*

$$\begin{array}{rccc} \Phi : & \mathbb{K}[\boldsymbol{x}] & \longrightarrow & \mathbb{K}^{\frac{n!}{|G|}} \\ & p & \longmapsto & (p(\boldsymbol{\rho}_\sigma))_{\sigma \in L} \end{array}$$

3.3.2 Propriétés

Nous nous intéressons, dans cette section, aux propriétés du morphisme Φ sur lesquelles s'appuiera l'algorithme de calcul des invariants secondaires.

Lemme 3.3.3. *La restriction de Φ à l'algèbre des invariants $\mathbb{K}[\boldsymbol{x}]^G$; donnée par :*

$$\begin{array}{rccc} \Phi : & \mathbb{K}[\boldsymbol{x}]^G & \longrightarrow & \mathcal{E}^G \\ & p & \longmapsto & (p(\boldsymbol{\rho}_\sigma))_{\sigma \in L} \end{array}$$

est surjective et induit un isomorphisme d'algèbre entre $\mathbb{K}[\boldsymbol{x}]^G/\mathcal{I}^G$ et $\mathcal{E}^G \equiv \mathbb{K}^{\frac{n!}{|G|}}$.

Démonstration. Le raisonnement est le même que pour la proposition 3.2.1. On montre la surjectivité en exhibant un antécédent pour chaque élément de la base de $\mathcal{E}^G$. Soit $\sigma \in L$, la somme sur orbite d'un interpolateur de Lagrange bien choisi réalise :

$$\Phi\Big(\sum_{orb(G)}\Big(\prod_{\substack{1 \leqslant i \leqslant n \\ j \notin \{\sigma(k)\}_{k \leqslant i}}} (x_i - \rho^{j-1})\Big)\Big) = (\delta_{\sigma,\mu})_{\mu \in L}$$

Ayant la surjectivité, la bijectivité entre $\mathcal{E}^G$ et le quotient $\mathbb{K}[\mathbf{x}]^G/\mathcal{I}^G$ se déduit en remarquant l'égalité des dimensions de ces espaces en tant que $\mathbb{K}$-espace vectoriel, c'est à dire $\frac{n!}{|G|}$. Comme morphisme d'évaluation, Φ vérifie tous les axiomes attendus d'un morphisme d'algèbre. □

Prenons tout de suite le plus petit exemple non trivial. Soit G le groupe cyclique d'ordre 3, $G = C_3 = \langle (1,2,3) \rangle$. On a alors

$$e_1 = x_1 + x_2 + x_3, \quad e_2 = x_1x_2 + x_1x_3 + x_2x_3, \quad e_3 = x_1x_2x_3.$$

Étant de cardinal 3, nous devrions chercher deux invariants secondaires. Soit j une racine primitive 3-ième de l'unité. Une transversale possible à droite de $\mathfrak{S}_3/C_3$ peut être donnée par $L = \{(), (1,2)\}$ (i.e. l'identité et la première réflexion simple de $\mathfrak{S}_3$). Les deux points d'évaluation deviennent $\boldsymbol{\rho}_{Id} = (1, j, j^2)$ et $\boldsymbol{\rho}_{(1,2)} = (j, 1, j^2)$. Évaluons maintenant le polynôme 1 et la somme sur orbite du monôme $x_1^2x_2$ (de degré 3, ici $S(\mathbb{K}[\mathbf{x}]^G, z) = 1 + z^3$). On a trivialement $\Phi(1) = (1, 1)$ et

$$\begin{array}{ll} \Phi(\sum_{orb(G)}(x_1^2x_2)) & = \Phi(x_1^2x_2 + x_2^2x_3 + x_3^2x_1) \\ & = (1^2j + j^2j^2 + (j^2)^21, j^21 + 1^2j^2 + (j^2)^2j) \\ & = (j + j + j, j^2 + j^2 + j^2) \\ \Phi(\sum_{orb(G)}(x_1^2x_2)) & = 3(j, j^2) \end{array}$$

Les deux images forment une base de $\mathbb{K}^2$. Les deux polynômes sont homogènes et respectent les degrés attendus par la série de Hilbert, ainsi il forment un système d'invariants secondaires. On a donc une description fine de l'algèbre des invariants.

$$\mathbb{K}[x_1, x_2, x_3] = \mathbb{K}[e_1, e_2, e_3] \oplus \sum_{orb(G)} (x_1^2 x_2) \cdot \mathbb{K}[e_1, e_2, e_3]$$

Remarque 3.3.4. *Pour tout polynôme P dans $\mathbb{K}[e_1, \ldots, e_n]$, l'image de P par Φ s'obtient en substituant les élémentaires par les neutres $0_{\mathcal{E}^G}$ et $1_{\mathcal{E}^G}$ de l'espace des évaluations.*

$$\Phi(P) = P(0_{\mathcal{E}^G}, \ldots, 0_{\mathcal{E}^G}, (-1)^{n+1} 1_{\mathcal{E}^G})$$

Ainsi, l'image de l'ensemble des polynômes symétriques forme une droite vectorielle dans l'espace des évaluations $\mathcal{E}$: $\Phi(\mathrm{Sym}(\mathbf{x})) = \langle (1, 1, \ldots, 1) \rangle_{\mathbb{K}}$

Le résultat suivant montre comment faire pleinement le calcul des invariants secondaires dans le quotient modulo une complication mineure.

Theorème 3.3.5. *Soit G un groupe de permutations, sous groupe de $\mathfrak{S}_n$ et $\mathbb{K}$ un corps de caractéristique 0 contenant une racine primitive n-ième de l'unité. Soit S un système d'invariants secondaires associé au système d'invariants primaires formé par $e_1, \ldots, e_n$. Notons $\langle S \rangle_{\mathbb{K}}$ l'espace vectoriel engendré par les secondaires et S_d le sous ensemble de S formé par les secondaires de degré d, alors :*

$$\begin{aligned} \textit{Pour } 0 \leqslant d < n : &\quad \Phi(\mathbb{K}[\boldsymbol{x}]_d^G) = \Phi(\langle S_d \rangle_{\mathbb{K}}) \\ \textit{Pour } d \geqslant n : &\quad \Phi(\mathbb{K}[\boldsymbol{x}]_d^G) = \Phi(\langle S_d \rangle_{\mathbb{K}}) \oplus \Phi(\mathbb{K}[\boldsymbol{x}]_{d-n}^G) \end{aligned}$$

En particulier, Φ se rétracte en un isomorphisme entre $\langle S \rangle_{\mathbb{K}}$ et $\mathcal{E}^G$.

Démonstration. On construit la décomposition de Hironaka fournie par le système S

$$\mathbb{K}[\mathbf{x}]^G = \bigoplus_{\eta \in S} \eta \mathbb{K}[e_1, \ldots, e_{n-1}, e_n]$$

Ainsi, tout polynôme P invariant homogène de degré d se décompose de manière unique :

$$P = \sum_{\eta \in S} \eta Q_\eta(e_1, \ldots, e_{n-1}, e_n), \qquad \forall \eta \in S : \deg(\eta) + \deg(Q_\eta) = d$$

où les polynômes Q_η sont tous homogènes. Lorsque l'on applique le morphisme clé Φ à P, en utilisant la remarque 3.3.4

$$\Phi(P) = \sum_{\eta \in S} \Phi(\eta) Q_\eta(0_{\mathcal{E}^G}, \ldots, 0_{\mathcal{E}^G}, (-1)^{n+1} 1_{\mathcal{E}^G})$$

Quand $d < n$, pour tout η, $\deg(Q_\eta) < n$ et donc $\Phi(Q_\eta)$ prend pour valeur son coefficient constant $a_\eta \in \mathbb{K}$. Q_η étant homogène, ce coefficient est non nul si et seulement si Q_η est de degré 0, on obtient donc

$$\Phi(P) = \sum_{\eta \in S_d} a_\eta \Phi(\eta).$$

Ce qui donne la première partie du résultat. De manière générale, comme seules les puissances pures de e_n, apparaissant avec un coefficient non nul, contribuent dans l'évaluation par Φ ; il existe un polynôme R tel que l'évaluation de P se réduise à :

$$\Phi(P) = \sum_{\eta \in S_d} \Phi(\eta) Q_\eta(0_{\mathcal{E}^G}, \ldots, 0_{\mathcal{E}^G}, (-1)^{n+1} 1_{\mathcal{E}^G}) + \Phi(e_n)\Phi(R)$$

où R est un invariant homogène de degré $d-n$. Ce qui donne une première inclusion : $\Phi(\mathbb{K}[\mathbf{x}]_d^G) \subset \Phi(\langle S_d \rangle_{\mathbb{K}}) \oplus \Phi(\mathbb{K}[\mathbf{x}]_{d-n}^G)$. L'inclusion inverse est immédiate en utilisant la multiplication par e_n dans $\mathbb{K}[\mathbf{x}]^G$. □

Dans la pratique, ce théorème ajoute deux nouvelles caractérisations des systèmes d'invariants secondaires à celles de la proposition 3.1.1.

Corollaire 3.3.6. *Prenons les polynômes symétriques élémentaires $e_1, \ldots, e_n$ pour système d'invariants primaires. Soit $S = (\eta_1, \ldots, \eta_t)$ une famille de polynômes invariants homogènes sous l'action du groupe G dont les degrés respectent la série des invariants secondaires. Les assertions suivantes sont équivalentes :*

(i) S est un système d'invariants secondaires ;

(iv) $\Phi(S)$ forme une base de $\mathcal{E}^G$;

(v) Pour tout d, les éléments de $\Phi(S_d)$ sont linéairement indépendants dans $\mathcal{E}^G$ modulo le sous espace

$$\sum_{0 \leqslant j < d,\ n \,|\, d-j} \langle \Phi(S_j) \rangle_{\mathbb{K}} .$$

Notons que, récursivement, cette dernière somme est directe.

Soit P un polynôme invariant sous l'action d'une permutation $\sigma \in \mathfrak{S}_n$, ainsi dans l'espace des évaluations, on a :

$$P(\boldsymbol{\rho}) = P(\boldsymbol{\rho}_\sigma)$$

La contraposée de cette observation permet d'identifier les dissymétries d'un polynôme lorsque l'on travaille dans l'espace des évaluations. Ainsi, si G est un groupe de permutations et P un polynôme invariant sous l'action de G tel que les coordonnées de $\Phi(P)$ sont toutes deux à deux différentes, alors P est un G-invariant primitif. Énonçons le de manière générale pour caractériser les polynômes G-invariant H-primitif.

Proposition 3.3.7. *Soit G et H deux sous groupes de $\mathfrak{S}_n$ tels que $G \subset H$. Soit L une transversale à droite de G/H. Soit P un polynôme invariant sous l'action de G, si*

$$\forall (\sigma, \tau) \in L : P(\boldsymbol{\rho}_\sigma) \neq P(\boldsymbol{\rho}_\tau)$$

Alors P est un G-invariant H-primitif.

Ainsi, l'approche par évaluation fournit une caractérisation pour identifier des invariants primitifs au vol durant le calcul des secondaires.

3.4 Algorithme

En utilisant la dernière caractérisation (v) du corollaire 3.3.6, on peut alors élaborer un nouvel algorithme de calcul des invariants secondaires. Presque semblable à l'algorithme classique 3, il intègre une correction pour compenser la déformation du quotient. Nous verrons par la suite que cette correction n'est pas trop coûteuse.

Algorithm 4 Calcul des invariants secondaires et des invariants secondaires irréductibles d'un groupe de permutations G correspondant au système d'invariants primaires constitué par les polynômes symétriques et utilisant le morphisme d'évaluation Φ.

On suppose que ce qui suit a déjà été calculé à partir de la série de Hilbert :
- s_d : le nombre d'invariants secondaires de degré d
  (c'est le coefficient de degré d de $S(\mathbb{K}[\mathbf{x}]^G, z)$)
- e_d : la dimension de $\dim \Phi(\mathbb{K}[\mathbf{x}]_d^G)$
  (c'est en fait s_d si $d < n$ et $e_{d-n} + s_d$ sinon)

À chaque itération de la boucle principale, on a :
- S_d est un ensemble d'invariants secondaires de degré d ;
- I_d est un ensemble d'invariants secondaires irréductibles de degré d ;
- E_d forme une base de $\Phi(\mathbb{K}[\mathbf{x}]_d^G)$.

```
def SecondaryInvariants(G) :
    for d ∈ {0, 1, 2, ..., deg(S(K[x]^G, z))} :
        I_d = {}
        S_d = {}
        if d ⩾ n :
            E_d = E_{d−n}        (Defect of direct sum of Theorem ??)
        else :
            E_d = {0⃗}
        # Consider all products of secondary invariants of lower degree
        for (η, η′) ∈ S_k × I_l  with k + l = d :
            if Φ(ηη′) ∉ E_d :
                S_d = S_d ∪ {ηη′}
                E_d = E_d ⊕ K.Φ(ηη′)
        # Complete with orbitsums of monomials under the staircase
        for m ∈ CanonicalMonomialsUnderStaircaseOfDegree(d) :
            if dim E_d == e_d :
                break        (All secondary invariants were found)
            η = OrbitSum(m)
            if Φ(η) ∉ E_d :
                I_d = I_d ∪ {η}
                S_d = S_d ∪ {η}
                E_d = E_d ⊕ K.Φ(η)
    return ({S_0, S_1, ...}, {I_0, I_1, ...})
```

Proposition 3.4.1. *Pour tout groupe de permutations G, l'algorithme ci-dessus 4 retourne un système d'invariants secondaires associé au système d'invariants primaires formé par $e_1, \ldots, e_n$. Parmi les secondaires, l'algorithme identifie les invariants secondaires irréductibles.*

3.5 Optimisations

3.5.1 Évaluation d'une somme sur orbite aux racines de l'unité

Comme pour la transformée de Fourrier des polynômes en une variable, et sa variante rapide, l'intérêt de l'approche par évaluation est fortement conditionné par l'efficacité du calcul d'évaluation. Le titre de cette thèse est en partie motivé par ce point. Ici, le choix des polynômes à évaluer est primordial. Comme nous avons choisi les sommes sur orbite, le problème à résoudre est :

Problème 3.5.1. *Soit $v = (\alpha_1, \ldots, \alpha_n)$ un vecteur d'entiers naturels représentant les exposants d'un monôme en n variables. Soit G un groupe de permutations, sous groupe du groupe symétrique $\mathfrak{S}_n$. Soit $L \in \mathfrak{S}_n$, une transversale à droite du quotient $\mathfrak{S}_n/G$. Soit ρ, une racine n-ième primitive de l'unité. Construire le vecteur $(P(a_i))_{a_i \in A}$ défini par les évaluations du polynôme*

$$P = \sum_{\sigma \in orbite(v)} \sigma \cdot (x_1^{\alpha_1} \ldots x_n^{\alpha_n})$$

évalué en chaque point de l'ensemble

$$A = \{(\rho^{\tau(1)}, \rho^{\tau(2)}, \ldots, \rho^{\tau(n)})\}_{\tau \in L}.$$

Nous utilisons malheureusement un algorithme glouton pour faire ce calcul. Le seul raffinement que nous nous sommes permis est de travailler le plus possible dans les entiers naturels et non avec des éléments du corps cyclotomique (il s'agit donc plus d'un raffinement informatique que mathématique ici).

Soit deux vecteurs d'entiers $v = (\alpha_1, \ldots, \alpha_n)$ et $u = (\beta_1, \ldots, \beta_n)$, on définit leur produit scalaire $\langle v, u \rangle$ comme il suit :

$$\langle v, u \rangle := \alpha_1\beta_1 + \cdots + \alpha_n\beta_n$$

Le problème 3.5.1 se reformule avec le produit scalaire des vecteurs d'entiers comme il suit :

$$V = (\sum_{\sigma \in Orbite(v)} \rho^{\langle \sigma \cdot (\alpha_1, \ldots, \alpha_n), (\tau(1), \ldots, \tau(n)) \rangle})_{\tau \in L}$$

Au final, nous avons implémenté dans `Sage` une fonction `Cython` (i.e. une sorte de `Python` compilé) qui calcule pour tout élément σ dans l'orbite et tout représentant τ des classes à droite de $\mathfrak{S}_n/G$ la somme

$$\sum_{i=1}^{n} \alpha_{\sigma(i)} \tau(i)$$

Ce calcul est mené dans $\mathbb{Z}/n\mathbb{Z}$ puis on rassemble les différentes occurrences de chaque puissance de ρ en chaque point pour finalement construire le vecteur V.

Toutefois, nous travaillons encore sur une simplification combinatoire du problème 3.5.1.

3.5.2 Raffinements et optimisations degré par degré

Pour cette discussion, qui fixe les détails dans l'algorithme de calcul des secondaires, G est un sous groupe de $\mathfrak{S}_n$, ρ est une racine primitive n-ième de l'unité et e_i est le i-ème polynôme symétrique élémentaire en n variables.

En pratique, comme le montre l'algorithme, les calculs sont menés degré par degré. On ne cherche pas brutalement à construire une base du quotient $\mathbb{K}[\mathbf{x}]^G/\langle e_1,\ldots,e_n\rangle^G$ mais soit l'on construit une base du sous espace du quotient $(\mathbb{K}[\mathbf{x}]^G/\langle e_1,\ldots,e_n\rangle^G)_d$ formée par les polynômes de degré d, soit l'on complète avec des polynômes de degré d une base du sous espace du quotient formé des polynômes de degré $n-d$.

Pour chaque degré, la première source d'invariants secondaires est le produit d'invariants de degré inférieur. Dans le cas des approches par base de (SAGBI- Gröebner), le coût des produits peut rapidement devenir très important. Si, en première approximation, un polynôme invariant sous G comporte environ $|G|$ monômes, le produit de deux de ces polynômes devrait fournir environ $|G|^2$ monômes. Avec toute l'explosion arithmétique générée par les combinaisons linéaires et la réduction de Gauss, le coût des produits devient très élevé. Aussi, pour deux polynômes P et Q sous forme normale relative à une base de Gröbner, le produit PQ n'est pas sous forme normale ; et réduire ce produit lorsqu'il comporte un grand nombre de monômes est lourd en calcul.

Pour l'approche par évaluation, les produits sont de complexité constante. C'est un produit coordonnée par coordonnée sur deux vecteurs de $\mathbb{Q}(\rho)^{\frac{n!}{|G|}}$. Sa complexité est ainsi de $\frac{n!}{|G|}$ opérations arithmétiques sur le corps $\mathbb{Q}(\rho)$ (ici, ce sont même que des multiplications).

La réduction de Gauss permet d'exhiber les relations entre générateurs vivant dans le quotient $\mathbb{K}[\mathbf{x}]^G/\langle e_1,\ldots,e_n\rangle^G$. Lorsqu'un produit de secondaires de degré inférieur se réduit dans l'espace des polynômes évalués $\mathcal{E}^G$, on enregistre ce produit comme une relation. Ainsi, lors d'un passage à un degré supérieur, si un produit de secondaires contient une relation, nous évitons la réduction de Gauss et passons au produit suivant. Le nombre de produits à calculer grandit très rapidement mais la géométrie du produit de Hadamard se comporte particulièrement bien avec la réduction de Gauss ; pour réduire un vecteur d'évaluation, on fait apparaître des zéros sur les premières coordonnées en modifiant ce dernier avec une combinaison linéaire des secondaires déjà calculés. Si $\Phi(P)$ commence par k zéros et $\Phi(Q)$ commence par l zéro, $\Phi(PQ)$ voit ses $max(k,l)$ premières coordonnées nulles, c'est autant de travail de réduction déjà effectué à ne pas recommencer. Ainsi, pour l'approche par évaluation, deux polynômes partiellement réduits ont tendance à fournir un produit réduit.

Les vecteurs canoniques de somme d fournissent les candidats pour être invariants secondaires. On les prend par ordre lexicographique et on construit progressivement les images par le morphisme d'évaluation Φ de leur somme sur orbite sous l'action de G.

Chapitre 4

Complexité et bancs d'essais

4.1 Complexité théorique

Remarque 4.1.1. *Soit $m = \boldsymbol{x}^{\alpha}$ un monôme. Évaluer ce monôme en un point $\boldsymbol{\rho}_{\sigma}$ nécessite au plus $\mathcal{O}(n)$ opérations arithmétiques dans $\mathbb{Z}$. Supposons en effet que les puissances de ρ^k ont été pré-calculées dans $\mathbb{Q}(\rho)$ et stockées pour tout k dans $0, 1, \ldots, n-1$; alors on utilise*

$$m(\boldsymbol{\rho}_{\sigma}) = \rho^{\langle \alpha, \sigma \rangle mod n},$$

où σ est écrite, dans le produit scalaire, comme permutation de $\{0, 1, \ldots, n-1\}$.

L'évaluation d'une somme sur orbite $\sum_{orb(G)} m$ en un point $\boldsymbol{\rho}_{\sigma}$ peut être obtenue en évaluant chacun des monômes, puis en additionnant progressivement dans $\mathbb{Q}(\rho)$. Cela nous donne une borne de complexité de $\mathcal{O}(n|G|)$ opérations arithmétique dans $\mathbb{Z}$ est $\mathcal{O}(|G|)$ additions dans $\mathbb{Q}(\rho)$, que nous nous autorisons à simplifier en $\mathcal{O}(|G|)$ opérations arithmétiques dans $\mathbb{Q}(\rho)$.

Proposition 4.1.2. *Il existe un algorithme résolvant le problème d'évaluation 3.5.1 en $\mathcal{O}(n!)$ opérations arithmétiques sur le corps $\mathbb{Q}(\rho)$*

Démonstration. On utilise la méthode présenté plus haut 4.1.1 en chacun des $n!/|G|$, ce qui donne au final une complexité de $(n!/|G|)\mathcal{O}(|G|) = \mathcal{O}(n!)$ opérations. □

Theorème 4.1.3. *Soit $G \subset \mathfrak{S}_n$ un groupe de permutations et ρ une racine primitive n-ième de l'unité. La complexité de l'algorithme 4 de calcul des invariants secondaires par évaluation est bornée par $O(n!^2 + \frac{n!^3}{|G|^2})$ opérations arithmétiques dans $\mathbb{Q}(\rho)$.*

Démonstration. Voici la dissection des étapes nécessaires au calcul des invariants secondaires, elles sont collectées par une analyse fine de l'algorithme 4 ainsi que ses dépendances :

(i) Calcul d'un système de générateurs forts pour G ;

(ii) Calcul d'une transversale à droite de $\mathfrak{S}_n/G$;

(iii) Calcul des classes de conjugaison de G ;

(iv) Construction explicite des monômes sous l'escalier et canoniques pour G ;

(v) Construction de la série des invariants secondaires ;

(vi) Évaluation par Φ des sommes sur orbite des monômes sous l'escalier et canoniques pour G ;

(vii) Calculs des produits $\Phi(\eta)\Phi(\eta')$ d'invariants secondaires de degrés inférieurs ;

(viii) Réduction de Gauss des vecteurs d'évaluation.

Les complexités de (i), (ii) et (iii) sont dominées par des petits polynômes en n (voir [Ser03]) et sont négligeables dans la pratique. (iv) a été suffisamment optimisé pour ne pas être observable dans le coût final de l'algorithmique (au moins négligeable devant le nombre de monômes sous l'escalier, soit $O(n!)$). (v) peut se réduire en la somme de c polynômes de degré au plus $\binom{n}{2}$ où $c \leqslant G \leqslant n!$ est le nombre de classes de conjugaison de G. Pour (vi), on utilise le fait que le nombre de monômes canoniques sous l'escalier est borné par $n!$. En utilisant la proposition 4.1.2, l'évaluation des sommes sur orbites de ces monômes coûte au plus $n!O(n)! = \mathcal{O}(n!^2)$ opérations arithmétiques sur $\mathbb{Q}(\rho)$. Un calcul grossier de (vii) contraint le calcul du produit de Hadamard de $(n!/|G|)^2$ couples de secondaires ce qui donne $\mathcal{O}(n!/|G|)^3$ opérations arithmétiques dans $\mathbb{Q}(\rho)$. En imaginant réduire les $n!$ évaluations des monômes sous l'escalier et sachant que le rang de l'espace qu'il engendre est maximal, c'est à dire $n!/|G|$, $(viii)$ se borne avec $\mathcal{O}(n!^3/|G|^2)$ opérations arithmétiques. □

En pratique, la complexité est plus fine que cela mais il apparaît très difficile d'évaluer finement nombre de paramètres. Tout d'abord, les calculs sont effectués degré par degré. La dimension visée pour chaque degré est coordonnée par les coefficients de la série des invariants secondaires et du correctif dû à la déformation du quotient 3.3.5.

Les candidats pour constituer le système d'invariants secondaires sont aussi construits de manière feignante ; on construit un monôme canonique, son orbite et son évaluation que lorsque c'est nécessaire. Il n'y a pas d'évaluation systématique mais l'algorithme recherche de nouveaux secondaires irréductibles que lorsque les produits d'invariants de degrés inférieurs ne suffisent pas à engendrer toute la composante homogène pour un degré donnée. En pratique, pour les valeurs atteignables de n (pas plus de 14 dans le cas non modulaire), l'algorithme évalue un nombre de somme sur orbite linéaire en le nombre d'invariants secondaires irréductibles qui eux-mêmes sont en nombre négligeable devant le nombre de secondaires (sauf cas extrême pour groupe alterné A_n et groupe symétrique $\mathfrak{S}_n$). Les produits points par points $\Phi(\eta)\Phi(\eta')$ sont particulièrement rapides à construire et ces derniers présentent l'avantage important de conserver le coté partiellement réduits de η et η'. En effet, si η(resp. η') commence par l(resp. l') termes nuls, le vecteur d'évaluation produit commence alors par $\max(l, l')$ coefficients nuls et c'est autant d'étapes évitées pour la réduction de Gauss. Bien que très difficilement majorables finement, les degrés des invariants irréductibles sont globalement petits (comparé au degré du secondaire maximal) et ainsi les monômes leurs correspondants possèdent une orbite de taille réduite comparée à $|G|$.

4.2 Protocole de tests

Dans cette section, nous exhibons le code source utilisé pour effectuer nos bancs d'essais. Nous expliquerons les avantages, inconvénients et limites de tels tests et nous exposerons les motivations de nos choix entre faisabilité et efficacité.

Tous les tests ont été lancés sur une machine équipée d'un processeur de type `Intel(R) Xeon(R) CPU X7460 @2.66GHz`. C'est un processeur composé de 24 coeurs cadencés chacun à

2, 66 Gigahertz. La machine comporte 128 Gigaoctets de mémoire RAM mais les plus gros de nos calculs n'ont atteint que 12 Gigaoctets de mémoire RAM (ce qui tout de même relativement important). Chacun des calculs lancés ont utilisé un seul coeur car nous n'avons pas fourni, pour le moment, d'implantation utilisant le parallélisme des tâches. Lançant les tests quatre par quatre, nous avons monopolisé 4 coeurs du processeur durant 1 mois non stop. Cette machine nommée `sage` et rattachée au cluster `sagemath.org` est partagée par les contributeurs de `Sage` en vu du développement informatique du projet et aussi de la recherche mathématique. Ces machines ont été, en partie, financées par "National Science Foundation Grant No. DMS-0821725".

4.2.1 Cas tests

Pour exécuter des tests, il faut un algorithme mais aussi une famille de données intéressantes comme arguments de chacun des tests. Le logiciel `GAP` possède une base de données des groupes de permutations transitifs sur un petit ensemble $\{1, 2, \ldots, n\}$ énumérés à isomorphisme près. Cette base de données constitue une excellente source d'arguments pour tester nos algorithmes. Pour accéder dans `Sage` à cette base de données, il suffit de l'installer via la commande `sage -i database_gap-4.4.12.p0` dans un terminal. Une fois ce paquet optionnel installé, on a accès à

```
sage: S = TransitiveGroups(5); S
Transitive Groups of degree 5
sage: G = S.an_element(); G
Transitive group number 1 of degree 5
sage: G.gens()
[(1,2,3,4,5)]
sage: H = TransitiveGroup(12,34); H
Transitive group number 34 of degree 12
sage: H.cardinality()
72
sage: H.gens()
[(1,3,5,7,9,11)(2,4,6,8,10,12), (1,8)(2,3)(4,5)(6,7)(9,12)(10,11)]
sage: TransitiveGroup(31,1)
Traceback (most recent call last):
...
NotImplementedError: Only the transitive groups of
order less than 30 are available in GAP's database
```

Ainsi, plusieurs objets deviennent accessibles comme l'ensemble de tous les groupes transitifs sur n éléments à isomorphisme près. Les groupes eux-mêmes peuvent être appelés si l'on fournit leur numéro ainsi que le nombre d'entrées sur lequel ils agissent.

```
sage: sum([TransitiveGroups(i).cardinality() for i in range(31)])
40227
```

Cette base de données comporte donc 40227 différents groupes. Notre stratégie, ensuite, fut de lancer systématiquement les algorithmes sur ces groupes en commençant par les groupes de petit ordre pour monter petit à petit.

Il existe de nombreux algorithmes pour calculer dans l'anneau des invariants ; la seule constante, pour tous ces algorithmes, est qu'ils terminent. Par contre, n'ayant aucune indication, *a priori* du temps de calcul nécessaire pour obtenir la réponse, nous avons choisi de stopper tout calcul dépassant les 24 heures d'exécution.

4.2.2 Obtenir une décomposition de Hironaka

Nous avons choisi pour test systématique le calcul d'une décomposition de Hironaka. En effet, cette dernière fournit une description fine de l'anneau des invariants. La construire est équivalent à fournir une famille d'invariants primaires et une famille d'invariants secondaires adaptée à la première.

Aussi, nous nous sommes focalisés au long de cette étude sur le cas non modulaire. Nous avons opté pour prendre les polynômes sommes de puissances comme famille d'invariants primaires. Le nombre d'invariants secondaires n'est sûrement pas optimum compte tenu de ce choix. Il ne convient donc pas de tester séparément invariants primaires et secondaires. L'approche par évaluation que nous proposons porte sur le calcul des secondaires pour une famille de primaires fixe alors que d'autres stratégies, notamment par bases de Gröbner, font rechercher une famille de primaires à faible degré pour ensuite diminuer le nombre de secondaires à rechercher. C'est plus ou moins un tout que nous ne pouvons pas diviser.

4.2.3 Protocole de test pour `Sage`

Voici la fonction écrite en `Python` pour construire les invariants secondaires associés aux invariants primaires sommes de puissances et aussi récupérer le temps de calcul.

```
def timetest_secondary_invariants_sage(G):
    r"""
    Provide some timing information about computing secondary invariants
    of the invariant ring of a permutation group in Sage.

    EXAMPLES::

        sage: timetest_secondary_invariants_sage(TransitiveGroup(4,1))
        ((4, 1), 0.00042700767517089844)
        sage: timetest_secondary_invariants_sage(TransitiveGroup(5,1))
        ((5, 1), 0.10614800453186035)
    """
    I = InvariantRingPermutationGroup(G, QQ)
    A = I.representation_as_orbit_sum()
    E = I.representation_as_evaluation()
    s = I.secondary_invariants_series()
    points = E.evaluation_points()
    perms = E._points._list_exponents_points()
    t = walltime()
    I.secondary_invariants()
    return ((G._d, G._n), walltime()-t)
```

La seule commande dont le temps d'exécution est prise en compte est la dernière qui construit les invariants secondaires. Nous avons volontairement exclu quelques calculs préliminaires car ces derniers perturbaient une juste exploitation des résultats. Les raisons sont multiples, tout d'abord, durant le début de l'année 2011, un bogue des noyaux `Linux` produisait des temps de latence pour l'interface entre `Sage` et `GAP`. Le coût des calculs demandés à `GAP`, pouvait être jusqu'à 100 fois négligeable devant le coût de conversions des objets entre les deux systèmes. Aussi, ces calculs préliminaires présentaient une réelle perturbation pour les cas de petite taille (i.e. inférieure ou égale à 5 variables) mais leurs coûts deviennent complètement négligeables lorsque le groupe est plus important (à partir de plus de 6 variables).

4.2.4 Protocole de test pour `Singular`

Voici maintenant la fonction `Python` pour obtenir le temps de calcul dans `Singular` des invariants primaires et secondaires. `Singular` est accessible depuis `Sage` mais toutes les fonctionnalités de `Singular` ne sont pas interfacées avec les objets `Sage`. Ainsi, la fonction suivante présente une sorte d'interface entre `Sage` et une partie de la librairie `Singular` «finvar». L'interface se fait via des manipulations de chaînes de caractères, ce qui n'est pas efficace en pratique. Toutefois, ici, la quantité d'informations à communiquer entre les deux systèmes est négligeable devant les calculs qui suivent.

```
singular.eval('LIB "finvar.lib";');

def timetest_secondary_invariants_singular(G):
    r"""
    Provide some timing information about computing secondary invariants
    of the invariant ring of a permutation group in Singular.

    EXAMPLES::

        sage: for i in range(1,6):
        ...       for G in TransitiveGroups(i):
        ...           print secondary_invariants_singular(G)
        ((1, 1), 0.02)
        ((2, 1), 0.02)
        ((3, 1), 0.029999999999999999)
        ((3, 2), 0.040000000000000001)
        ((4, 1), 0.040000000000000001)
        ((4, 2), 0.040000000000000001)
        ((4, 3), 0.050000000000000003)
        ((4, 4), 0.050000000000000003)
        ((4, 5), 0.059999999999999998)
        ((5, 1), 0.080000000000000002)
        ((5, 2), 0.059999999999999998)
        ((5, 3), 0.10000000000000001)
        ((5, 4), 0.16)
        ((5, 5), 0.41999999999999998)
    """
    ...
    ...  Nous avons construits les générateurs du groupe de permutations
    ...  et les avons stockés dans une liste 'alphabet_generator' via des
    ...  fonctions techniques d'interface entre Gap, Sage et Singular.
    ...
```

```
# time function of singular seems to be precise at 1/50 secondes!
t = singular.cputime()

# the primary invariants
singular.eval('list L=primary_invariants('
              +join(alphabet_generator[:nb_gens], sep=",")
              +');')

# the secondary invariants according the primary previously found
singular.eval('matrix S,IS=secondary_char0(L[1..size(L)],1);')

# it returns ((d, n), time)
# d : degree of the group
# n : number of the group in gap database
# time : time of computation for prim/sec invariants
return ((G._d, G._n), singular.cputime(t))
```

Le temps d'exécution est récupéré via la fonction `cputime` de l'interface `Sage-Singular`. Seul le calcul des invariants primaires et secondaires est pris en compte dans le résultat.

4.3 Bancs d'essais

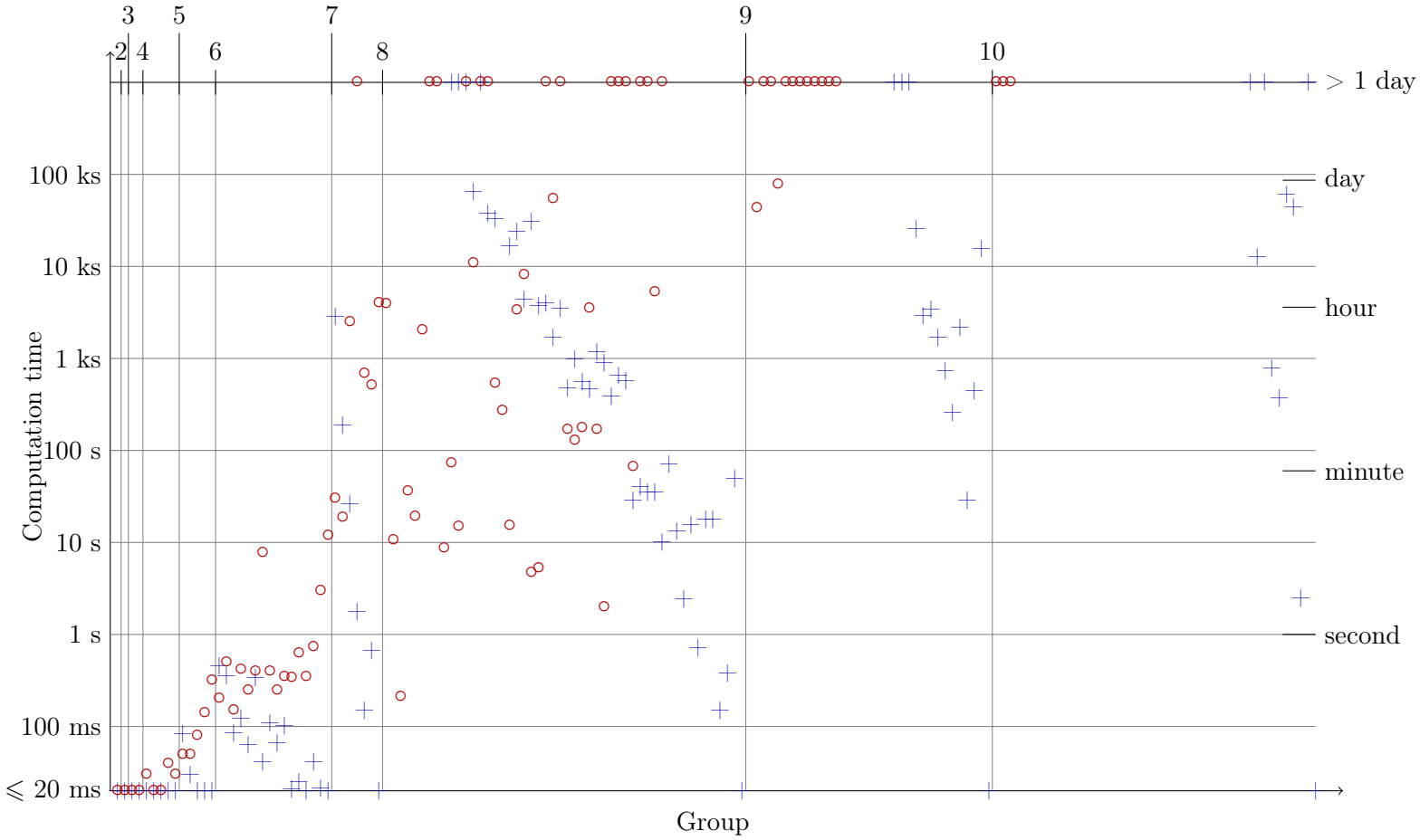


FIGURE 4.1 – Banc d'essais comparatif entre `Sage` (+) et `Singular` (○) pour obtenir une décomposition de Hironaka pour l'anneau des invariants d'un groupe de permutations. Pour chaque logiciel, on calcule un système d'invariants primaires puis un système de secondaires lui correspondant. Les primaires obtenus ne seront pas les mêmes pour les deux logiciels (en particulier pour `Sage` pour lequel les primaires sont systématiquement les polynômes symétriques), mais les algorithmes fourniront une description complète de l'anneau comme algèbre de Cohen-Macaulay. Les groupes testés sont issus de la base de données `GAP` des groupes transitifs ordonnés par degré puis par cardinalité.

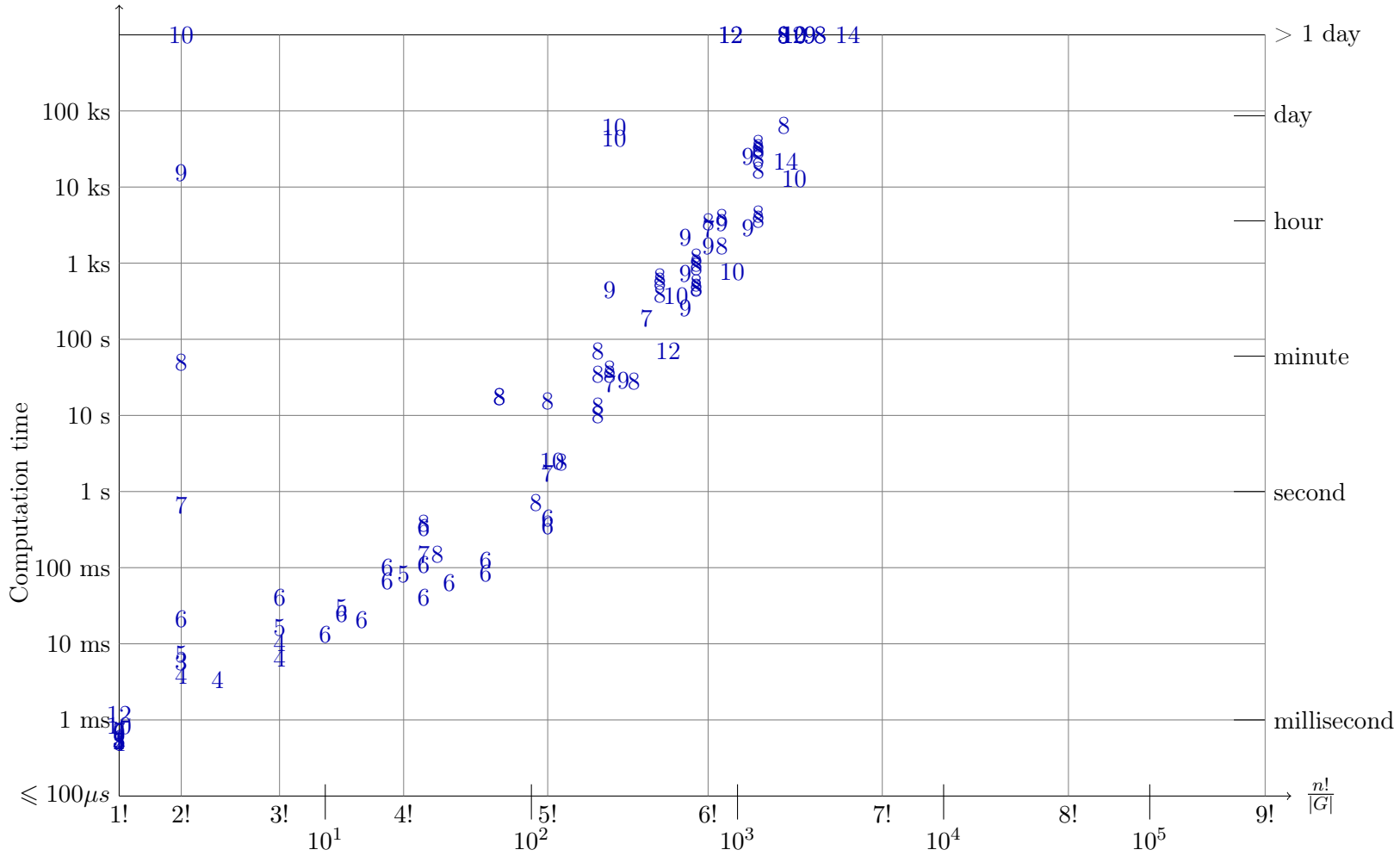


FIGURE 4.2 – Graphe de la complexité en temps pour `Sage` pour calculer un système d'invariants secondaires associé aux polynômes symétriques. Le temps est dessiné ici en fonction de nombre de points d'évaluation : $\frac{n!}{|G|}$ (ou le nombre de secondaires à calculer). Pour chaque groupe calculé (issu de la base de données du logiciel `GAP`), on a placé un n qui correspond au degré du groupe de permutations testé.

Chapitre 5

Polynômes de Schubert, candidats alternatifs pour la recherche des invariants

L'algorithme que nous avons étudié en 4 repose sur la sélection progressive d'invariants secondaires parmi une liste de candidats. Dans notre étude de complexité, nous avons pris comme candidats les sommes sur orbites sous l'escalier.

En pratique, avec l'algèbre linéaire, le goulot d'étranglement principal est le calcul de l'évaluation de ces candidats. Dans cette section, nous étudions d'autres familles potentielles de candidats dans l'espoir d'en obtenir soit de plus petites, soit avec de meilleures propriétés d'évaluation.

Les polynômes de Schubert [LS82] sont des candidats naturels car, indexés par le groupe symétrique $\mathfrak{S}_n$, ils forment une base de l'anneau des polynôme $\mathbb{K}[\mathbf{x}]$ comme module sur les polynômes symétriques $\mathrm{Sym}(\mathbf{x})$. Cette famille de polynômes définie sur deux alphabets de variables peut être définie comme un ensemble de polynômes interpolateurs en une famille de points formant une orbite sous l'action du groupe symétrique $\mathfrak{S}_n$. De ce fait, nous verrons que les images par le morphisme d'évaluation de ces polynômes sont, en partie, creuses.

Plus généralement, et sachant que c'est un problème difficile, nous avons en ligne de mire la construction combinatoire des invariants secondaires. Nous verrons comment ajouter de la combinatoire peut rigidifier le problème des invariants en espérant à terme trouver une caractérisation combinatoire d'une famille d'invariants secondaires.

Nous commençons par quelques rappels sur les polynômes de Schubert et poursuivons par une discussion sur les atouts et inconvénients des différents candidats possibles pour former un système d'invariants secondaires.

5.1 Polynômes de Schubert

Soit $\mathbb{K}$ un corps, n un entier naturel non nul ; soit $\mathbf{x}$ un alphabet de n variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ et $\mathbf{y}$ un second alphabet infini $\mathbf{y} = \{y_1, y_2, \ldots\}$. Les polynômes de Schubert

$\{Y_v : v \in \mathbb{N}^n\}$ sont des polynômes en les variables $\mathbf{x}$ dont les coefficients vivent dans l'anneau $\mathbb{K}[\mathbf{y}]$. Ils forment une base de l'anneau des polynômes en $\mathbf{x}$ qui est triangulaire sur la base des monômes.

Les polynômes de Schubert peuvent être définis par des conditions d'annulation [Las08]. Notons $|v| := v_1 + \cdots + v_n$. Il suffit alors d'établir $d-1$ points d'annulation où $d = \binom{n+|v|}{n}$ est la dimension de l'espace des polynômes en $\mathbf{x}$ de degré $\leqslant |v|$. Il ne reste plus qu'a définir une condition de normalisation pour déterminer de manière unique les polynômes de Schubert.

Ces points d'interpolation sont choisis de manière à obtenir une relation simple entre Y_v et Y_{vs_i} où s_i est une transposition simple du groupe symétrique $\mathfrak{S}_n$. En fait, ces relations se réduisent à un calcul simple dans l'anneau des polynômes en x_i, x_{i+1} vu comme module libre sur les polynômes symétriques en x_i, x_{i+1}. Sous cette considération, c'est un espace de dimension 2 ayant pour base $\{1, x_i\}$. Ces relations sont données par les différences divisées de Newton.

5.1.1 Différences divisées et définition des polynômes de Schubert

Dans toute cette première section, nous rappelons des résultats connus sur les polynômes de Schubert. Nous insistons vraiment sur leur construction en tant que polynôme interpolateur comme le décrit Alain Lascoux dans [Las08], c'est cette caractérisation qui nous intéressera par la suite lors de notre retour aux invariants.

La définition des polynômes de Schubert repose sur les différences divisées, opérateurs de dérivation discrète introduits par Newton.

Définition 5.1.1 (Différence divisée)**.** *Soit un entier $n \geqslant 2$ et un entier i tel que $1 \leqslant i < n$. On définit un opérateur sur les polynômes en n variables, appelé différence divisée comme il suit :*

$$\begin{array}{rccc} \partial_i : & (\mathbb{K}[\boldsymbol{y}])[\boldsymbol{x}] & \longrightarrow & (\mathbb{K}[\boldsymbol{y}])[\boldsymbol{x}] \\ & f & \longmapsto & \frac{f - f^{s_i}}{x_i - x_{i+1}} \end{array}$$

où s_i est la transposition élémentaire $(i, i+1)$ du groupe symétrique $\mathfrak{S}_n$ et f^{s_i} désigne le polynôme f sur lequel on a appliqué l'action de s_i.

On vérifie instantanément la consistance de cette définition en remarquant que le polynôme $f - f^{s_i}$ est antisymétrique en les variables x_i et x_{i+1}. De ce fait, il est divisible par $x_i - x_{i+1}$. On remarque aussi que le résultat de la différence divisée donne un polynôme symétrique en les variables x_i et x_{i+1}, c'est à dire que $\partial_i s_i = \partial_i$. Aussi, un polynôme f déjà symétrique en les variables x_i et x_{i+1} réalise $f = f^{s_i}$, son image par ∂_i est donc nulle. Ajoutée à la remarque précédente, cela donne pour l'opérateur $\partial_i^2 = 0$.

Proposition 5.1.2. *Les différences divisées vérifient les relations de Nil-Coxeter. Pour i, j deux entiers et un nombre de variables n suffisant, les différences divisées vérifient :*

$$\begin{array}{cc} \partial_i \partial_j = \partial_j \partial_i & \text{pour } |i-j| > 1 \\ \partial_i \partial_{i+1} \partial_i = \partial_{i+1} \partial_i \partial_{i+1} & \text{(relations de tresse)} \\ \partial_i^2 = 0 & \end{array}$$

Un tel comportement des différences divisées motive une indexation de ces dernières et de leurs produits par les permutations. Les relations de commutation et de tresse sont effectivement les mêmes que celles du groupe symétrique $\mathfrak{S}_n$.

Définition 5.1.3. *Si $a_1 \cdots a_n$ est un mot réduit pour la permutation $\omega \in \mathfrak{S}_n$, alors on définit $\partial_\omega = \partial_{a_1} \cdots \partial_{a_n}$. Le mot correspondant à l'identité est le mot vide $\partial_{Id} = Id$.*

Notons ω_0 la permutation la plus longue du groupe symétrique $\mathfrak{S}_n$. La permutation ω_0 représente le renversement complet ; pour tout i dans $\{1, \ldots, n\}$, on a $\omega_0(i) = n - i + 1$. En notation par ligne des images, $\omega_0 = (n, \ldots, 2, 1)$.

Définition 5.1.4 (Polynôme de Schubert)**.** *Soit $v \in \mathbb{N}^n$ un code de Lehmer, le* polynôme de Schubert $Y_v(\boldsymbol{x})$*, aussi noté X_σ avec $\sigma = \langle v \rangle$ la permutation correspondant au code de Lehmer v, est l'unique polynôme de $(\mathbb{K}[\boldsymbol{y}])[\boldsymbol{x}]$ tel que :*

$$Y_v(\langle u \rangle \cdot (\boldsymbol{y})) := 0, u \neq v, |u| \leqslant |v|,$$

$$Y_v(\sigma \cdot (\boldsymbol{y})) := \Delta(\sigma) := \prod_{i<j, \sigma(i)>\sigma(j)} (y_{\sigma(i)} - y_{\sigma(j)})$$

On remarque qu'il y a autant de contraintes dans la définition du polynôme de Schubert indexé par le code v que la dimension de l'espace des polynôme $\mathbb{K}[\mathbf{y}][\mathbf{x}]_{\leqslant |v|}$ de degré inférieur ou égal à $|v|$ en le premier alphabet de variables (et en tant que $\mathbb{K}[\mathbf{y}]$ espace vectoriel).

Pour un *code dominant* $v = (v_1, \ldots, v_n) \in \mathbb{N}^n$, c'est à dire que v forme une partition : $v_1 \geqslant \cdots \geqslant v_n$; on connaît une expression simple du polynôme de Schubert Y_v :

Proposition 5.1.5. *Soit $v = (v_1, \ldots, v_n) \in \mathbb{N}^n$ un code dominant. Alors :*

$$Y_v(x_1, \ldots, x_n) = \prod_{i=1}^{n} \prod_{j=1}^{v_i} (x_i - y_j)$$

Pour tout n, la permutation ω_0, la plus longue de $\mathfrak{S}_n$ a pour code de Lehmer le vecteur d'entiers $(n-1, n-2, \ldots, 1, 0) \in \mathbb{N}^n$. On peut ainsi construire le polynôme de Schubert correspondant :

$$Y_{\omega_0}(x_1, \ldots, x_n) = \prod_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant n-i}} (x_i - y_j)$$

Pour obtenir tous les polynômes de Schubert explicitement, on utilise les différences divisées comme il suit

Proposition 5.1.6. *Soit σ une permutation du groupe symétrique $\mathfrak{S}_n$ telle que la transposition s_i est une descente à droite de σ. Soit X_σ le polynôme de Schubert indexé par σ, alors :*

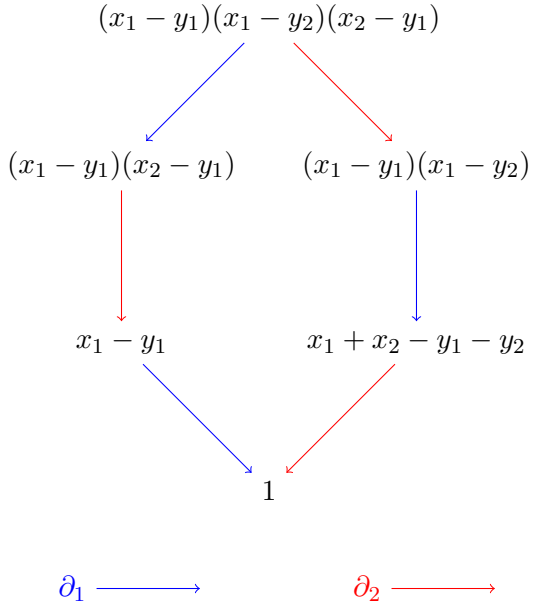
$$X_{\sigma s_i} = \partial_i(X_\sigma) = \partial_{s_i}(X_\sigma)$$

Plus généralement, pour σ une permutation du groupe symétrique $\mathfrak{S}_n$, ω_0 désignant la permutation la plus longue de $\mathfrak{S}_n$, on a :

$$X_\sigma = \partial_{\sigma^{-1}\omega_0}(X_{\omega_0})$$

Regardons cela sur un petit exemple : Le groupe symétrique $\mathfrak{S}_3$. Le jeu consiste à partir de la permutation la plus longue pour laquelle le polynôme de Schubert est $(x_1 - y_1)(x_1 - y_2)(x_2 - y_1)$ et ensuite on applique les différences divisées (ici ∂_1 et ∂_2) de manière à déplier un graphe isomorphe au permutoèdre dont les noeuds sont libellés par les polynômes de Schubert.

FIGURE 5.1 – Polynômes de Schubert du groupe $\mathfrak{S}_3$



Comme annoncé précédemment, le résultat suivant est fondamental.

Theorème 5.1.7. *L'ensemble $\mathcal{B}$ formé de polynômes de Schubert :*

$$\mathcal{B} = \{X_\sigma(\boldsymbol{x})\}_{\sigma \in \mathfrak{S}_n}$$

forme une base de l'anneau $(\mathbb{K}[\boldsymbol{y}])[\boldsymbol{x}]$ en tant que module sur l'anneau des polynômes symétriques en $\boldsymbol{x}$ à coefficients dans $\mathbb{K}[\boldsymbol{y}]$.
Lorsque $\boldsymbol{y} = 0 \in \mathbb{K}$ (i.e. $0 = y_1 = y_2 = \dots$), $\mathcal{B}$ forme une base homogène des polynômes de l'anneau $\mathbb{K}[\boldsymbol{x}]$ en tant que $\mathrm{Sym}(\mathbf{x})$-module.

Ayant une base en tant que module de l'algèbre $(\mathbb{K}[\mathbf{y}])[\mathbf{x}]$, on finit de décrire l'algèbre en regardant le comportement du produit sur la base nouvellement exhibée. Dans les faits, il apparaît que calculer un produit est chose difficile et coûteuse en calcul ; on a alors recours à la règle de Monk.

Proposition 5.1.8 (Règle de Monk)**.** *Soit un code de Lehmer $v \in \mathbb{N}^n$ et sa permutation $\sigma = \langle v \rangle$ correspondante. Soit $i \in \{1, \dots, n\}$, alors :*

$$(x_i - y_{\sigma_i}) X_\sigma = \sum_{j>i} X_{\sigma\tau_{i,j}} - \sum_{j<i} X_{\sigma\tau_{i,j}},$$

Les deux sommes sont sur toutes les transpositions $\tau_{i,j}$ telles que $l(\sigma\tau_{i,j}) = l(\sigma) + 1$

La règle de Monk ne donne pas directement l'expression d'un produit $X_\sigma X_\tau$ pour σ et τ deux permutations. Ainsi, son utilisation exige de construire un des deux polynômes dans la base des monômes et ensuite de diviser le problème pour ne faire que des multiplications simples par une des variables $x_1, \ldots, x_n$.

5.1.2 Polynômes de Schubert et invariants de groupes de permutations

Revenons aux invariants et regardons comment utiliser, si possible, les polynômes de Schubert pour la recherche d'un système d'invariants secondaires associé aux polynômes symétriques.

Action du groupe symétrique sur les polynômes de Schubert

Pour tout groupe de permutations, l'opérateur de Reynolds étant un morphisme de $\mathrm{Sym}(\mathbf{x})$-module, la famille $\mathcal{B}$ du théorème 5.1.7 peut être utilisée pour la construction d'un système d'invariants secondaires. Toutefois, cela exige la compréhension de l'action du groupe symétrique sur les polynômes de Schubert. Ce problème est en fait relativement difficile, aussi difficile que celui de la multiplication entre deux Schubert.

On rappelle que :

$$\partial_i(f) = \frac{f - f^{s_i}}{x_i - x_{i+1}}$$

Ainsi pour un polynôme de Schubert X_σ, où σ est une permutation pour laquelle la transposition s_i une descente à droite , on a :

$$X_{\sigma s_i} = \frac{X_\sigma - s_i \cdot X_\sigma}{x_i - x_{i+1}}$$

et donc :

$$s_i \cdot X_\sigma = X_\sigma + X_{\sigma s_i}(x_{i+1} - x_i)$$

(Lorsque s_i n'est pas une descente à droite, X_{sigma} est alors invariant en x_i et x_{i+1}, et alors $s_i \cdot X_\sigma = X_\sigma$).

Ainsi, deux écritures de la règle de Monk vont fournir l'action de la transposition simple s_i sur le polynôme de Schubert X_σ. On remarque même qu'il y a une équivalence de complexité entre une multiplication d'un polynôme de Schubert avec un polynôme de degré d et l'action d'une permutation de longueur d sur un Schubert. Ainsi, calculer les invariants avec les polynômes de Schubert dont le second alphabet $\mathbf{y}$ est spécialisé en 0 présente un coût supplémentaire pour la construction des polynômes de Schubert et le calcul de l'opérateur de Reynolds est lui aussi important dans cette nouvelle base.

Utilisation du caractère creux de l'évaluation des Schubert

La proposition suivante fournit, pour tout groupe de permutations, une base explicite du quotient $\mathbb{K}[\mathbf{x}]^G/\langle e_1, \ldots, e_n - \epsilon\rangle^G$. Ici, contrairement aux sommes sur orbite des interpolateurs de Lagrange, ces polynômes ne sont pas tous de degré $\binom{n}{2}$. Un seul de ces séparateurs atteint ce degré maximal et les autres sont tous de degré strictement plus petit.

Proposition 5.1.9. *Soit G un groupe de permutations, sous groupe de $\mathfrak{S}_n$ et ρ une racine primitive n-ième de l'unité. Soit $T = \{t_1, \dots, t_n\}$ une transversale à droite du quotient $\mathfrak{S}_n/G$. Prenons pour représentants les permutations pour lesquels leur code de Lehmer est maximal dans la classe d'équivalence :*

$$\forall i, 1 \leqslant i \leqslant n : t'_i = \langle \max_{\sigma \in G} \{\mathcal{C}(t_i \sigma)\} \rangle, \quad T' = \{t'_1, \dots, t'_n\}$$

Alors la famille de polynômes suivante, sommes sur orbite de Schubert,

$$\{ \sum_{orb(G)} X_{t'_i}(x_1, \dots, x_n, 1, \rho, \dots, \rho^{n-1}) \}_{t'_i \in T'}$$

forme une base du quotient $\mathbb{K}[\boldsymbol{x}]^G / \langle e_1, \dots, e_n - \epsilon \rangle^G$.

Démonstration. En fait l'image de cette base par le morphisme d'évaluation Φ est triangulaire dans l'espace des évaluations $\mathcal{E}^G$ modulo un possible réordonnement des colonnes. À permutation près de ces éléments, supposons que T' est trié par ordre croissant pour l'ordre *deglex* (le degré prime puis à même degré, c'est l'ordre lexicographique qui prend le relais) des codes de Lehmer. Supposons que la transversale L définissant Φ peut aussi s'écrire $L = \{t'_1, \dots, t'_n\}$ (c'est toujours le cas modulo un réordonnement dans la transversale). Par définition des polynômes de Schubert, on a alors :

$$\forall t'_j, \mathcal{C}(t'_j) <_{deglex} \mathcal{C}(t'_i) : X_{t'_i}(t'_j \cdot (1, \rho, \dots, \rho^{n-1}), 1, \rho, \dots, \rho^{n-1}) = 0$$

$$X_{t'_i}(t'_i \cdot (1, \rho, \dots, \rho^{n-1}), 1, \rho, \dots, \rho^{n-1}) = \prod_{k<l, t'_i(k) > t'_i(l)} (y_{t'_i(k)} - y_{t'_i(l)}) = \Delta(t'_i) \neq 0$$

Ces contraintes sur les évaluations étant posées, en passant sur les sommes sur orbites, on a alors :

$$\forall i : \Phi(\sum_{orb(G)} X_{t'_i}(\mathbf{x}, 1, \rho, \dots, \rho^{n-1})) = (0, \dots, 0, \Delta(t'_i), *, \dots, *)$$

□

Malheureusement, ces polynômes ont globalement des degrés élevés et ne sont pas homogènes. En extraire une famille d'invariants secondaires n'est donc pas une tâche facile. Toutefois, cette famille est déterminée par des arguments combinatoires sans aucun recours à l'algèbre linéaire.

Propriété de factorisation

Toujours dans le même registre, Vincent Prosper a mené des investigations sur les évaluations des polynômes de Schubert en des alphabets de la forme $\frac{1}{1-q}$ [Pro00]. Ce qui signifie que l'on va spécialiser les deux alphabets de variables $(x_1, x_2, x_3, \dots)$ et $(y_1, y_2, y_3, \dots)$ en $(1, q, q^2, \dots)$.

Définition 5.1.10 (Permutation vexillaire)**.** *Une permutation σ du groupe symétrique $\mathfrak{S}_n$ est dite vexillaire lorsqu'il n'existe pas de quadruplet d'entiers $i < j < k < l$ tel que $\sigma(j) < \sigma(i) < \sigma(l) < \sigma(k)$*

C'est une contrainte sur l'imbrication des supports deux à deux des descentes de la permutation. Dans $\mathfrak{S}_4$, la seule permutation non vexillaire est la permutation $\sigma = (2, 1, 4, 3)$.

Theorème 5.1.11 (Prosper 1999). *Soit σ un permutation vexillaire, $\mathcal{C}(\sigma)$ son code de Lehmer et $\mathcal{J}(\sigma)$ le réordonnement décroissant du code $\mathcal{C}(\sigma)$, alors :*

$$X_\sigma\left(\frac{z}{1-q},\frac{1}{1-q}\right)=X_\sigma\left(0,\frac{1}{1-q}\right)Y_{\mathcal{J}(\sigma)}\left(\frac{z}{1-q},\frac{1}{1-q}\right)Y_{\mathcal{J}(\sigma)}\left(0,\frac{1}{1-q}\right)^{-1}$$

Soit ρ une racine primitive n-ième de l'unité. L'alphabet $(1,\rho,\dots,\rho^{n-1})$ est une spécialisation de l'alphabet $(1,q,q^2,\dots)$ (c'est en fait le même avec l'adjonction de la relation $q^n=1$). Ainsi, notant $\boldsymbol{\rho}$ l'alphabet $(1,\rho,\dots,\rho^{n-1})$, le résultat de Prosper s'adapte aux racines de l'unité et donne :

$$X_\sigma\left((z,z\rho,\dots,z\rho^{n-1}),\boldsymbol{\rho}\right)=X_\sigma\left(0,\boldsymbol{\rho}\right)Y_{\mathcal{J}(\sigma)}\left((z,z\rho,\dots,z\rho^{n-1}),\boldsymbol{\rho}\right)Y_{\mathcal{J}(\sigma)}\left(0,\boldsymbol{\rho}\right)^{-1}$$

Posons $Z(\sigma)$ la constante de $\mathbb{K}$ définie par $Z(\sigma):=X_\sigma\left(0,\boldsymbol{\rho}\right)Y_{\mathcal{J}(\sigma)}\left(0,\boldsymbol{\rho}\right)^{-1}$. Lorsque z parcourt l'ensemble $1,\rho,\dots,\rho$, le premier alphabet parcourt les points de l'ensemble $\{\sigma\cdot\boldsymbol{\rho}\}_{\sigma\in C_n}$ où C_n est le sous groupe cyclique de $\mathfrak{S}_n$ engendré par le n cycle $(1,2,\dots,n)$. Pour σ une permutation vexillaire :

$$(X_\sigma(\tau\cdot\boldsymbol{\rho},\boldsymbol{\rho}))_{\tau\in C_n}=Z(\sigma)(Y_{\mathcal{J}(\sigma)}(\tau\cdot\boldsymbol{\rho},\boldsymbol{\rho}))_{\tau\in C_n}$$

Comme $\mathcal{J}(\sigma)$ est un code dominant, l'évaluation du polynôme de Schubert correspondant est immédiate. Toutefois, comme la réduction n'est valable que pour n points parmi $n!$, ce résultat est trop faible pour être utilisé à la fois sur quelconque somme sur orbite et à la fois sur quelconque transversale. Modulo le groupe d'automorphisme d'un polynôme, pour tout groupe de permutations, évaluer la somme sur orbite de ce polynôme en des points définis par la transversale correspondante nécessite la connaissance complète de l'action du groupe symétrique $\mathfrak{S}_n$ sur ce polynôme. Cela n'en demeure pas un résultat encourageant.

5.1.3 Exploration informatique sur les polynômes de Schubert

En utilisant une structure de données utilisant le type basique `function` du langage `Python` et inspirée des S.L.P. (Straight-Line Program : programme en ligne droite), nous avons exploré comment les polynômes de Schubert pouvaient s'adapter à l'approche par évaluation pour calculer les invariants des groupes de permutations.

En effet, si un polynôme est un programme simple qui, à une famille d'arguments $x_1,\dots,x_n$, retourne une sortie simple composée de sommes et produits sur ces arguments, l'action d'un groupe de permutations est aussi très aisée à utiliser pour l'informaticien. Il suffit de permuter l'ordre des arguments suivant une permutation donnée pour récupérer l'action usuelle par position du groupe symétrique sur les polynômes. Ainsi, l'opérateur de Reynolds, ou les sommes sur orbites, sont des opérations relativement simples sur les S.L.P. De même, les différences divisées peuvent être implantées de manière simple. L'informaticien doit juste se méfier de l'erreur provoquée par la division par 0 lorsqu'il spécialise le premier alphabet.

Le terme ligne droite reflète le fait que l'évaluation d'un S.L.P. se fait par un programme ne comportant qu'une branche d'exécution (pas de condition) et sans boucle (pas de commande de type *for, while, loop, ...*), ainsi le programme évaluant le S.L.P. s'exécute en ligne droite. Le S.L.P. décrit une succession d'opérations, c'est un outil très puissant en particulier pour les polynômes ; en effet un S.L.P. peut à la fois modéliser une forme factorisée (succession de multiplications de sous S.L.P.) ou bien une somme déployée. Un Vandermonde, un polynôme de

Schubert associé à un code dominant, un produit, ont souvent tout intérêt à rester factorisés et à ne pas être déployés dans la base des monômes. Une structure de données informatique d'espace vectoriel avec une base explicite contraint le calcul des coordonnées d'un vecteur pour sa création. Le S.L.P. n'entre pas ce type de structure de données, il exploite le fait que l'algèbre des polynôme est une algèbre libre avec deux opérations explicites, ainsi tout élément de cette algèbre n'est rien d'autre qu'une succession de ces deux opérations sur les générateurs de cette algèbre (variables $x_1, \ldots, x_n$). L'action du groupe symétrique (et ainsi les différences divisées) étant simple sur les S.L.P., il nous est apparu naturel de tenter d'implanter les polynômes de Schubert comme des programmes en ligne droite.

Nous présentons quelques résultats expérimentaux pour une première visualisation du comportement des polynômes de Schubert avec le morphisme clé Φ explicité précédemment dans cette étude. Cette approche des polynômes de Schubert s'inspire des travaux précédemment fait pour leur implantation, notamment ceux de Sébastien Veigneau [KV97, Vei97].

Schubert Simple aux racines de l'unité :

Schubert simple indexé par $\mathfrak{S}_4$. On spécialise les variables x_1, x_2, x_3, x_4 en $1, \zeta_4, \zeta_4^2, \zeta_4^3$ où ζ_4 est une racine 4-ième primitive de l'unité. C'est une matrice carrée de taille 24.

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & i & 1 & 1 & -1 & i & i & 1 & 1 & -i & -1 & -1 & i & i & 1 & -i & -i & -1 & -1 & i & -i & -i & -1 & -i \\
i+1 & i+1 & 0 & i+1 & 0 & i-1 & i+1 & -i+1 & 0 & -i+1 & i-1 & 0 & 0 & i-1 & -i+1 & 0 & -i+1 & -i-1 & i-1 & 0 & -i-1 & 0 & -i-1 & -i-1 \\
i & i & i & 1 & i & i & 1 & 1 & -i & 1 & i & -i & 1 & -1 & -i & 1 & -i & -i & -1 & -1 & -i & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\
i & i & -1 & i & -1 & -i & i & -i & -1 & -i & -i & -1 & 1 & -i & -i & 1 & -i & i & -i & 1 & i & 1 & i & i \\
i & -1 & i & 1 & -i & -1 & i & 1 & -i & -i & -i & i & i & -i & -i & -i & -1 & i & 1 & -i & -1 & i & 1 & i \\
i & i & 1 & i & 1 & -i & i & -i & 1 & -i & -i & 1 & -1 & -i & -i & -1 & -i & i & -i & -1 & i & -1 & i & i \\
-1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 \\
1 & -i & 1 & 1 & -1 & -i & -i & 1 & 1 & i & -1 & -1 & -i & -i & 1 & i & i & -1 & -1 & -i & i & i & -1 & i \\
i & -1 & -1 & i & 1 & 1 & -1 & -i & -1 & -1 & i & 1 & i & 1 & -i & -i & -1 & -i & i & i & 1 & -i & -i & 1 \\
i-1 & 0 & i-1 & 0 & i+1 & 0 & i-1 & 0 & -i-1 & -i-1 & i+1 & -i+1 & i-1 & i+1 & -i-1 & -i-1 & 0 & -i+1 & 0 & i+1 & 0 & -i+1 & 0 & -i+1 \\
i-1 & i-1 & 0 & i-1 & 0 & i+1 & i-1 & -i-1 & 0 & -i-1 & i+1 & 0 & 0 & i+1 & -i-1 & 0 & -i-1 & -i+1 & i+1 & 0 & -i+1 & 0 & -i+1 & -i+1 \\
-i & -i & -i & 1 & -i & -i & 1 & 1 & i & 1 & -i & i & 1 & -1 & i & 1 & i & i & -1 & -1 & i & -1 & -1 & -1 \\
-1 & -1 & i & i & i & 1 & i & -i & -i & -i & 1 & -i & -1 & i & -1 & -1 & -1 & 1 & i & 1 & 1 & 1 & -i & -i \\
i & -i & -1 & i & -1 & i & -i & -i & -1 & i & -i & -1 & -1 & i & -i & -1 & i & i & -i & -1 & -i & -1 & i & -i \\
i-1 & 0 & i-1 & 0 & -i-1 & 0 & -i-1 & 0 & -i-1 & i-1 & -i-1 & i-1 & -i-1 & i-1 & -i-1 & i-1 & 0 & i-1 & 0 & i-1 & 0 & -i-1 & 0 & -i-1 \\
-1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-i & 1 & -i & 1 & i & 1 & i & 1 & i & -i & i & -i & i & -i & i & -i & 1 & -i & 1 & -i & 1 & i & 1 & i \\
-i+1 & -i+1 & 0 & i+1 & 0 & i+1 & i+1 & -i+1 & 0 & -i+1 & i+1 & 0 & 0 & -i+1 & i+1 & 0 & i+1 & -i+1 & -i+1 & 0 & -i+1 & 0 & i+1 & i+1 \\
-1 & -i & 1 & -1 & -1 & -i & -i & -1 & 1 & i & 1 & -1 & i & -i & -1 & -i & i & 1 & 1 & i & i & -i & 1 & i \\
-i & i & -i & 1 & -i & i & -1 & 1 & i & -1 & -i & i & -1 & 1 & i & -1 & -i & i & -1 & 1 & -i & 1 & -1 & 1 \\
1 & 1 & i & i & i & -1 & i & -i & -i & -i & -1 & -i & 1 & i & 1 & 1 & 1 & -1 & i & -1 & -1 & -1 & -i & -i \\
1 & i & i & i & -i & -i & -1 & -i & -i & -1 & 1 & i & i & -1 & 1 & -i & -i & 1 & -i & -i & i & i & i & -1
\end{pmatrix}$$

Reynolds de Schubert Simple aux racines de l'unité :

Soit $C_4 = \langle (1,2,3,4) \rangle$ le groupe cyclique d'ordre 4. Cette matrice est obtenue de la précédente en faisant la somme des colonnes par classe d'équivalence à droite de $\mathfrak{S}_4/C_4$. On obtient donc l'évaluation par le morphisme clé Φ des polynômes $4R(Y_\sigma)$ où σ parcourt $\mathfrak{S}_4$.

La première ligne $(4,4,4,4,4,4)$ est donc $\Phi(4R(1))$. Pour la seconde, la permutation qui l'indexe est la transposition simple $(1,2)$ et $Y_{(1,2)} = x_1$. $4R(Y_{(1,2)}) = 4(x_1 + x_2 + x_3 + x_4)$ qui est un polynôme symétrique. Son évaluation est ainsi nulle.

$$\begin{pmatrix}
4 & 4 & 4 & 4 & 4 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2i & -2i & 2i & -2i & 0 \\
0 & -2i & 2i & -2i & 2i & 0 \\
0 & 2i & -2i & 2i & -2i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2i+2 & 2i-2 & -2i-2 & -2i+2 & 0 \\
0 & 2i+2 & 2i-2 & -2i-2 & -2i+2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2i+2 & 2i-2 & -2i-2 & -2i+2 & 0 \\
4i & -2 & -2 & -2 & -2 & -4i \\
4i-4 & -2 & -2 & -2 & -2 & -4i-4 \\
-4 & 0 & 0 & 0 & 0 & -4 \\
-4i & 2 & 2 & 2 & 2 & 4i \\
-4i+4 & 2 & 2 & 2 & 2 & 4i+4 \\
0 & 2i-2 & 2i+2 & -2i+2 & -2i-2 & 0 \\
0 & -2i+2 & -2i-2 & 2i-2 & 2i+2 & 0 \\
0 & 2i-2 & 2i+2 & -2i+2 & -2i-2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Schubert Double aux racines de l'unité :

Schubert double indexé par $\mathfrak{S}_4$. Soit i une racine 4-ième primitive de l'unité. On spécialise le premier alphabet x_1, x_2, x_3, x_4 en $1, i, i^2, i^3$ et le second alphabet y_1, y_2, y_3, y_4 aussi en $1, i, i^2, i^3$. C'est une matrice carrée de taille 24.

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\cdot & i-1 & \cdot & \cdot & -2 & i-1 & i-1 & \cdot & \cdot & -i-1 & -2 & -2 & i-1 & i-1 & \cdot & -i-1 & -i-1 & -2 & -2 & i-1 & -i-1 & -i-1 & -2 & -i-1 \\
\cdot & \cdot & -i-1 & \cdot & -i-1 & -2 & \cdot & -2i & -i-1 & -2i & -2 & -i-1 & -i-1 & -2 & -2i & -i-1 & -2i & -2i-2 & -2 & -i-1 & -2i-2 & -i-1 & -2i-2 & -2i-2 \\
\cdot & \cdot & \cdot & -i+1 & \cdot & \cdot & -i+1 & -i+1 & -2i & -i+1 & \cdot & -2i & -i+1 & -i-1 & -2i & -i+1 & -2i & -2i & -i-1 & -i-1 & -2i & -i-1 & -i-1 & -i-1 \\
\cdot & \cdot & \cdot & \cdot & 2i+2 & \cdot & \cdot & \cdot & \cdot & 2i-2 & 2i+2 & 2i+2 & \cdot & \cdot & \cdot & 2i-2 & 2i-2 & 2i+2 & 2i+2 & \cdot & 2i-2 & 2i-2 & 2i+2 & 2i-2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & -2i+2 & \cdot & \cdot & \cdot & \cdot & -2i+2 & \cdot & 2 & -2i+2 & \cdot & 2 & \cdot & 2i+2 & -2i+2 & 2 & 2i+2 & 2 & 2i+2 & 2i+2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2i & \cdot & \cdot & -2 & \cdot & 4i & 2i & 2 & \cdot & -2 & 2i-2 & 4i & 2i+2 & 2 & 2i-2 & 2i & 2i+2 & 2i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -2i-2 & \cdot & -2i-2 & \cdot & \cdot & -2 & \cdot & -2i-2 & -2 & -2i-2 & 2i-2 & \cdot & -2 & 2i-2 & -2 & 2i-2 & 2i-2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2i-2 & \cdot & \cdot & 2i-2 & \cdot & 2i+2 & 2i-2 & \cdot & 2i-2 & 2i-2 & 2i+2 & 2i+2 & 2i-2 & 2i+2 & 2i+2 & 2i+2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4i & \cdot & \cdot & \cdot & \cdot & \cdot & 4i & 4i & \cdot & \cdot & \cdot & 4i & 4i & \cdot & 4i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4 & \cdot & \cdot & \cdot & \cdot & -4i & \cdot & -4i & -4 & \cdot & -4i+4 & -4i & -4i & -4i+4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4i+4 & \cdot & \cdot & \cdot & \cdot & 4 & -4i+4 & -4i & \cdot & 4 & -4i+4 & -4i & -4i+4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -2i+2 & \cdot & \cdot & -2i+2 & \cdot & -4i+4 & \cdot & -2i+2 & -4i+4 & -2i+2 & -4i+4 & -4i+4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4 & \cdot & \cdot & \cdot & \cdot & -4 & -4 & \cdot & -4 & -4 & -4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4i & \cdot & 4i & 4 & \cdot & 4 & 4 & 4 & -4i+4 & -4i+4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4i-4 & \cdot & \cdot & \cdot & \cdot & -8i & -4i-4 & \cdot & -8i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -4i+4 & \cdot & \cdot & \cdot & -4i+4 & -8i & \cdot & -8i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -8 & \cdot & \cdot & -8 & \cdot & -8 & -8 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4i+4 & \cdot & \cdot & 8i & 4i+4 & 8i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4i-4 & \cdot & 4i-4 & 8i & 8i \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 8i-8 & \cdot & \cdot & 8i-8 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 8i+8 & \cdot & 8i+8 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -8i+8 & -8i+8 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -16i
\end{pmatrix}$$

Reynolds de Schubert double aux racines de l'unité :

Soit $C_4 = \langle (1,2,3,4) \rangle$ le groupe cyclique d'ordre 4. Cette matrice est obtenue de la précédente en faisant la somme des colonnes par classe d'équivalence à droite de $\mathfrak{S}_4/C_4$. On obtient donc l'évaluation par le morphisme clé Φ des polynômes $4R(Y_\sigma)$ où σ parcourt $\mathfrak{S}_4$.

$$\begin{pmatrix}
4 & 4 & 4 & 4 & 4 & 4 \\
-4 & -4 & -4 & -4 & -4 & -4 \\
-4i-4 & -4i-4 & -4i-4 & -4i-4 & -4i-4 & -4i-4 \\
-4i & -4i & -4i & -4i & -4i & -4i \\
4i & 4i & 4i & 4i & 4i & 4i \\
4 & 2i+4 & -2i+4 & 2i+4 & -2i+4 & 4 \\
4i & 2i & 6i & 2i & 6i & 4i \\
-4 & 2i-4 & -2i-4 & 2i-4 & -2i-4 & -4 \\
4i & 4i & 4i & 4i & 4i & 4i \\
4i & 4i & 4i & 4i & 4i & 4i \\
-4i & -4i+4 & -4 & -8i & -4i & -4i \\
-4i+4 & 4 & -4i+4 & -4i & -8i+8 & -4i+4 \\
-4i+4 & -6i+6 & -2i+2 & -6i+6 & -2i+2 & -4i+4 \\
-4 & -4 & -4 & -4 & -4 & -4 \\
4 & 8 & 4i & -4i+4 & 4 & 4 \\
0 & -8i & 0 & -4i-4 & -4i-4 & -8i \\
0 & -4i+4 & -4i+4 & 0 & -8i & -8i \\
-8 & -8 & 0 & -8 & 0 & -8 \\
0 & 0 & 4i+4 & 4i+4 & 8i & 8i \\
0 & 4i-4 & 0 & 8i & 4i-4 & 8i \\
0 & 8i-8 & 0 & 0 & 0 & 8i-8 \\
0 & 0 & 0 & 0 & 8i+8 & 8i+8 \\
0 & 0 & 0 & -8i+8 & 0 & -8i+8 \\
0 & 0 & 0 & 0 & 0 & -16i
\end{pmatrix}$$

5.2 Choix alternatifs pour les invariants d'un groupe de permutation

Le tableau suivant récapitule et compare les différentes familles rencontrées pour construire un système d'invariants secondaires.

Familles remarquables de polynômes pour l'anneau des invariants

On désigne par $\mathcal{F}$ la famille de polynômes de Schubert construite dans la proposition 5.1.9.

homogène : porte une marque ✓ lorsque la famille est composée de polynômes homogènes.
gén. quo. : porte une marque ✓ lorsque la famille génère le quotient $\mathbb{K}[\mathbf{x}]^G/\operatorname{Sym}(\mathbf{x})^+$.
base quo. : porte une marque ✓ lorsque la famille forme une base du quotient $\mathbb{K}[\mathbf{x}]^G/\operatorname{Sym}(\mathbf{x})^+$.
gén. mod. : porte une marque ✓ lorsque la famille génère le module $\mathbb{K}[\mathbf{x}]^G$ comme $\operatorname{Sym}(\mathbf{x})$-module.
base mod. : porte une marque ✓ lorsque la famille forme une base du module $\mathbb{K}[\mathbf{x}]^G$ comme $\operatorname{Sym}(\mathbf{x})$-module.
évaluation : donne des informations sur la complexité du calcul des images par le morphisme d'évaluation Φ

Famille de polynôme	**cardinal**	**homogène**	**degrés**	**gén. quo.**	**base quo.**	**gén. mod.**	**base mod.**	**évaluation**
Monômes sous l'escalier	$n!$	✓	$0, 1, \ldots, \binom{n}{2}$	✓		✓		légère
avec réduction de Gauss	$\frac{n!}{\lvert G\rvert}$	✓	respecte $S(\mathbb{K}[\mathbf{x}]^G, z)$	✓	✓	✓	✓	légère
Interpolateurs de Lagrange	$\frac{n!}{\lvert G\rvert}$		$\binom{n}{2}$	✓				gratuite
Polynômes de Schubert simples	$n!$	✓	$0, 1, \ldots, \binom{n}{2}$	✓		✓		lourde
avec réduction de Gauss	$\frac{n!}{\lvert G\rvert}$	✓	respecte $S(\mathbb{K}[\mathbf{x}]^G, z)$	✓	✓	✓	✓	lourde
Polynômes de Schubert doubles	$n!$		$0, 1, \ldots, \binom{n}{2}$	✓				creuse
Schubert doubles de $\mathcal{F}$	$\frac{n!}{\lvert G\rvert}$		globalement élevé $\leqslant \binom{n}{2}$	✓	✓			creuse

TABLE 5.1 – Tableau récapitulatif des familles de polynômes dont les images par Reynolds (ou par somme sur orbite) présentent des propriétés remarquables pour l'anneau des invariants.

Toutes les familles précédentes présentent des caractéristiques et une utilité différente. Elles ont en commun d'être toute génératrices du quotient $\mathbb{K}[\mathbf{x}]^G/\operatorname{Sym}(\mathbf{x})^+$. C'est une propriété minimale avant de pouvoir espérer rechercher ou construire des invariants secondaires.

Seules deux familles présentent ici la possibilité d'en extraire un système d'invariants secondaires : les polynômes de Schubert simples (le second alphabet $\mathbf{y}$ est spécialisé en $\{0, 0, \ldots\}$) et les monômes sous l'escalier. En effet, il faut tout d'abord trouver des candidats homogènes pour espérer construire une sous famille qui génère $\mathbb{K}[\mathbf{x}]^G$ en tant que $\operatorname{Sym}(\mathbf{x})$-module.

Pour l'évaluation, celle des sommes sur orbite de monômes a déjà été présentée plus haut. C'est un calcul relativement praticable sur les vecteurs d'entiers. Par définition et construction, les évaluations des interpolateurs de Lagrange sont triviales. En ce qui concerne les polynômes de Schubert, leurs évaluations exigent leur construction et cette dernière se révèlent relativement coûteuse lorsque le nombre de variables grandit. Les polynômes de Schubert doubles sont légèrement plus simple à évaluer parce qu'en tant que polynômes d'interpolation, leurs évaluations sont légèrement creuses.

Quatrième partie

Quotient de l'algèbre de Hecke affine au niveau 0

Voici une étude suggérée par Anne Schilling et Nicolas M. Thiéry. Ce travail, partiellement supporté par NSF grants DMS-0652652, commença durant une visite de l'auteur à l'université de Californie à Davis au printemps 2008.

Ce travail a été mené via l'exploration informatique utilisant les systèmes open-source pour les mathématiques `GAP` [GAP08], `Sage` [S$^+$09] et ses modules de combinatoires algébriques développés par la communauté de `Sage-Combinat` [SCc08].

5.3 Introduction

La groupe algèbre de Hecke $H\mathring{W}$ d'un groupe de Coxeter fini, introduite par Florent Hivert et Nicolas M. Thiéry, est obtenue à partir de $\mathring{W}$ en recollant de manière appropriée sa 0-Hecke algèbre et son algèbre de groupe [HT09].

C'est une algèbre de matrices, endomorphismes de l'algèbre du groupe $\mathbb{C}[\mathring{W}]$. Avec la contribution d'Anne Schilling, ils donnèrent une construction alternative de cette algèbre [HST09]. Cette nouvelle construction en fait toujours une algèbre d'opérateurs vivant dans le même espace. La différence est qu'ils exhibent de nouveaux générateurs comportant deux paramètres formels q_1, q_2, qui peuvent être spécialisés. Ces nouveaux générateurs $T_1, \dots, T_n, T_0$ sont construits à partir des générateurs de l'algèbre de Hecke affine dont l'action est vue au niveau 0. Ils prouvèrent que lorsque $q = -\frac{q_1}{q_2}$ n'est pas une petite racine de l'unité, l'algèbre $cl(H(W)(q_1, q_2))$ est un quotient de l'algèbre de Hecke affine et est isomorphe à la groupe algèbre de Hecke $H\mathring{W}$. Pour construire ces générateurs, il faut tout d'abord accéder aux réflexions et projections du groupe fini $\mathring{W}$ et aussi définir l'extra générateur d'index 0 issu du côté affine. Ce générateur particulier est une interpolation entre la réflexion et la projection correspondant à la plus haute racine mais la projection projetant sur le côté négatif de l'hyperplan. Ainsi construits, les opérateurs $T_1, \dots, T_n, T_0$ vérifient les relations quadratiques déformées usuelles dans les algèbres de Hecke $(T_i - q_1)(T_i - q_2) = Id$. Mais aussi deux opérateurs T_i, T_j pour $i \neq j$, vérifient la même relation de tresse que les correspondantes réflexions s_i, s_j dans le groupe de Coxeter affine W.

L'objectif de cette partie est de déterminer pour chaque type de Cartan, quelles sont les racines de l'unité pour lesquelles $cl(H(W)(q_1, q_2))$ dégénère.

Nous présenterons tout d'abord le contexte formel et le théorème qui motive de telles investigations. Nous exposerons les structures algébriques qui contiennent la solution d'un tel problème. A partir de là, nous montrerons comment utiliser l'ordinateur pour rechercher ces racines de l'unité mais aussi les limites de tels calculs relatifs à leur taille. La lecture des résultats motivera l'énonciation de quelques conjectures. L'objectif ultime d'un tel travail étant de détecter mathématiquement de telles dégénérations sans calculer explicitement les bases des algèbres considérées.

5.4 Préliminaires

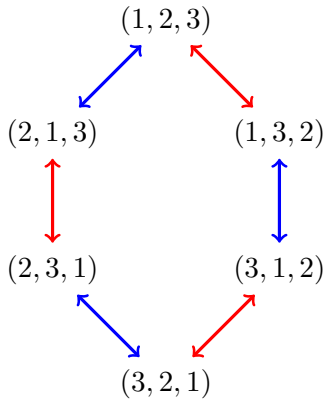
Dans ce chapitre, nous rappelons les notations et définitions des groupes de Coxeter, des algèbre de Iwahori-Hecke et des groupes algèbres de Hecke. Nous exposons ensuite le résultat central de [HST09] qui est le point de départ de ce travail.

5.4.1 Groupes de Coxeter

Définition 5.4.1 (groupe de Coxeter)**.** *Un* Système de Coxeter *[Hum90] est une paire (W, S) consistant en un groupe W et un ensemble de générateurs $S \in W$, sujets à des relations de la forme : $(ss')^{m(s,s')} = 1$, où $m(s, s) = 1$, $m(s, s') = m(s', s) \geqslant 2$ pour $s, s' \in S$. Dans le cas où aucune relation intervient entre une paire s, s', on adapte la convention $m(s, s') = \infty$. Formellement, W est le quotient F/N, où F est le groupe libre engendré par S et N est le sous groupe normal engendré par les éléments de la forme $(ss')^{m(s,s')}$.*

Par exemple, $\mathfrak{S}_3$ est engendré par deux réflexions simples s_1 et s_2. Ces deux générateurs sont tout d'abord des involutions $s_1^2 = 1, s_2^2 = 1$. Ils sont, de plus, soumis à la relation de tresse $s_1 s_2 s_1 = s_2 s_1 s_2$. Ces relations permettent de déplier un groupe fini.

FIGURE 5.2 – Graphe de Cayley du groupe $\mathfrak{S}_3$



Dans `Sage`, on construit le groupe symétrique avec la commande suivante.

```
sage: clW = WeylGroup(['A',2]); clW
Weyl Group of type ['A', 2] (as a matrix group acting on the ambient space)
sage: latex(clW)
```

$$\left\langle \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right\rangle$$

```
sage: latex(clW.list())
```

$$\left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right]$$

5.4.2 Algèbres de (Iwahori)-Hecke

Définition 5.4.2 (Algèbres de Iwahori-Hecke)**.** *Soit W un groupe de Coxeter et q_1, q_2 deux paramètres formels. Lorsque $q_2 \neq 0$, on pose $q =: -\frac{q_1}{q_2}$. La* (générique, Iwahori) (q_1, q_2)-algèbre de Hecke $H(W)(q_1, q_2)$ *de W est la $\mathbb{C}$-algèbre engendrée par les opérateurs T_i sujets aux relations :*

$$(T_i - q_1)(T_i - q_2) = 0 \quad and \quad \underbrace{T_i T_j \cdots}_{m(i,j)} = \underbrace{T_j T_i \cdots}_{m(i,j)} \quad for i \neq j. \tag{5.4.1}$$

La dimension de cette algèbre est $|W|$. Une base est donnée par les éléments $T_w := T_{i_1} \cdots T_{i_r}$ où $w \in W$ et $i_1, \ldots, i_r$ est un mot réduit pour w. La multiplication à droite dans cette base est donnée par :

$$T_w T_i = \begin{cases} (q_1 + q_2)T_w - q_1 q_2 T_{ws_i} & \text{si i est une descente de w,} \\ T_{ws_i} & \text{sinon.} \end{cases} \tag{5.4.2}$$

À $q_1 = 1, q_2 = -1$ (soit $q = 1$), on retrouve l'algèbre du groupe $\mathbb{C}[W]$; en général, lorsque $q_1 + q_2 = 0$ on retrouve toujours $\mathbb{C}[W]$ à dilatation près des générateurs : $s_i = \frac{1}{q_1} T_i$. Aussi, lorsque q_1 et q_2 sont tout deux non nuls et q n'est pas une racine de l'unité, $H(W)(q_1, q_2)$ est toujours isomorphe à l'algèbre du groupe mais l'isomorphisme n'est pas trivial.

D'un autre côté, prenant $q_1 = 0$ et $q_2 \neq 0$, on obtient la 0-algèbre de Hecke $H(W)(0)$, cette algèbre est toujours une algèbre de monoïde pour le 0-monoïde de Hecke $\{\pi_w | w \in W\}$ engendré par les idempotents $\pi_i := \frac{1}{q_2} T_i$. À $q_1 = q_2 = 0$, on obtient l'algèbre de Nil-Coxeter.

Utilisant la représentation régulière à droite, s_i, π_i et T_i peuvent être réalisés comme opérateurs agissant sur $\mathbb{C}.\mathring{W}$.

```
sage: w1 = clW.simple_reflection(1)
sage: w2 = clW.simple_reflection(2)
sage: s1 = matrix(QQ, [[1 if i*w1 == j else 0 for i in clW] for j in clW])
sage: s2 = matrix(QQ, [[1 if i*w2 == j else 0 for i in clW] for j in clW])
sage: latex([s1, s2])
```

$$\left[\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}\right]$$

Dans ce même espace, voici les projections simples :

```
sage: p1 = clW.simple_projection(1)
sage: p2 = clW.simple_projection(2)
sage: pi1 = matrix(QQ, [[1 if p1(i) == j else 0 for i in clW] for j in clW])
sage: pi2 = matrix(QQ, [[1 if p2(i) == j else 0 for i in clW] for j in clW])
sage: latex([pi1, pi2])
```

$$\left[\begin{pmatrix} 0&0&0&0&0&0\\ 1&1&0&0&0&0\\ 0&0&1&0&1&0\\ 0&0&0&1&0&1\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&1&0&0\\ 1&0&0&0&1&0\\ 0&1&0&0&0&1 \end{pmatrix}\right]$$

Ainsi, les générateurs de l'algèbre de Hecke classique sont :

```
sage: K.<q> = LaurentPolynomialRing(QQ, 1)
sage: T1 = (1-q)*pi1 + q*s1             # les générateurs de Hecke sont des
sage: T2 = (1-q)*pi2 + q*s2             # interpolations entre projections
sage: latex([T1, T2])                   # et symétries
```

$$\left[\begin{pmatrix} 0&q&0&0&0&0\\ 1&-q+1&0&0&0&0\\ 0&0&-q+1&0&1&0\\ 0&0&0&-q+1&0&1\\ 0&0&q&0&0&0\\ 0&0&0&q&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&0&q&0\\ 0&0&0&0&0&q\\ 0&0&0&q&0&0\\ 0&0&1&-q+1&0&0\\ 1&0&0&0&-q+1&0\\ 0&1&0&0&0&-q+1 \end{pmatrix}\right]$$

```
sage: Mspace = gap.MatrixAlgebra(QQ, clW.cardinality())
sage: # dimension de la représentation régulière à droite
sage: gap.SubalgebraWithOne(Mspace, [s1,s2]).Dimension()
6
sage: # dimension de la 0-algèbre de Hecke
sage: gap.SubalgebraWithOne(Mspace, [pi1,pi2]).Dimension()
6
```

5.4.3 Groupe algèbres de Hecke

Définition 5.4.3 (Groupe algèbre de Hecke)**.** *[HT09] Soit $\mathring{W}$ un groupe de Coxeter fini, et $\mathbb{C}\mathring{W}$ l'espace vectoriel de dimension $|\mathring{W}|$ qu'il engendre. Comme nous avons vu, l'on peut réaliser simultanément la 0-algèbre de Hecke $H(\mathring{W})(0)$ et l'algèbre du groupe $\mathbb{C}[\mathring{W}]$ dans $End(\mathbb{C}\mathring{W})$ à travers leur représentation régulière à droite. La* Groupe algèbre de Hecke $H\mathring{W}$ *de $\mathring{W}$ est la plus petite sous-algèbre de $End(\mathbb{C}\mathring{W})$ contenant ces deux algèbres.*

L'algèbre de Hecke groupe est ainsi engendrée par les réflexions simples $(s_i)_{i\in I}$ et les projections simples $(\pi_i)_{i\in I}$. Par interpolation, $H\mathring{W}$ contient toutes les algèbres de Hecke $H(W)(z)$ pour z complexe. Une base de $H\mathring{W}$ est donnée par $\{w\pi_{w'}|DR(w)\cap DL(w')=\emptyset\}$ [HT09]. Une vision plus conceptuelle de cette algèbre a été proposée par Hivert et Thiéry comme il suit ; un vecteur v dans $\mathbb{C}W$ est i-antisymétrique à gauche si $s_iv=-v$; alors $H\mathring{W}$ est la sous algèbre de $End(\mathbb{C}W)$ contenant tous les opérateurs préservant les antisymétries à gauche.

```
sage: # Dimension de l'algèbre de Hecke groupe pour S_3
sage: gap.SubalgebraWithOne(Mspace, [s1,s2,pi1,pi2]).Dimension()
19
```

5.4.4 Action au niveau 0 des types affines

L'action au niveau zéro [Kas05] consiste à regarder l'action des éléments d'un groupe affine de Weyl W sur le groupe fini $\mathring{W}$. Ces deux groupes de Coxeter sont définis par une liste de générateurs et des relations. La version affine possède un générateur supplémentaire usuellement indexé par 0. Ce générateur correspond à un noeud spécial du diagramme de Dynkin. Les autres générateurs s'identifient naturellement avec ceux de la version finie nous permettant de définir une action à droite de W sur $\mathring{W}$. Soit $cl : W \to \mathring{W}$ l'application quotient canonique, on construit une action à droite au niveau 0 en posant pour $\omega \in \mathring{W}$ et $s_i \in W$, $w.s_i := wcl(s_i)$. Ainsi l'extra générateur affine s_0 est envoyé sur la réflexion associée à la plus longue racine (pour le type A_n, $cl(s_0) = s_1 s_2 \dots s_{n-1} s_n s_{n-1} \dots s_2 s_1$) [Ram03].

Regardons ce qui se passe dans $\mathbb{C}[\mathring{W}]$ pour les réflexions et projections simples issues de la version affine en type A.

TABLE 5.2 – Réflections et projections simples pour le type de Cartan A_4

$$\begin{array}{cc} \underline{35241}\underline{} & \underline{35241}\underline{} \\ {\scriptstyle s_1}\downarrow \quad \downarrow{\scriptstyle s_4} & {\scriptstyle \pi_1}\downarrow \quad \downarrow{\scriptstyle \pi_4} \\ \underline{53}214 & \underline{53}\underline{241} \\ {\scriptstyle s_1}\downarrow \quad \downarrow{\scriptstyle s_4} & {\scriptstyle \pi_1}\downarrow \quad \downarrow{\scriptstyle \pi_4} \\ \underline{35}\underline{241} & \underline{53}\underline{241} \end{array} \tag{5.4.3}$$

Ceci illustre l'action de s_1 qui échange les deux premières positions de la permutation 35241. Appliquée deux fois, on retrouve notre point de départ, ce qui est consistant avec la relation $s_1^2 = 1$ du groupe symétrique $\mathfrak{S}_5$. π_1, la première projection simple est un idempotent, il faut voir cet opérateur comme un opérateur de tri. Une fois triée, trier de nouveau ne change pas le résultat.

Regardons l'action de l'extra générateur issue de la version affine pour $\mathring{W} = \mathfrak{S}_5$. La plus haute racine est ici $e_1 - e_5$ si $(e_1, e_2, \dots, e_5)$ désigne la base canonique de l'espace ambiant $\mathbb{R}^5$.

TABLE 5.3 – Introduction de s_0 et π_0 pour le type de Cartan A_4

$$\begin{array}{cc} \underline{3}5241\underline{} & \underline{3}5241\underline{} \\ {\scriptstyle s_0}\downarrow & {\scriptstyle \pi_0}\downarrow \\ \underline{15243} & \underline{15243} \\ {\scriptstyle s_0}\downarrow & {\scriptstyle \pi_0}\downarrow \\ \underline{3}5241\underline{} & \underline{1}5243\underline{} \end{array} \tag{5.4.4}$$

```
sage: w0 = clW.simple_reflections()[0]
sage: p0 = clW.simple_projection(0)
sage: s0 = matrix(QQ, [[1 if i*w0 == j else 0 for i in clW] for j in clW])
sage: pi0 = matrix(QQ, [[1 if p0(i) == j else 0 for i in clW] for j in clW])
sage: latex([s0, pi0])
```

$$\left[\begin{pmatrix} 0&0&0&1&0&0\\ 0&0&1&0&0&0\\ 0&1&0&0&0&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0 \end{pmatrix}, \begin{pmatrix} 1&0&0&1&0&0\\ 0&1&1&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&0 \end{pmatrix}\right]$$

5.5 La groupe algèbre de Hecke comme quotient de l'algèbre de Hecke affine

Définition 5.5.1. *On définit la hauteur d'une racine duale $\lambda^\vee$ de $\mathring{W}$ comme $ht(\lambda^\vee) := \langle \lambda^\vee, \mathring{\rho} \rangle$, où $\mathring{\rho} := \frac{1}{2}\sum_{\alpha \in \mathring{R}^+} \alpha$. En particulier, une racine duale a pour hauteur 1 si et seulement si c'est en fait une racine duale simple ($\mathring{\rho}$ est par ailleurs la somme des poids fondamentaux du système de racines associé à $\mathring{W}$). Pour $\mathring{W}$ un groupe de Weyl fini, nous noterons $\theta^\vee$ la plus haute racine duale de $\mathring{W}$.*

Theorème 5.5.2. *[HST09] Soit W un groupe affine de Weyl. Hormis quand $q := -\frac{q_1}{q_2}$ est une k-ième racine de l'unité avec $k \leqslant 2ht(\theta^\vee)$, le morphisme $cl(H(W)(q_1, q_2)) \to H\mathring{W}$ est surjectif et réalise l'algèbre de Hecke groupe $H\mathring{W}$ dans un quotient de l'algèbre de Hecke affine $H(W)(q_1, q_2)$.*

Ce résultat motive l'énonciation du problème suivant qui constitue le point de départ de cette étude :

Problème 5.5.3. *Déterminer pour quelles racines de l'unité q le morphisme $cl(H(W)(q_1, q_2)) \to H\mathring{W}$ n'est pas surjectif.*

Définition 5.5.4. *Soit W un groupe affine de Weyl. Une racine de l'unité q est dite* bonne *pour W (ou* bonne *pour la version finie $\mathring{W}$) si le morphisme $cl(H(W)(q_1, q_2)) \to H\mathring{W}$ est surjectif. Sinon, on dira que cette racine q est* mauvaise *pour W.*

```
sage: # On calcule ici la dimension pour q = 0
sage: gap.SubalgebraWithOne(Mspace, [pi0,pi1,pi2]).Dimension()
19
sage: # On calcule ici la dimension pour q = -1
sage: gap.SubalgebraWithOne(Mspace, [2*pi0-s0,2*pi1-s1,2*pi2-s2]).Dimension()
18
sage: # On calcule ici la dimension pour q = 1^(1/3)
sage: K.<zeta> = CyclotomicField(3)
sage: gens = [(1-zeta)*pi0+zeta*s0,(1-zeta)*pi1+zeta*s1,(1-zeta)*pi2+zeta*s2]
sage: Mspace = gap.MatrixAlgebra(K, clW.cardinality())
sage: gap.SubalgebraWithOne(Mspace, gens).Dimension()
11
```

Ainsi, nous dirons ici que les racines seconde et troisième de l'unité sont mauvaises pour le type de Cartan A_2.

5.6 Méthodologie

5.6.1 Génération et spécialisation

Nous rappelons le lemme suivant bien connu.

Lemme 5.6.1. *Soit q un paramètre formel et un nombre complexe z. Soit $X_1(q), X_2(q), \ldots, X_n(q)$ une famille de vecteurs vivant dans un $\mathbb{C}(q)$-espace vectoriel de dimension d. On suppose de plus que z n'est racine d'aucun dénominateur apparaissant dans les $X_i(q)$. Soit $X_1(z), X_2(z), \ldots, X_n(z)$ les vecteurs de $\mathbb{C}^d$ obtenus en spécialisant le paramètre q en la valeur z.*

Si la famille $X_1(z), X_2(z), \ldots, X_n(z)$ est libre sur $\mathbb{C}$, alors la famille $X_1(q), X_2(q), \ldots, X_n(q)$ est libre sur $\mathbb{C}(q)$. En particulier,

$$dim < X_1(z), X_2(z), \ldots, X_n(z) >_{\mathbb{C}} \qquad \leqslant \qquad dim < X_1(q), X_2(q), \ldots, X_n(q) >_{\mathbb{C}(q)}$$

Si la famille $X_1(z), X_2(z), \ldots, X_n(z)$ est libre sur $\mathbb{C}$, alors la famille $X_1(q), X_2(q), \ldots, X_n(q)$ est libre $\mathbb{C}[q]$. En particulier,

$$dim < X_1(z), X_2(z), \ldots, X_n(z) >_{\mathbb{C}} \qquad \leqslant \qquad rang < X_1(q), X_2(q), \ldots, X_n(q) >_{\mathbb{C}[q]}$$

Il y a aussi équivalence entre être libre sur $\mathbb{C}[q]$ et $\mathbb{C}(q)$. Ainsi

$$dim < X_1(z), X_2(z), \ldots, X_n(z) >_{\mathbb{C}(q)} \qquad = \qquad rang < X_1(q), X_2(q), \ldots, X_n(q) >_{\mathbb{C}[q]}$$

Démonstration. Nous prouvons ce lemme par contraposition. En effet, une relation sur $\mathbb{C}(q)$ induit une relation non triviale sur $\mathbb{C}$.

Supposons que $X_1(q), X_2(q), \ldots, X_n(q)$ n'est pas libre sur $\mathbb{C}(q)$. Il existe alors une relation de la forme :

$$\sum_{l=1}^{k} \frac{P_l(q)}{Q_l(q)} X_l(q) = 0$$

Nous voulons spécialiser cette relation en z mais il nous faut être prudent pour obtenir quelque chose de consistant. Soit $v_{(q-z)}$ la valuation de $\mathbb{C}[q]$ en $(q-z)$ que nous étendons à $\mathbb{C}(q)$ par :

$$v_{(q-z)}(\frac{P(q)}{Q(q)}) = v_{(q-z)}P(q) - v_{(q-z)}Q(q)$$

Avec ceci, on construit un entier relatif m comme il suit :

$$m = min\{v(q-z)(\frac{P_l(q)}{Q_l(q)}), 1 \leqslant l \leqslant k\}$$

Maintenant, la relation :

$$\sum_{l=1}^{k} (q-z)^{-m} \frac{P_l(q)}{Q_l(q)} X_l(q) = 0$$

est toujours vérifiée dans $\mathbb{C}(q)$ mais sa spécialisation en z est maintenant non triviale sur $\mathbb{C}$ car un des coefficients est non nul par définition de m. □

Ce premier lemme de spécialisation montre que, pour n'importe quel nombre complexe, le résidu au niveau zéro de l'algèbre affine de Hecke, une fois spécialisée, a pour dimension sur $\mathbb{C}$, au plus, la dimension de l'algèbre de Hecke groupe sur $\mathbb{C}(q)$. Nous savons que par construction, il a une inclusion entre ces deux structures ; toutefois il n'existe pas de morphisme évident qui les relie et seulement un lemme comme le précédant fournit une comparaison de ces deux dimensions.

Regardons un autre résultat simple sur la spécialisation. Les algèbres considérées dans cette étude sont des algèbres avec représentations concrètes définies par générateurs. Les générateurs du coté formel sont des matrices dont les coefficients sont des polynômes en q, des générateurs deviennent des matrices à coefficients complexes une fois spécialisés. L'évaluation $q \to z$ possède quelques propriétés.

Remarque 5.6.2. *Soit A une $\mathbb{C}[q]$-algèbre définie par générateurs $A :=< X_1(q), X_2(q) \ldots, X_n(q) >_{\mathbb{C}[q]}$. Soit z un nombre complexe et A_z l'algèbre spécialisée en z. Ainsi $A_z :=< X_1(z), X_2(z), \ldots, X_n(z) >_{\mathbb{C}}$. La fonction de spécialisation*

$$\phi : \left\{ \begin{array}{ccc} A & \twoheadrightarrow & A_z \\ X_i(q) & \mapsto & X_i(z) \end{array} \right.$$

définit un morphisme surjectif de $\mathbb{C}$-algèbre.

5.6.2 Utilisation de forme normale de Smith

Dans cette section, nous montrons comment l'utilisation de forme normale de Smith permet d'obtenir, en une fois, toutes les racines mauvaises.

Les observations sur ces algèbres de matrices et leurs spécialisations motivent l'utilisation de formes normales de Smith. Le point important est que les opérations de dépliage d'algèbres définies par générateurs et la spécialisation commutent. Cela signifie que déplier une base formelle de l'algèbre puis spécialiser ou spécialiser les générateurs et ensuite déplier l'algèbre pour en obtenir une base donneront des résultats similaires.

Définition 5.6.3 (Forme normale de Smith)**.** *Soit d un entier positif et $X_1(q), X_2(q), \ldots, X_n(q)$ des vecteurs du $\mathbb{C}[q]$-module libre de rang d. Soit M une matrice de taille (n, d) dont les lignes sont les vecteurs $(X_i)_{1 \leqslant i \leqslant n}$. On définit une* forme normale de Smith *pour la matrice M une expression de la forme :*

$$U \cdot M \cdot V = D$$

où $U \in SL_n(\mathbb{C}[q])$, $V \in SL_d(\mathbb{C}[q])$ et $D \in Diag_{n,d}(\mathbb{C}[q])$. Les entrées diagonales $\{D_i\}_{1 \leqslant i \leqslant n}$ de D seront appelées les diviseurs élémentaires. Il vérifient $D_i \mid D_{i+1}$ pour $1 \leqslant i \leqslant n-1$ et sont uniques à unité près (i.e. un complexe non nul).

Proposition 5.6.4. *Soit d un entier naturel et A une algèbre, sous-module de $\mathbb{C}[q]^d$. A est définie par une famille finie de vecteurs de $\mathbb{C}[q]^d$. On suppose, de plus, que le produit de A est une opération polynomiale (les coefficients des produits doivent être polynomiaux en les coefficients des termes). Il existe alors une famille de vecteurs $(b_1, b_2, \ldots, b_p)$ de $\mathbb{C}[q]^d$ et une famille de polynômes $(P_1(q), P_2(q), \ldots, P_p(q))$ telles que :*

$$P_1(q) \mid P_2(q) \mid \cdots \mid P_p(q)$$

$$(P_1(q)b_1(q), P_2(q)b_2(q), \ldots, P_p(q)b_p(q)) \qquad \textit{est une base du module } A$$

$$\forall z \in \mathbb{C}, \qquad r = \max_{1 \leqslant i \leqslant p}\{i | P_i(z) \neq 0\}$$

$$(P_1(z)b_1(z), P_2(z)b_2(z), \ldots, P_r(z)b_r(z)) \qquad \textit{is a basis of } A_z$$

Démonstration. Les hypothèses sont suffisamment fortes pour que le module A possèdent une base (naturellement finie). En calculant une forme normale de Smith de la matrice formée par cette base, on construit les deux familles attendues dans le résultat. En utilisant le fait que la spécialisation est surjective, on obtient aussi le résultat portant sur toutes les spécialisations. □

Corollaire 5.6.5. *Soit W un groupe affine de Weyl. Soit $T_1(q), T_2(q), \ldots, T_n(q), T_0(q)$ les générateurs formels de $cl(H(W)(q))$. Soit D la matrice diagonale d'une forme normale de Smith d'une matrice formée par une famille génératrice sur $\mathbb{C}[q]$ (en tant que module) de l'algèbre engendrée par les $\{T_i\}_{0 \leqslant i \leqslant n}$. Une racine de l'unité z est mauvaise (i.e. $cl(H(W)(z))$ présente une chute de dimension) si et seulement si z est racine d'un des polynômes diagonaux de D.*

Dans `Sage`, les formes normales de Smith sont implantées mais ne sont pas optimisées pour tous les anneaux principaux. Pour le type A_1, le résultat est instantané ; pour le type A_2, l'obtention du résultat prend plusieurs heures.

5.6.3 Algorithmes, implantation et complexité

Les commandes explicites, présentées dans les sections précédentes, pour tester si une racine de l'unité est bonne ou mauvaise, proviennent de l'intégration dans `Sage` de nouveaux modules sur les systèmes de racines. L'auteur a, par ailleurs, participé à l'élaboration d'une partie de ce code. Les calculs présentés dans les tables 5.5 5.6 5.7 ont nécessité des développements plus spécifiques pour contrer la lenteur de *gap.Subalgebra* avec de grosses entrées et des racines de l'unité de grand ordre. Comme exemple, nous présentons le pseudo code pour ce problème 5 ainsi que sa complexité.

Algorithm 5 algorithm by length

```
AlgorithmByLength(Generators) :
Elements ← {Identity}
NewElements ← ∅
Basis ← {Identity}
while Elements ≠ ∅ do
  for Element ∈ Elements do
    for Generator ∈ Generators do
      Operator = Element ∗ Generator
      if Operator ∉ VectorSpace(Basis) then
        NewElement ← NewElements ∪ {Operator}
        Basis ← GaussReduction(Basis ∪ {Operator})
      end if
    end for
  end for
  print NumberOfElement(Basis)
  Elements ← NewElements
  NewElement ← ∅
end while
```

Soit $\mathring{W}$ un groupe affine de Weyl, et $|\mathring{W}|$ le cardinal de $\mathring{W}$. L'algèbre de Hecke groupe est une sous-algèbre de $End(\mathbb{C}[\mathring{W}])$ qui est de dimension $|\mathring{W}|^2$. Tester pour savoir si z est une racine bonne ou mauvaise nécessite de faire une réduction de Gauss dans cet espace, globalement et par excès, c'est une complexité en $O(|\mathring{W}|^3)$ opérations arithmétiques.

Pour z une racine de l'unité, l'anneau des coefficients est un corps cyclotomique dont la dimension sur $\mathbb{Q}$ est donné par l'indicatrice d'Euler $\phi(n)$. Ceci induit un autre facteur de complexité lorsque l'ordre n de la racine grandit. Il y a aussi une explosion arithmétique des pivots difficiles à évaluer (les calculs étant effectués sur $\mathbb{Q}(\zeta_n)$ avec ζ_n une racine primitive n-ième de l'unité, les numérateurs et dénominateurs des pivots deviennent rapidement très imposants).

TABLE 5.4 – Complexité de la taille de la réponse au pire en nombre de rationnels à calculer

	A_1	A_2	A_3	A_4	B_2	B_3	C_2	C_3	G_2
Cardinal	2	6	24	120	8	48	8	48	12
Taille	12	684	121.536	52.574.400	2.112	1.886.976	2.112	1.886.976	10.512

5.7 Exploration informatique

5.7.1 Détails de l'exploration informatique

Les calculs ont été exécutés sur un Apple Macbook santa rosa 4, 1 utilisant Ubuntu 9.10 (Karmic). L'ordinateur est équipé d'un processeur double coeur à 2, 40 GHz. Sur ce système, nous avons utilisé `Sage` [S+09] version 4.3 ainsi que la boite à outils Sage-Combinat [SCc08].

5.7.2 Tables de résultats

La première table 5.5 présente des séquences de dimensions des résidus au niveau zéro des algèbres de Hecke affines spécialisées en quelques nombres complexes algébriques. Une colonne intitulée "$W \leqslant n$" signifie que l'algorithme a calculé la dimension du sous espace vectoriel engendré par tous les produits de longueur au plus n sur l'alphabet formé par les générateurs.

TABLE 5.5 – Dimension des sous-espace vectoriels engendrés par les produits de longueur au plus n pour les types de Cartan type A_3

A_3	Dimension								
	Full Space	$W \leqslant 1$	$W \leqslant 2$	$W \leqslant 3$	$W \leqslant 4$	$W \leqslant 5$	$W \leqslant 6$	$W \leqslant 7$	$W \leqslant 8$
q = 2	**211**	5	15	35	69	121	181	207	211
q = 0	**211**	5	15	35	69	121	181	207	211
q = 1	**24**	5	15	23	24	.	.	.	.
q = -1	**125**	5	15	33	59	89	115	125	.
q = $\sqrt[3]{1}$	**152**	5	15	35	68	112	139	149	152
q = i	**112**	5	15	33	58	86	108	112	.
q = $\sqrt[5]{1}$	**211**	5	15	35	69	121	181	207	211
q = $\sqrt[6]{1}$	**211**	5	15	35	69	121	181	207	211
q = $\sqrt[7]{1}$	**211**	5	15	35	69	121	181	207	211
q = $\sqrt[8]{1}$	**211**	5	15	35	69	121	181	207	211

TABLE 5.6 – Dimensions aux racines de l'unité

Spécialisation	A_1	A_2	A_3	A_4	B_2	B_3	C_2	C_3	G_2
q = 2	3	19	211	tl	33	tl	33	tl	73
q = 0	3	19	211	3651	33	819	33	819	73
q = 1	2	6	24	120	8	48	8	48	12
q = -1	**2**	**18**	**125**	tl	**16**	**341**	**16**	**268**	**36**
q = $\sqrt[3]{1}$	3	**11**	**152**	tl	**26**	tl	**26**	tl	**50**
q = i	3	19	**112**	tl	**22**	tl	**22**	tl	**46**
q = $\sqrt[5]{1}$	3	19	211	tl	33	tl	33	tl	**62**
q = $\sqrt[6]{1}$	3	19	211	tl	33	tl	33	tl	**61**
q = $\sqrt[7]{1}$	3	19	211	tl	33	tl	33	tl	73
q = $\sqrt[8]{1}$	3	19	211	tl	33	tl	33	tl	73

TABLE 5.7 – Borne théorique et plus grande hauteur k telle que la racine k-ième est mauvaise.

Cartan type	Degree bound	k such that z is bad
A_1	2	$\{1, 2\}$
A_2	4	$\{1, 2, 3\}$
A_3	6	$\{1, 2, 3, 4\}$
A_4	8	$\{1, 2, \ldots\}$
A_n	$2n$	$\{1, \ldots\}$
B_2	6	$\{1, 2, 3, 4\}$
B_3	10	$\{1, 2, \ldots\}$
B_n	$2(2n-1)$	$\{1, \ldots\}$
C_2	6	$\{1, 2, 3, 4\}$
C_3	10	$\{1, 2, \ldots\}$
C_n	$2(2n-1)$	$\{1, \ldots\}$
G_2	10	$\{1, 2, 3, 4, 5, 6\}$

5.7.3 Exploitations, Discussion

La lecture des tables 5.5, 5.6 and 5.7 nous suggère l'énonciation des conjectures suivantes.

Conjecture 5.7.1. *Pour tout entier $n \geqslant 1$, -1 est une mauvaise racine de l'unité pour le type de Cartan A_n*

Conjecture 5.7.2. *Soit un entier $n \geqslant 1$. Soit $\mathring{W}$ le groupe de Weyl de type de Cartan A_n (i.e. le groupe symétrique $\mathfrak{S}_{n+1}$). Soit M l'opérateur de $End(\mathbb{C}[\mathring{W}])$ définit comme :*

$$M = \sum_{s \in S_{n+1}} s$$

Alors $M \in H\mathring{W}$ mais $M \notin cl(H(W)(-1))$ où W est le groupe affine de type de Cartan A_n.

Problème 5.7.3. *Pour tout type de Cartan, les différents ordres des mauvaises racines de l'unité pour le groupe associé forment un intervalle d'entiers de la forme $\{1, 2, \ldots, k\}$ pour un certain entier $k \geqslant 1$.*

Définition 5.7.4. *Soit A une algèbre définie par une famille finie de générateurs et i un entier positif. On définit $A_{\leqslant i}$ le sous-espace vectoriel de A engendré comme module par tous les produits de longueur au plus i sur les générateurs.*

Conjecture 5.7.5. *Soit $\mathring{W}$ un groupe de Weyl fini et W sa version affine correspondante. Soit q un paramètre formel et z un nombre complexe. Si*

$$dim_{\mathbb{C}}(cl(H(W)(z))_{\leqslant i}) < dim_{\mathbb{C}(q)}(cl(H(W)(q))_{\leqslant i})$$

alors

$$dim_{\mathbb{C}}(cl(H(W)(z))) < dim_{\mathbb{C}(q)}(cl(H(W)(q)))$$

Remarque 5.7.6. *Prouver cette dernière conjecture fournirait un critère de terminaison pour achever les calculs prématurément. L'algorithme par longueur est d'autant plus rapide que la longueur des mots sur les générateurs est réduite. Par exemple, en type A_3 5.5, il serait suffisant de calculer la dimension jusqu'à la longueur trois, ce qui nécessite une seconde au lieu de la minute nécessaire pour terminer le calcul.*

Remarque 5.7.7. *Cette dernière conjecture 5.7.5 est fausse en générale. Elle n'est pas vérifiée pour une algèbre finie définie par générateurs. Toutefois, dans le cadre de ces algèbres de Hecke groupe, tous nos résultats la vérifient.*

Considérons la $\mathbb{C}(q)$-algèbre $A(q)$ engendrée par :

$$\left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & q+1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} q-1 & q+1 & q-1 \\ 0 & q+1 & q-1 \\ 0 & q+1 & 0 \end{pmatrix}\right]$$

et sa spécialisation $A(1)$.

Comme annoncé plus haut, $A_{\leqslant i}(q)$ (resp. $A_{\leqslant i}(1)$) est le sous espace de $A(q)$ (resp. $A(1)$) engendré par tous les produits de longueur au plus i sur les générateurs. Ainsi, les séquences de dimension de $A_i(q)$ et $A_i(1)$ sont données par :

	$A_{\leqslant 0}(q)$	$A_{\leqslant 1}(q)$	$A_{\leqslant 2}(q)$	$A_{\leqslant 3}(q)$	$A_{\leqslant 4}(q)$
dimension	1	3	7	9	.
	$A_{\leqslant 0}(1)$	$A_{\leqslant 1}(1)$	$A_{\leqslant 2}(1)$	$A_{\leqslant 3}(1)$	$A_{\leqslant 4}(1)$
dimension	1	3	6	8	9

On remarque que $dim(A_{\leqslant 1}(1)) < dim(A_{\leqslant 1}(q))$, alors que $dim(A(1)) = dim(A(q))$.

La question reste, tout de même, ouverte dans ce contexte d'algèbre de Hecke.

Conjecture 5.7.8. *Soit $\mathring{W}$ un groupe de Weyl fini et $\mathring{W}_I$ un sous groupe parabolique de $\mathring{W}$. Soit W (resp. W_I) la version affine associée au groupe $\mathring{W}$ (resp. $\mathring{W}_I$). Pour tout nombre complexe z, on a :*

$$dim H\mathring{W}_I - dim(cl(H(W_I)(z))) \leqslant dim H\mathring{W} - dim(cl(H(W)(z)))$$

Ainsi, si z est une mauvaise racine pour $\mathring{W}_I$, z est alors aussi mauvaise pour $\mathring{W}$.

Annexe

Chapitre 6

Polynômes harmoniques déformés pour les groupes de réflexions complexes

Ce travail, écrit en langue anglaise, fut effectué en collaboration avec François Bergeron et Nicolas Thiéry et fut présenté lors de la conférence «Formal Power Series and Algebraic Combinatorics 2011» [BBT11].

6.1 Introduction

The aim of this work is to give support to an extension and a generalization of the main conjecture of [HT04], to the diagonal case as well as to the context of finite complex reflection groups. This is stated explicitly in the new Conjecture 6.1.2 below, after a few words concerning notations and a description of the overall context.

Let X denote a $\ell \times n$ matrix of variables

$$X := (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\,,$$

with each of the columns $\mathbf{x}_j = (x_{ij})_{1 \leqslant i \leqslant \ell}$ containing ℓ variables. For any fixed i (a row of X), we say that the variables $x_{i1}, x_{i2}, \ldots, x_{in}$ form a *set of variables* (the i^{th} set), and thus X consists in ℓ sets of n variables. For $\mathbf{d} \in \mathbb{N}^\ell$, we set

$$|\mathbf{d}| := d_1 + d_2 + \cdots + d_\ell \qquad \text{and} \qquad \mathbf{d}! := d_1! d_2! \cdots d_\ell!\,,$$

and write $\mathbf{x}_j^{\mathbf{d}}$ for the column monomial of *degree* $\mathbf{x}_j^{\mathbf{d}}$:

$$\mathbf{x}_j^{\mathbf{d}} := \prod_{i=1}^{\ell} x_{ij}^{d_i}\,.$$

The ground field $\mathbb{K}$ is assumed to be of characteristic zero and, whenever needed, to contain roots of unity and/or a parameter q ; typically, $\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{C}(q)$, although algebraic or transcendental extensions of $\mathbb{Q}$ are better suited for some of the computer calculations. The parameter q is called formal if it is transcendental over $\mathbb{Q}$.

Let W be a complex reflection group of rank n. Elements of W may be thought of as $n \times n$ matrices with complex entries. The *diagonal action* of W on a polynomial $f(X)$ is defined, for $w \in W$, by :

$$w \cdot f(X) := f(X\, w), \tag{6.1.1}$$

where $X\, w$ stands for matrix multiplication. In other words, W acts in a similar "diagonal" manner on each set of variables in X. A polynomial is *diagonally W-invariant* if

$$w \cdot f(X) = f(X), \qquad \text{for all} \quad w \in W\,.$$

We denote by $\mathcal{I}_W^{(\ell)}$ the ideal generated by constant-term-free diagonally W-invariant polynomials.

For each of the variables $x_{ij} \in X$, there is an associated partial derivation denoted here by $\partial_{x_{ij}}$ or ∂_{ij} for short. For a polynomial $f(X)$, let $f(\partial_{\checkmark})$ stand for the differential operator obtained by replacing variable in $\checkmark$ by the corresponding derivation in $\partial_{\checkmark}$. The space $\mathcal{H}W^{(\ell)}$ of *diagonally W-harmonic polynomials* (or *harmonic polynomials* for short) is then defined as the set of the polynomials $g(X)$ that satisfy all of the linear partial differential equations

$$f(\partial_{\checkmark})(g(X)) = 0, \qquad \text{for} \quad f(X) \in \mathcal{I}_W^{(\ell)}\,. \tag{6.1.2}$$

In the following, we first restrict ourselves to the complex reflection groups $W = G(m, n)$, for $m, n \in \mathbb{N}$, and then extend our discussion to the subgroups $G(m, p, n)$. Recall that the *generalized symmetric group* $G(m, n)$ may be constructed as the group of $n \times n$ matrices having exactly one non zero entry in each row and each column, whose value is a m-th root of unity. Since the cases $\ell = 1$ and $W = \mathfrak{S}_n$ have been extensively considered previously (see [Ber09]), we write for short $\mathcal{H}W = \mathcal{H}W^{(1)}$, $\mathcal{H}_n^{(\ell)} = \mathcal{H}\mathfrak{S}_n^{(\ell)}$, and $\mathcal{H}_n = \mathcal{H}\mathfrak{S}_n$.

The ring $\mathbb{K}[X]^W$ of diagonally W-invariant polynomials for $W = G(m, n)$ is generated by *polarized* powersums, this is to say the polynomials

$$P_{\mathbf{d}} = \sum_{j=1}^{n} \mathbf{x}_j^{\mathbf{d}}\,,$$

for $|\mathbf{d}| = m\, k$, with $1 \leqslant k \leqslant n$. Let us write $D_{\mathbf{d}}$ for the operator $P_{\mathbf{d}}(\partial_{\checkmark})$:

$$D_{\mathbf{d}} = \sum_{j=1}^{n} \mp_j^{\mathbf{d}}\,,$$

where

$$\mp_j^{\mathbf{d}} := \partial_{1j}^{d_1} \partial_{2j}^{d_2} \cdots \partial_{\ell j}^{d_\ell}\,.$$

Then, the space $\mathcal{H}W^{(\ell)}$ is the intersection of the kernels of all the operators $D_{\mathbf{d}}$, for $|\mathbf{d}| = m\, k$, with $1 \leqslant k \leqslant n$. The space $\mathcal{H}W^{(\ell)}$ is graded by (multi-)degree, and thus decomposes as a direct sum

$$\mathcal{H}W^{(\ell)} = \bigoplus_{\mathbf{d} \in \mathbb{N}^\ell} \mathcal{H}W_{\mathbf{d}}^{(\ell)}\,,$$

of its homogeneous components of degree $\mathbf{d}$. Recall that $f(X)$ is homogeneous of degree $\mathbf{d}$, if and only if we have

$$f(\mathbf{t}\, X) = \mathbf{t}^{\mathbf{d}}\, f(X)\,,$$

where $\mathbf{t}\,X$ stands for the multiplication of the matrix X by the diagonal matrix

$$\begin{pmatrix} t_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & t_\ell \end{pmatrix}.$$

The *Hilbert series of the space* $\mathcal{H}W^{(\ell)}$ is defined as

$$\mathcal{H}W^{(\ell)}(\mathbf{t}) := \sum_{\mathbf{d}\in\mathbb{N}^\ell} \dim(\mathcal{H}W^{(\ell)}_{\mathbf{d}})\,\mathbf{t}^{\mathbf{d}}\,.$$

It is well known that, for $\ell = 1$, the graded space $\mathcal{H}W$ is isomorphic, as a W-module, to the graded regular representation of W. In particular, its Hilbert series is given by the formula

$$\mathcal{H}W(t) = \prod_{k=1}^{n} \frac{t^{k\,m}-1}{t-1}\,.$$

Our specific story starts with a q-deformation of the polarized powersums :

$$P_{q,\mathbf{d}} := \sum_{j=1}^{n} \mathbf{x}_j^{\mathbf{d}}\,(1 + q\,(x_{1j}\partial_{1j} + \ldots + x_{nj}\partial_{nj})). \tag{6.1.3}$$

and the corresponding q-deformation of the operators $D_{\mathbf{d}} = P_{\mathbf{d}}(\partial_{\swarrow})$ above :

$$D_{q,\mathbf{d}} := \sum_{j=1}^{n} (1 + q\,(x_{1j}\partial_{1j} + \ldots + x_{nj}\partial_{nj}))\;\mp_j^{\mathbf{d}}\,. \tag{6.1.4}$$

An homogeneous polynomial $f(X)$ is said to be *diagonally W,q-harmonic* (or q-harmonic for short) if

$$D_{q,\mathbf{d}}\,f(X) = 0, \tag{6.1.5}$$

for all $\mathbf{d} \in \mathbb{N}^\ell$ such that $|\mathbf{d}|$ is divisible by m. The W-module of all q-harmonic polynomials is henceforth denoted by $\mathcal{H}W_q^{(\ell)}$, and thus by $\mathcal{H}_{n,q}^{(\ell)}$ when $W = \mathfrak{S}_n$. Our aim here is to discuss for which q the following assertion holds.

Assertion 6.1.1. *$\mathcal{H}W_q^{(\ell)}$ is isomorphic (as a graded W-module) to $\mathcal{H}W^{(\ell)}$.*

Conjecture 6.1.2. *Let $W = G(m,n)$, $\ell \in \mathbb{N}$. Then, Assertion 6.1.1 holds for q a formal parameter. In particular $\mathcal{H}W_q^{(\ell)}(\mathbf{t}) = \mathcal{H}W^{(\ell)}(\mathbf{t})$.*

This conjecture is an extension of the main conjecture of [HT04] (case $W = \mathfrak{S}_n$, $\ell = 1$), which itself is a q-analogue of a conjecture of Wood [Woo98, Woo01] on the "hit polynomials" for the *rational Steenrod algebra* $\mathcal{S} := \mathbb{K}[P_{1,d} \mid d \geqslant 1]$. Beside the extensive computer exploration and results reported on in [HT04] for $W = \mathfrak{S}_n$ and $\ell = 1$, our supporting evidence for this conjecture, includes the following results :

(1) Applying a classical specialization argument at $q = 0$ (see e.g. [HT04]), gives that $\dim \mathcal{H}W_q^{(\ell)} \leqslant \dim \mathcal{H}W^{(\ell)}$ (also homogeneous component by homogeneous component). Furthermore, equality holds if and only if Conjecture 6.1.2 does.

(2) Conjecture 6.1.2 holds for all groups $G(m,2)$ for $\ell = 1$ (see Section 6.5), as well as for all ℓ when $m \leqslant 5$. For example, with $W = G(3,2)$, we get

$$\begin{aligned}\mathcal{H}W_q^{(\ell)}(\mathbf{t}) &= 1 + 2\,h_1(\mathbf{t}) + 2\,h_2(\mathbf{t}) + h_1^2(\mathbf{t}) + h_3(\mathbf{t}) + 2\,h_2(\mathbf{t})h_1(\mathbf{t}) \\ &\quad + 2\,h_4(\mathbf{t}) + h_2^2(\mathbf{t}) + 3\,h_5(\mathbf{t}) + 2\,h_6(\mathbf{t}) + h_7(\mathbf{t}).\end{aligned}$$

(3) Conjecture 6.1.2 holds for all ℓ, in the case $W = \mathfrak{S}_n = G(1,n)$ for $n \leqslant 4$. For example, we get the Hilbert series

$$\mathcal{H}_{3,q}^{(\ell)}(\mathbf{t}) = 1 + 2\,h_1(\mathbf{t}) + h_{11}(\mathbf{t}) + h_2(\mathbf{t}) + h_3(\mathbf{t})\,.$$

(4) There seems to be an analogue of Conjecture 6.1.2 for the subgroups $G(m,p,n)$ of $G(m,n)$ (see Section 6.2); in particular, Conjecture 6.1.2 holds for $n = 2$ (including the dihedral groups $G(m,m,2)$) when $\ell = 1$ (see Section 6.5), and for any ℓ for small values of m, p, n.

Another interesting feature of the space $\mathcal{H}W_q^{(\ell)}$, is that it may be characterized as the intersection of the kernels of a much smaller family of operators than the set

$$\{D_{q,\mathbf{d}} \mid |\mathbf{d}| = m\,k, \quad 1 \leqslant k \leqslant n\}. \tag{6.1.6}$$

Indeed, a straightforward calculation shows that the usual Lie-bracket relation between the generators of the rational Steenrod algebra generalize naturally :

$$[D_{q,\mathbf{d}}, D_{q,\mathbf{d}'}] = q(|\mathbf{d}| - |\mathbf{d}'|)\,D_{q,\mathbf{d}+\mathbf{d}'}. \tag{6.1.7}$$

An efficient way to setup this calculation is to let both sides act on the generating function for all monomials, namely the formal series

$$\exp(Z.X) = \sum_{\mathbf{d}\in\mathbb{N}^{\ell\times n}} \mathbf{x}^{\mathbf{d}}\frac{\mathbf{z}^{\mathbf{d}}}{\mathbf{d}!}\,,$$

where Z stands for a matrix of variables just as X does, and $Z.X := \sum_{ij} z_{ij}x_{ij}$. It follows from (6.1.7) that a polynomial is in the kernel of $D_{q,\mathbf{d}+\mathbf{d}'}$, whenever it lies in the kernels of both $D_{q,\mathbf{d}}$ and $D_{q,\mathbf{d}'}$. From this, we can immediately deduce that

Proposition 6.1.3. *The space of q-harmonic polynomials for $W = G(m,n)$ can be obtained as*

$$\mathcal{H}W_q^{(\ell)} = \bigcap_{|\mathbf{d}|=m \text{ or } 2\,m} \mathrm{Ker}(D_{q,\mathbf{d}})\,.$$

For example, when $\ell = 1$, and as already noted in [HT04] in the case $W = \mathfrak{S}_n$, the space $\mathcal{H}W_q$ is defined by just two linear differential equations :

$$\mathcal{H}W_q = \mathrm{Ker}(D_{q,m}) \cap \mathrm{Ker}(D_{q,2\,m})\,.$$

Similarly, when $\ell = 2$, the space $\mathcal{H}W_q^{(2)}$ is the intersection of the kernels of only five operators :

$$D_{q,(1,0)}, \qquad D_{q,(2,0)}, \qquad D_{q,(1,1)}, \qquad D_{q,(0,1)}, \quad \text{and} \quad D_{q,(0,2)}\,.$$

This is striking because, assuming that Conjecture 6.1.2 holds, a direct calculation of this joint kernel and a specialization at $q = 0$ would yield back the famous space $\mathcal{H}_n^{(2)}$ of diagonally harmonic polynomials. Yet, even if the mysterious structure of $\mathcal{H}_n^{(2)}$ has been extensively studied

in the past 20 years (see [Hai03]), no nice Gröbner basis for the ideal $\mathcal{I}_W^{(2)}$ is known, even for $W = \mathfrak{S}_n$.

It is also noteworthy that systematic variations on the main conjecture in [HT04] have been extensively studied in [BGW10]. In an upcoming work, we plan to describe how these variations may be adapted to the context of the reflection groups considered here, including the diagonal point of view. In particular, since there is a close tie (*loc. sit.*) between the case $\ell = 1$, with $W = \mathfrak{S}_n$, and Wood's conjecture (stated in [Woo98] or [Woo01]), we also plan to analyze how to generalize it to our new expanded context.

6.2 Deformed harmonic polynomials for $G(m, p, n)$

This section presents work in progress toward generalizing the construction of q-harmonic polynomials, and Conjecture 6.1.2, to all finite complex reflection groups. For simplicity, we restrict ourselves to a single set of variables : namely $\ell = 1$. However, computer calculations suggest that the extension to the diagonal case is straightforward.

Recall that all but a small number of finite complex reflection groups are part of an infinite family of natural subgroups of the generalized symmetric groups which we consider now. For $m, n \in \mathbb{N}$, let p be a divisor of m. Then, the complex reflection group $G(m, p, n)$ is defined as :

$$G(m, p, n) := \{g \in G(m, n) \mid \det g^{m/p} = 1\}.$$

In particular, setting $p = 1$, we get back $G(m, n) = G(m, 1, n)$. Recall, for example, that the classical dihedral groups correspond to the family $G(m, m, 2)$.

The invariant ring for $W = G(m, p, n)$ is obtained by adjoining $e_n^{m/p}$ to the invariant ring of $G(m, n)$, with $e_n = e_n(\mathbf{x}) := x_1 \cdots x_n$ standing for the product of the variables. It is well known that the invariant ring $\mathbb{K}[\mathbf{x}]^W$ may then be described as the graded free commutative algebra :

$$\mathbb{K}[\mathbf{x}]^W = \mathbb{K}[p_m, \ldots, p_{(m-1)n}, e_n^{m/p}],$$

with $\deg(p_k) = k$ and $\deg(e_n^{m/p}) = nm/p$. Notice the necessary suppression of the generator p_{mn} for this presentation to be free.

The choice of a canonical q-analogue of $e_n^{m/p}$ does not appear to be straightforward. Indeed, and as far as we know (see the discussion in [HT04, Section 7.1]), there is no natural analogue of the elementary symmetric polynomial inside the rational Steenrod algebra $\mathcal{S}_q$. Besides, experienced gained in [BGW10] suggests that (generically) any choice of q-analogue would give an isomorphic space. We therefore take the simplest option, which is to not deform $e_n^{m/p}$ at all. Hence, for $W = G(m, p, n)$, we define the *q-deformed rational W-Steenrod algebra* as

$$\mathcal{S}_q^W := \mathbb{K}[P_{q,m}, \ldots, P_{q,mn}, e_n^{m/p}].$$

Accordingly, we obtain the graded space $\mathcal{H}W_q$ of the q-harmonic polynomials for $W = G(m, p, n)$, just as previously. Specifically, writing ε for the operator $e_n(\partial_\checkmark)$, $\mathcal{H}W_q$ is the intersection of $\mathcal{H}G(m, n)_q$ and $\mathrm{Ker}(\varepsilon^{m/p})$. A natural question here is to ask whether Conjecture 6.1.2 holds for $G(m, p, n)$. We will show in Section 6.5 that it does for $n = 2$ and $\ell = 1$.

A first step to confirm the choice of $e_n^{m/p}$ would be to prove the following conjecture.

Conjecture 6.2.1. *The q-harmonic polynomials for $G(m,1,n)$, as defined above, coincide with those for $G(m,n)$.*

An equivalent but more concrete condition is that no q-harmonic polynomial for $G(m,n)$ shall contain a monomial divisible by e_n^m. This property holds for $n=2$, and all q-harmonic polynomials for $\mathfrak{S}_n$ calculated in [HT04].

In fact, we expect $\mathcal{H}G(m,n)_q$ to decompose as a direct sum of m layers $L_0(q)\oplus\cdots\oplus L_{m-1}(q)$ such that all the elements of $L_k(q)$ are divisible by e_n^k but not by e_n^{k+1}, as in Figure 6.5. Furthermore, ε (depicted by the gray down arrows in this figure) would be an isomorphism from $L_k(q)$ to $L_{k-1}(q/(1+q))$, the change of q being due to the equation

$$e_n(\partial_\checkmark)D_{q,k}=(1+q)D_{\frac{q}{1+q},k}e_n(\partial_\checkmark)\,. \tag{6.2.1}$$

In that case, the q-harmonic polynomials for $G(m,p,n)$ would be given by

$$\mathcal{H}G(m,p,n)_q=L_0(q)\oplus\cdots\oplus L_{m/p-1}(q)\,. \tag{6.2.2}$$

For example, in Figure 6.5, the q-harmonic polynomials for the dihedral group $G(4,1,2)$ are given by $\mathcal{H}G(4,1,2)_q=L_0(q)$. Similarly, $\mathcal{H}G(4,2,2)_q=L_0(q)\oplus L_1(q)$. We expect that, in general, the space $\mathcal{H}G(m,n)_q$ consists of m copies of $\mathcal{H}G(m,n,1)_q$, that may be constructed by "lifting back" through ε.

This further suggests that the lack of operators commuting appropriately with the action of the rational Steenrod algebra, as reported in [HT04, Section 7] can be circumvented by allowing q to change during the commutation. For example, for $m=\ell=1$, the harmonic polynomials are usually constructed from the Vandermonde determinant by iterative applications of operators ∂_i ; it would be worth finding analogues of those operators which would construct new q-harmonic polynomials from q'-harmonic polynomials for some other q'.

6.3 Inflating q-harmonic polynomials from $\mathfrak{S}_n$ to $G(m,n)$ and $G(m,m,n)$

In this section, we do some preliminary steps in the following direction.

Problème 6.3.1. *Assume that a basis of the q-harmonic polynomials for $\mathfrak{S}_n$ is given. Is it possible to construct from it a basis of the q-harmonic polynomials for $G(m,n)$? for $G(m,m,n)$?*

Beware that the analogous problem for constructing diagonally q-harmonic polynomials from q-harmonic polynomials is already hard at $q=0$.

To start with, any q-harmonic polynomial for $W=\mathfrak{S}_n$ remains q-harmonic for $G(m,n)$. It also remains q-harmonic for $G(m,p,n)$ as soon as Conjecture 6.2.1 holds for $\mathfrak{S}_n$. We now construct more q-harmonic polynomials by inflating those of $\mathfrak{S}_n$. To this end, we consider the inflation algebra morphism and its analogue (which is just a linear morphism) on the dual basis :

$$\phi_r:\begin{cases}\mathbb{K}[X] & \hookrightarrow\!\!\!\!\rightarrow\ \mathbb{K}[X^r]\\ \mathbf{x}^{\mathbf{d}} & \mapsto\ \mathbf{x}^{r\mathbf{d}}\end{cases}\qquad\qquad \overline{\phi}_r:\begin{cases}\mathbb{K}[X] & \hookrightarrow\!\!\!\!\rightarrow\ \mathbb{K}[X^r]\\ \mathbf{x}^{(\mathbf{d})} & \mapsto\ \mathbf{x}^{(r\mathbf{d})}\end{cases}, \tag{6.3.1}$$

where, by a slight notational abuse, X^r stands for the matrix of the r-th powers of the variables and $\mathbf{x}^{(\mathbf{d})}:=\frac{1}{\mathbf{d}!}\mathbf{x}^{\mathbf{d}}$.

Proposition 6.3.2. *Let $W = G(m, n)$ and r be a divisor of m. Then, the morphism $\overline{\phi}_r$ restricts to a graded $\mathfrak{S}_n$-module embedding (resp. isomorphism if $r = m$) from $\mathcal{H}\mathfrak{S}_{nq}$ to $\mathcal{H}W_{q/m} \cap \mathbb{K}[X^r]$, up to an appropriate r-scaling of the grading. The statement extends to any $G(m, p, n)$ as soon as Conjecture 6.2.1 holds for $\mathfrak{S}_n$.*

The first step toward this proposition is to define an appropriate inflation on the q-rational Steenrod algebra. Note that the operators $P_{q,k}$ live inside the subalgebra $\mathbb{K}[\checkmark \cdot \partial_\checkmark, \checkmark]$ of the Weyl algebra, where $\checkmark \cdot \partial_\checkmark$ denotes the matrix of the Euler operators $x\partial_x$ for $x \in X$. The only non-trivial brackets in this algebra are $[x\partial_x, x] = x$, for $x \in X$, from which it follows that $[x\partial_x, x^k] = kx^k$. Similarly, the operators $D_{q,k}$ live inside the subalgebra $\mathbb{K}[XPX, \partial_\checkmark]$, with analogous relations.

Remarque 6.3.3. *Fix $r \in \mathbb{N}$. Then, the two mappings*

$$x\partial_x \mapsto 1/r(x\partial_x), \quad x \mapsto x^r, \text{ for } x \in X \qquad \text{and} \qquad x\partial_x \mapsto 1/r(x\partial_x), \quad \partial_x \mapsto \partial_x^r, \text{ for } x \in X$$

extend respectively to algebra isomorphisms

$$\Phi_r : \mathbb{K}[\checkmark \cdot \partial_\checkmark, \checkmark] \hookrightarrow\!\!\!\!\rightarrow \mathbb{K}[\checkmark \cdot \partial_\checkmark, \checkmark^m] \qquad \text{and} \qquad \overline{\Phi}_r : \mathbb{K}[\checkmark \cdot \partial_\checkmark, \partial_\checkmark] \hookrightarrow\!\!\!\!\rightarrow \mathbb{K}[\checkmark \cdot \partial_\checkmark, \partial_\checkmark^m] .$$

Furthermore, those isomorphisms are compatible with the action on inflated polynomials : for $f \in \mathbb{K}[X]$ and F in $\mathbb{K}[\checkmark \cdot \partial_\checkmark, \checkmark]$ (resp. in $\mathbb{K}[\checkmark \cdot \partial_\checkmark, \partial_\checkmark]$), we have

$$\Phi_r(F)(\phi_r(f)) = \phi_r(F(f)) \qquad \text{and} \qquad \overline{\Phi}_r(F)(\overline{\phi}_r(f)) = \overline{\phi}_r(F(f)) . \tag{6.3.2}$$

Using this remark, a straightforward calculation shows that :

$$\Phi_r(P_{q,\mathbf{d}}) = P_{q/r,r\mathbf{d}} \qquad \text{and} \qquad \overline{\Phi}_r(D_{q,\mathbf{d}}) = D_{q/r,r\mathbf{d}} . \tag{6.3.3}$$

This readily implies that Φ_m restricts to an isomorphism from the q-rational Steenrod algebra for $\mathfrak{S}_n$ to that for $G(m, n)$. Computer exploration suggests that the Gröbner basis for the right ideal generated by the Steenrod algebra for $G(m, n)$ is simply obtained by inflating that for $\mathfrak{S}_n$. This possibly opens the door for controlling the leading monomials of "q-hit polynomials" for $G(m, n)$ from those for $\mathfrak{S}_n$.

Returning to our main goal, we now have all the ingredients to prove Proposition 6.3.2.

of Proposition 6.3.2. Let $f \in \mathbb{K}[\checkmark]$. Then, using Equation (6.3.2),

$$D_{q/r,\, r\mathbf{d}}(\phi(f)) = \overline{\Phi}_r(D_{q,\mathbf{d}})(\phi(f)) = \phi(D_{q,\mathbf{d}}(f)) .$$

Hence $D_{q/r,r\mathbf{d}}(\phi(f)) = 0$ if and only if $D_{q,\mathbf{d}}(f) = 0$. The statements follows. □

6.4 Singular values

As discussed in [HT04], Assertion 6.1.1 may fail for very specific values of q. In that case, q is said to be *singular*. Computer exploration (see Table A.3 of [HT04]) and the complete analysis of the case $n = 2$ suggested that the only such singular values for $W = \mathfrak{S}_n$ and $\ell = 1$ are negative rational numbers of the form $-a/b$ with $a \leqslant n$. In [DM10] D'Adderio and Moci refined this statement to $a \leqslant n \leqslant b$ (with $q = -a/b$ not necessarily reduced), and proved that all such values are indeed singular by constructing explicit exceptional q-harmonic polynomials.

Proposition 6.4.1. *Let $W = G(m,n)$, $\ell \in \mathbb{N}$, and $q = -a/b$ with $a \leqslant n \leqslant b$, for $a, b \in \mathbb{N}$. Then, q is singular.*

(sketch of proof). Let $f(x_1, \dots, x_n)$ be the q-harmonic polynomial for $\mathfrak{S}_n$ constructed in [DM10]. Then, as stated in Proposition 6.3.2, $f(x_1^m, \dots, x_n^m)$ is a q/m-harmonic for W of high enough degree (as in [DM10]) to disprove the statement of Conjecture 6.1.2. Going from $\ell = 1$ to ℓ arbitrary is then straightforward, since the intersection of $\mathcal{H}W_q^{(\ell)}$ with the polynomial ring in the first set of variables is $\mathcal{H}W_q$. □

It is worth noting that, for $n = 2$, and $\ell = 1$ the singular values are exactly those listed in Proposition 6.4.1 (see Section 6.5). However, at this stage, we lack computer data to extend this to a conjecture for all n and ℓ.

6.5 Complete study for $n = 2$

In this section, we prove Conjecture 6.1.2 for any group $W = G(m,p,2)$ in the case $\ell = 1$. We denote for short the two variables x and y instead of x_1 and x_2. Naturally ∂_x and ∂_y are the corresponding differential operators. We also introduce the following q-analogue of the Pochhammer symbol $(d)_k$:

$$\langle d \rangle_k := d\,(d-1)\cdots(d-k+1)\,(1+q\,(d-k)).$$

Then, for any monomial $x^\alpha y^\beta$, one has :

$$D_{q,k}(x^\alpha y^b) = \langle \alpha \rangle_k\, x^{\alpha-k} y^\beta + \langle \beta \rangle_k\, x^\alpha y^{\beta-k}\,, \tag{6.5.1}$$

which is well defined for any nonnegative numbers α and β, since $\langle \alpha \rangle_k = 0$ whenever $\alpha < k$.

Remarque 6.5.1. *Let $W = G(m,m,2)$ be the dihedral group. Then, the space $\mathcal{H}W_q$ is isomorphic to $\mathcal{H}W$, and in fact coincides with it, if and only if q is not of the form $-1/b$ with $1 \leqslant b \leqslant m$. In that case, it is of dimension $2m$ and a basis is given by*

$$\{1, x, x^2, x^3, \dots, x^{m-1}, x^m - y^m, y^{m-1}, y^{m-2}, \dots, y^2, y\}\,. \tag{6.5.2}$$

Otherwise, the basis of $\mathcal{H}W_q$ contains additionally the monomials x^{b+m} and y^{b+m}, or just the binomial $x^{2m} - y^{2m}$ if $b = m$.

Démonstration. Let $f = f(x,y)$ be an homogeneous q-harmonic polynomial in $\mathbb{K}[x,y]$. It satisfies :

$$D_{q,m}(f) = 0, \qquad D_{q,2m}(f) = 0, \qquad \text{and} \qquad \varepsilon(f) = 0,$$

where $\varepsilon = \partial_x \partial_y$. By the last equation, f is of the form $\lambda x^d + \mu y^d$, and the two other equations rewrite as $(d)_k(\lambda x^{d-km} + \mu y^{d-km})$ for $k = 1, 2$. The statement follows. □

Proposition 6.5.2. *Let $W = G(m,2)$ and $\ell = 1$. Then, the space $\mathcal{H}W_q$ is isomorphic as a graded W-module to $\mathcal{H}W$ if and only if q is not of the form $-a/b$ with $1 \leqslant a \leqslant 2 \leqslant b$, and $a, b \in \mathbb{N}$. In that case, it is of dimension $2m^2$ and a basis is given by*

$$\{x^\alpha y^\beta\}_{\substack{0 \leqslant a < m \\ 0 \leqslant b < m}} \cup \{\langle \beta + m \rangle_m x^{\alpha+m} y^\beta - \langle \alpha + m \rangle_m x^\alpha y^{\beta+m}\}_{\substack{0 \leqslant \alpha < m \\ 0 \leqslant \beta < m}}\,. \tag{6.5.3}$$

Démonstration. As suggested by Equation (6.5.1), the implicit combinatorial ingredient is the length of the longest string $\dots, x^{\alpha-m}y^{\beta+m}, x^{\alpha}y^{\beta}, x^{\alpha+m}y^{\beta-m}, \dots$ containing any given monomial.

Obviously, whenever $\alpha < m$ and $\beta < m$, the monomial $x^{\alpha}y^{\beta}$ is killed by both operators $D_{q,m}$ and $D_{q,2\,m}$, and is therefore q-harmonic. This gives the first m^2 monomials in (6.5.3). Using Equation 6.5.1, a direct calculation shows that the remaining m^2 binomials

$$\langle \beta + m \rangle_m x^{\alpha+m}y^{\beta} - \langle \alpha + m \rangle_m x^{\alpha}y^{\beta+m}$$

for $\alpha < m$ and $\beta < m$ are also q-harmonic.

We now consider a monomial $x^{\alpha'}y^{\beta'}$ that does not appear in any of the q-harmonic polynomials of (6.5.3), and prove that it cannot appear in any other q-harmonic polynomial. It is straightforward that we can choose α and β such that

$$x^{\alpha'}y^{\beta'} \in \{x^{\alpha+m}y^{\beta-m}, x^{\alpha}y^{\beta}, x^{\alpha-m}y^{\beta+m}\}\,.$$

Let $h = c_1 x^{\alpha+m}y^{\beta-m} + c_2 x^{\alpha}y^{\beta} + c_3 x^{\alpha-m}y^{\beta+m} + \cdots$ be a q-harmonic polynomial. Then,

$$0 = D_{q,m}(h)_{|x^{\alpha}y^{\beta-m}} = c_1\langle \alpha + m \rangle_m + c_2 \langle \beta \rangle_m\,.$$

Looking similarly at $D_{q,m}(h)_{|x^{\alpha-m}y^{\beta}}$ and $D_{q,2\,m}(h)_{|x^{\alpha-m}y^{\beta-m}}$, shows that the coefficients c_1, c_2, and c_3 must satisfy the following system of equations :

$$\begin{array}{lllllll}
\langle \alpha + m \rangle_m\ c_1 & + & \langle \beta \rangle_m\, c_2 & & & = & 0 \\
 & & \langle \alpha \rangle_m\, c_2 & + & \langle \beta + m \rangle_m\ c_3 & = & 0 \\
\langle \alpha + m \rangle_{2m}\, c_1 & & & + & \langle \beta + m \rangle_{2m}\, c_3 & = & 0
\end{array}$$

whose determinant is :

$$\frac{(\alpha+m)!(\beta+m)!}{(\alpha-m)!(\beta-m)!}(1+q\,(\alpha-m))(1+q\,(\beta-m))(2+q\,(\alpha+\beta))\,.$$

Therefore $c_1 = c_2 = c_3 = 0$ whenever q is not one of the announced singular values. □

Corollaire 6.5.3. *For $W = G(m,p,2)$ and $q \neq -a/b, 1 \leqslant a \leqslant 2 \leqslant b$, the space $\mathcal{H}W_q$ is isomorphic as a graded W-module to $\mathcal{H}W$. Its basis is obtained by considering the layers $L_0(q), \dots, L_{m/p-1}(q)$ of the q-harmonic polynomials for $G(m,2)$.*

Démonstration. Select out of Equation (6.5.3) the elements which satisfy the extra equation $\varepsilon^{m/p}(f) = 0$. See also, in Figure 6.5, the vertical arrows which depict the action of ε. □

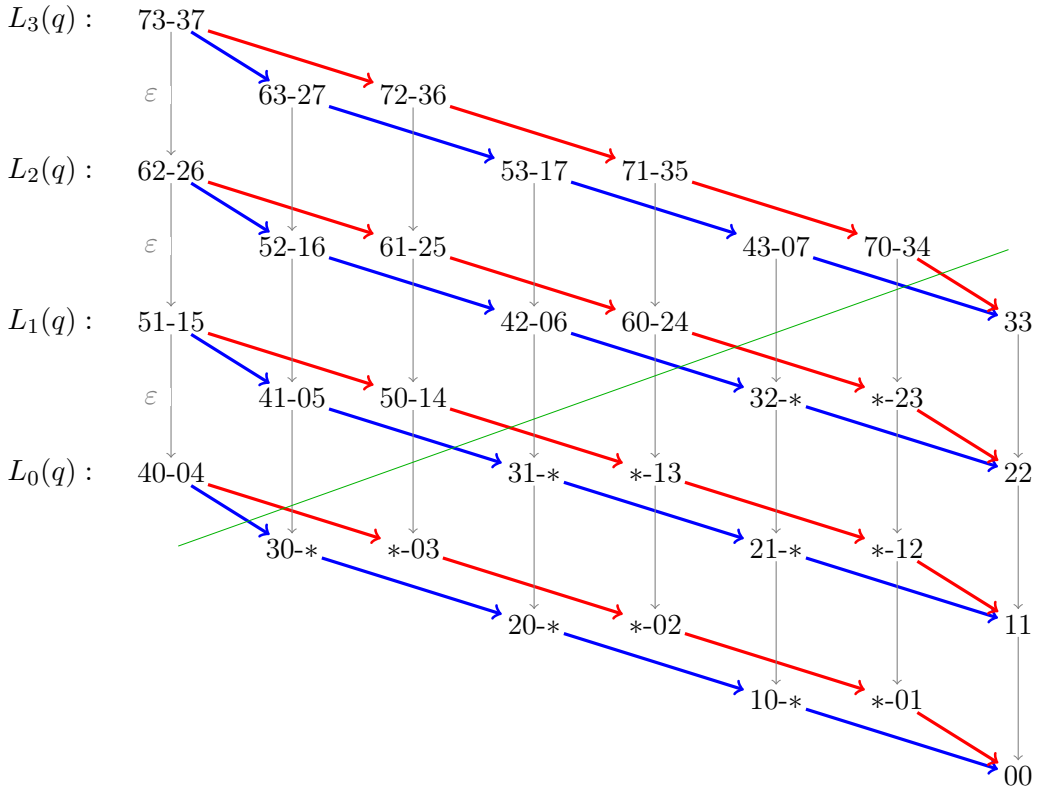


FIGURE 6.1 –
Structure of q-Harmonic polynomials of $G(4,2)$. For short, the q-harmonic binomial $\langle d\rangle_4 x^\alpha y^\beta - \langle a\rangle_4 x^{\alpha'} y^{\beta'}$ is denoted "$\alpha\beta$-$\alpha'\beta'$". Similarly, the q-harmonic monomial $x^\alpha y^\beta$ is denoted "$\alpha\beta$", "$\alpha\beta$-$*$", or "$*$-$\alpha\beta$". The blue (resp. red) arrows denote the action of the would be q-analogues of the operators ∂_x and ∂_y within each layer L_i. The gray arrows denote the action of the operator $\varepsilon = e_2(\partial_\checkmark) = \partial_x\partial_y$ (recall that it changes the value of q). The green line separates the q-harmonic monomials and binomials.

Pour fixer les idées et satisfaire la curiosité du lecteur voici, pour le groupe $G(4,2)$, une base

des polynômes Harmonique déformés dans l'anneau $\mathbb{Q}[q][x_1, x_2]$.

$$
\begin{aligned}
&1\\
&x_1\\
&x_2\\
&x_1^2\\
&x_1x_2\\
&x_2^2\\
&x_1^3\\
&x_1^2x_2\\
&x_1x_2^2\\
&x_2^3\\
&x_1^4 - x_2^4\\
&x_1^3x_2\\
&x_1^2x_2^2\\
&x_1x_2^3\\
&x_1^5 + (-5q-5)x_1x_2^4\\
&(-5q-5)x_1^4x_2 + x_2^5\\
&x_1^3x_2^2\\
&x_1^2x_2^3\\
&x_1^6 + (-30q-15)x_1^2x_2^4\\
&x_1^5x_2 - x_1x_2^5\\
&(-30q-15)x_1^4x_2^2 + x_2^6\\
&x_1^3x_2^3\\
&x_1^7 + (-105q-35)x_1^3x_2^4\\
&(-120q-120)x_1^6x_2 + (720q+360)x_1^2x_2^5\\
&(-720q-360)x_1^5x_2^2 + (120q+120)x_1x_2^6\\
&(-105q-35)x_1^4x_2^3 + x_2^7\\
&(-120q-120)x_1^7x_2 + (2520q+840)x_1^3x_2^5\\
&x_1^6x_2^2 - x_1^2x_2^6\\
&(-2520q-840)x_1^5x_2^3 + (120q+120)x_1x_2^7\\
&(-720q-360)x_1^7x_2^2 + (2520q+840)x_1^3x_2^6\\
&(-2520q-840)x_1^6x_2^3 + (720q+360)x_1^2x_2^7\\
&x_1^7x_2^3 - x_1^3x_2^7
\end{aligned}
$$

Cette base contient bien $4^2 \cdot 2! = 32$ polynômes.

Chapitre 7

Un module sage Pour l'énumération modulo l'action d'un groupe de permutation

Tout le travail d'implantation informatique connexe au travail de thèse a été entrepris dans une optique de développement sur le long terme. Un bon code se veut réutilisable par des tiers, et dont la maintenance est facilitée et non exclusivement assurée par son auteur. De manière pratique, il est aussi intéressant de l'intégrer dans un système plus complet, cela favorise les branchements vers d'autres théories, d'autres problèmes. Calculer les invariants n'étant pas forcément une fin en soi, on peut espérer sur le moyen ou long terme pouvoir faire de la théorie de Galois effective ou encore résoudre des systèmes d'équations avec symétries.

Sur un autre registre plus technique, il convient aussi de développer du code factorisé. Factorisé dans le sens que les fonctions de bases n'ont pas à être programmées plusieurs fois. Elles doivent être optimisées en vitesse, en adaptation et réutilisées le plus souvent possible. Chaque projet informatique doit être sagement divisé en problèmes atomiques et ces derniers doivent être développés non pas pour traiter notre problème recherché mais la plus grande classe de problème connexe. Ainsi, pour énumérer les vecteurs d'entiers modulo l'action d'un groupe de permutations, un des problèmes atomiques apparaissant était l'énumération en largeur et profondeur des structures arborescentes. Les structures arborescentes apparaissant dans un grand nombre de problèmes de combinatoire (en particulier énumérative), avec l'aide des conseils de Florent Hivert et Nicolas M. Thiéry, nous avons enrichi l'outil `SearchForest` du système `Sage`.

Définition 7.0.4 (itérable)**.** *Un* itérateur sur un ensemble *(à la mode* `Python`*) est un objet informatique qui se consume lors de son utilisation progressive. On l'utilise avec la méthode* `.next()`*, à chaque appel de cette méthode, un élément de l'ensemble est retourné. Chaque élément de l'ensemble considéré doit être retourné une et une fois lors du dépliage de l'itérateur.*

Voici un exemple zéro d'itérateur informatique en `Sage`.

```
sage: S = Primes(); S
Set of all prime numbers: 2, 3, 5, 7, ...
sage: it = iter(S); it
<generator object _iterator_from_next at 0xc04ce64>
```

```
sage: for i in range(10): it.next()
2
3
5
7
11
13
17
19
23
29
sage: it.next()
31
sage: it.next()
37
```

L'utilité de cet objet est cruciale lorsque l'on veut énumérer des ensembles infinis dénombrables de manière feignante et jusqu'à un certain ordre que le programmeur ne connaît pas *a priori*.

7.1 `SearchForest`, un outil pour le parcours paresseux d'arbres de recherche

Ce module permet d'avoir accès à différentes fonctionnalités d'énumération sur des structures arborescentes.

L'utilisateur doit fournir deux arguments primordiaux : un itérateur sur les différents noeuds-racines de la structure et une fonction qui associe à chaque noeud un itérateur sur ces enfants.

Voici un premier exemple d'utilisation de `SearchForest`, il correspond à construire $\mathbb{N}$ dans l'esprit des axiomes de Peano. On donne un ensemble formé d'une seule racine 0 et on programme une fonction qui, à chaque entier naturel n, retourne son successeur $n+1$. Une fois construit, `SearchForest` fournira un itérateur.

```
sage: S = SearchForest([0], lambda n: [n+1])
sage: it = iter(S)
sage: it.next()
0
sage: it.next()
1
sage: it.next()
2
sage: it.next()
3
sage: it.next()
4
sage: it.next()
5
```

Dans le cadre du problème d'énumération modulo l'action d'un groupe de permutations, nous sommes tombés devant un problème lié à la manière de parcourir la structure arborescente.

En effet, pour un ensemble infini dénombrable, suivant la structure de l'arbre, on n'est pas sur de pouvoir énumérer tous les objets présents dans notre structure. L'exemple suivant construit de toutes les paires (i, j) dans $\mathbb{N}^2$.

```
sage: I = SearchForest([(0,0)],
...                    lambda l: [(l[0]+1, l[1]), (l[0], 1)]
...                              if l[1] == 0 else [(l[0], l[1]+1)])
```

Avec un parcours de recherche en profondeur, toutes les paires ne seront pas atteintes. Seules les paires du type $(i, 0)$ seront énumérées par l'itérateur.

```
sage: depth_search = I.depth_first_search_iterator()
sage: [depth_search.next() for i in range(7)]
[(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0)]
```

En utilisant un parcours en largeur, on obtient l'itérateur anti-diagonal usuel.

```
sage: breadth_search = I.breadth_first_search_iterator()
sage: [breadth_search.next() for i in range(15)]
[(0, 0),
(1, 0), (0, 1),
(2, 0), (1, 1), (0, 2),
(3, 0), (2, 1), (1, 2), (0, 3),
(4, 0), (3, 1), (2, 2), (1, 3), (0, 4)]
```

L'amélioration de `SearchForest` pour supporter le parcours en largeur fut une des tâches préliminaires pour l'énumération des vecteurs d'entiers. Durant le travail d'implantation relatif à cette thèse, nous avons proposé un patch qui fut arbitré puis intégré au logiciel `Sage`.

Revenons aux vecteurs d'entiers. Pour tout groupe de permutations G, sous groupe de $\mathfrak{S}_n$, on peut énumérer les vecteurs d'entiers canoniques sous l'action de G en considérant la racine $(0, \ldots, 0)$ et la fonction `canonical_children`.

```
sage: from sage.combinat.enumeration_mod_permgroup import canonical_children
sage: G = PermutationGroup([[(1,2,3,4)]]); G
Permutation Group with generators [(1,2,3,4)]
sage: sgs = G.strong_generating_system()
sage: canonical_children(sgs, [0,0,0,0])
[[1, 0, 0, 0]]
sage: canonical_children(sgs, [1,0,0,0])
[[2, 0, 0, 0], [1, 1, 0, 0], [1, 0, 1, 0]]
sage: canonical_children(sgs, [1,1,0,0])
[[1, 1, 1, 0]]
```

```
sage: canonical_children(sgs, [1,1,1,0])
[[1, 1, 1, 1]]
sage: canonical_children(sgs, [2,0,0,0])
[[3, 0, 0, 0], [2, 1, 0, 0], [2, 0, 1, 0], [2, 0, 0, 1]]
```

Ce sont les deux pièces nécessaires à l'utilisation de `SearchForest`.

```
sage: S = SearchForest([[0,0,0,0]],
                   lambda x: canonical_children(sgs, x),
                   algorithm='breadth')
sage: it = iter(S)
sage: it.next()
[0, 0, 0, 0]
sage: it.next()
[1, 0, 0, 0]
sage: it.next()
[2, 0, 0, 0]
sage: it.next()
[1, 1, 0, 0]
sage: it.next()
[1, 0, 1, 0]
sage: it.next()
[3, 0, 0, 0]
sage: it.next()
[2, 1, 0, 0]
sage: it.next()
[2, 0, 1, 0]
sage: it.next()
[2, 0, 0, 1]
sage: it.next()
[1, 1, 1, 0]
sage: it.next()
[4, 0, 0, 0]
```

Ceci constitue la structure primitive en interne que nous allons ensuite habiller pour que l'utilisation soit la plus aisée possible pour l'utilisateur. Notons aussi que `SearchForest` permet de récupérer des itérateurs sur les éléments d'une profondeur donnée. Ici, cela revient à récupérer les vecteurs canoniques d'une somme donnée.

```
sage: S5 = S.elements_of_depth_iterator(5)
sage: [v for v in S5]
[[5, 0, 0, 0], [4, 1, 0, 0], [4, 0, 1, 0], [4, 0, 0, 1], [3, 2, 0, 0],
 [3, 1, 1, 0], [3, 1, 0, 1], [3, 0, 2, 0], [3, 0, 1, 1], [3, 0, 0, 2],
 [2, 2, 1, 0], [2, 2, 0, 1], [2, 1, 2, 0], [2, 1, 1, 1]]
```

Nous tenions à expliquer ce module pour ne pas occulter ce qui ce passe en interne lors de l'énumération des vecteurs d'entiers. Nous remercions particulièrement Florent Hivert et

Nicolas M. Thiéry pour toutes les suggestions de design qui font l'adaptabilité de ce module `SearchForest`.

7.2 Construction des canoniques sous l'action d'un groupe de permutations

Il s'agit maintenant de construire informatiquement l'ensemble infini mais énumérable de tous les vecteurs d'entiers représentants d'orbite sous l'action d'un groupe de permutation. En théorie de la calculabilité, on dit qu'un tel ensemble est récursivement énumérable. Comme exposé théoriquement dans la première partie de cette thèse, la structure d'arbre sous-jacente motive l'utilisation du module `SearchForest` (bien que, d'un point de vue pratique, c'est ce cahier des charges qui a motivé l'enrichissement de `SearchForest`).

Pour illustrer l'objet `IntegerVectorsModPermutationGroup`, nous prendrons, pour toute cette section, l'exemple du groupe G engendré par les permutations $(1, 2, 3, 4, 5)$ et $(1, 2, 4, 3)$.

```
sage: G = PermutationGroup([[(1,2,3,4,5)],[(1,2,4,3)]]); G
Permutation Group with generators [(1,2,3,4,5), (1,2,4,3)]
sage: G.cardinality()
20
```

Pour accéder à l'ensemble I de tous les vecteurs canoniques pour le groupe G, on appelle le constructeur `IntegerVectorsModPermutationGroup`.

```
sage: I = IntegerVectorsModPermutationGroup(G)
sage: I.category()
Category of infinite enumerated sets
sage: I.cardinality()
+Infinity
```

Comme annoncé, il est possible d'énumérer cet ensemble de manière feignante à tous les crans.

```
sage: it = iter(I)
sage: it.next()
[0, 0, 0, 0, 0]
sage: it.next()
[1, 0, 0, 0, 0]
sage: it.next()
[2, 0, 0, 0, 0]
sage: it.next()
[1, 1, 0, 0, 0]
sage: it.next()
[3, 0, 0, 0, 0]
sage: it.next()
```

```
[2, 1, 0, 0, 0]
sage: for i in range(1000): v = it.next()
....:
sage: it.next()
[7, 5, 1, 2, 1]
sage: it.next()
[7, 5, 1, 1, 2]
sage: it.next()
[7, 5, 1, 0, 3]
```

On peut récupérer facilement le sous-ensemble fini des coniques de somme donnée.

```
sage: I4 = I.subset(sum=4); I4
Integer vectors of length 5 and of sum 4 enumerated up to the action of
Permutation Group with generators [(1,2,3,4,5), (1,2,4,3)]
sage: I4.category()
Category of finite enumerated sets
sage: I4.cardinality()
6
sage: I4.list()
[[4, 0, 0, 0, 0], [3, 1, 0, 0, 0], [2, 2, 0, 0, 0], [2, 1, 1, 0, 0],
 [2, 1, 0, 0, 1], [1, 1, 1, 1, 0]]
```

Ces deux ensembles `Sage`, présentent de nombreuses fonctionnalités comme un test d'appartenance :

```
sage: [1,1,1,1,1] in I
True
sage: [1,0,1,0,1] in I
False
sage: [3,2,0,1,0] in I
True
sage: [3,2,0,2,0] in I
False
```

La méthode `orbit` permet de déployer l'orbite d'un vecteur d'entiers sous l'action du groupe de permutations.

```
sage: I.orbit([1,1,1,1,1])
[[1, 1, 1, 1, 1]]
sage: I.orbit([1,0,1,0,1])
[[1, 0, 1, 0, 1], [1, 1, 1, 0, 0], [1, 1, 0, 1, 0], [1, 0, 0, 1, 1],
 [0, 1, 0, 1, 1], [0, 0, 1, 1, 1], [0, 1, 1, 0, 1], [0, 1, 1, 1, 0],
 [1, 0, 1, 1, 0], [1, 1, 0, 0, 1]]
sage: I.orbit([3,2,0,1,0])
[[3, 2, 0, 1, 0], [3, 0, 0, 2, 1], [3, 0, 1, 0, 2], [3, 1, 2, 0, 0],
```

```
 [2, 0, 1, 0, 3], [2, 1, 3, 0, 0], [2, 3, 0, 1, 0], [2, 0, 0, 3, 1],
 [0, 1, 0, 3, 2], [0, 0, 2, 1, 3], [0, 2, 3, 0, 1], [0, 3, 1, 2, 0],
 [1, 0, 3, 2, 0], [1, 3, 0, 0, 2], [1, 0, 2, 3, 0], [1, 2, 0, 0, 3],
 [0, 3, 2, 0, 1], [0, 2, 1, 3, 0], [0, 1, 0, 2, 3], [0, 0, 3, 1, 2]]
sage: I.orbit([3,2,0,2,0])
[[3, 2, 0, 2, 0], [3, 0, 0, 2, 2], [3, 0, 2, 0, 2], [3, 2, 2, 0, 0],
 [2, 0, 2, 0, 3], [2, 2, 3, 0, 0], [2, 3, 0, 2, 0], [2, 0, 0, 3, 2],
 [0, 2, 0, 3, 2], [0, 0, 2, 2, 3], [0, 2, 3, 0, 2], [0, 3, 2, 2, 0],
 [2, 0, 3, 2, 0], [2, 3, 0, 0, 2], [2, 0, 2, 3, 0], [2, 2, 0, 0, 3],
 [0, 3, 2, 0, 2], [0, 2, 2, 3, 0], [0, 2, 0, 2, 3], [0, 0, 3, 2, 2]]
```

Ce module est en cours de finalisation. Il est possible de l'utiliser via l'installation de `Sage-Combinat`. Une fois corrigées et arbitrées, ces fonctionnalités intégreront le logiciel `Sage` et ses futures versions.

Chapitre 8

L'anneau des invariants d'un groupe de permutation dans Sage

Ce dernier chapitre montre la démarche suivie pour produire un code effectif pour calculer les invariants d'un groupe de permutations. Le module présenté ici est toujours en phase de développement, correction et arbitrage. La procédure d'intégration de nouveaux codes dans Sage est lourde mais le code qui en ressort est ainsi de meilleure qualité.

8.1 Esprit de l'implantation, «building the car»

Le schéma suivant montre l'état des lieux du logiciel Sage lorsque nous avons entrepris le développement d'un module pour le calcul des invariants d'un groupe de permutations. Les modules non soulignés correspondent aux fonctionnalités déjà présentes dans Sage et les soulignés ceux que nous devons développer.

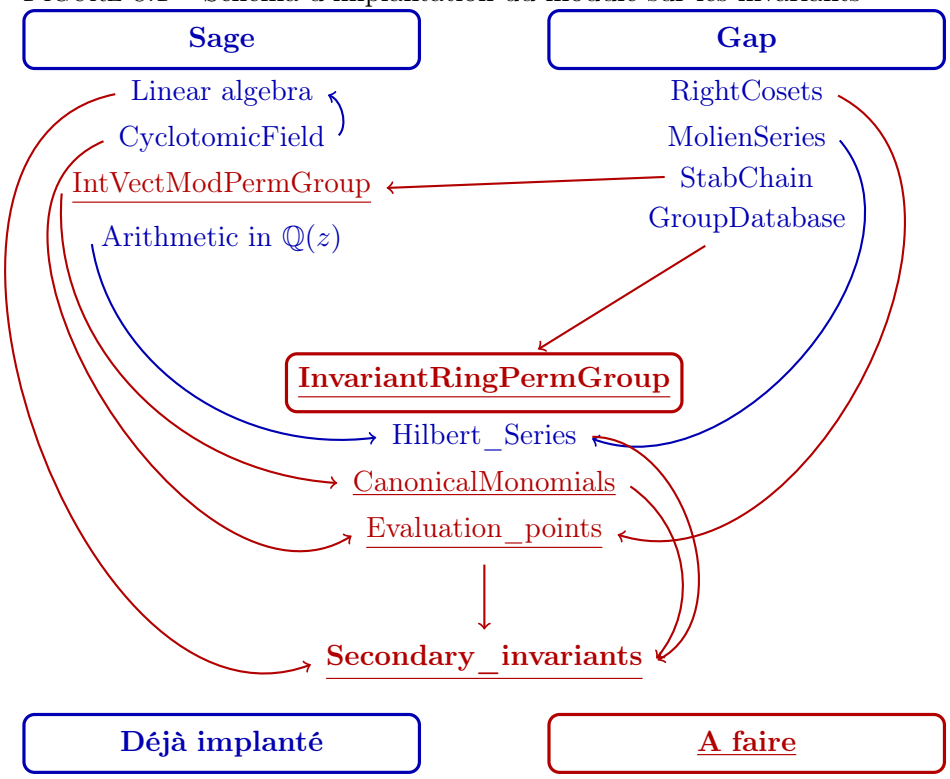
Le code que nous avons établi et testé est toujours en phase de développement. Présent sur le serveur Sage-Combinat, il est possible de l'installer et de l'utiliser par quiconque possédant une connexion internet (et une configuration matérielle supportant l'installation de Sage).

Le choix du logiciel pour illustrer effectivement cette thèse fut largement motivé par cet état des lieux prometteur que présentait Sage.

Pour donner une idée de la tâche, environ 2000 lignes de codes permettent d'énumérer les vecteurs d'entiers modulo l'action d'un groupe de permutation. Construire les points d'évaluation, le morphisme d'évaluation et la représentation des invariants comme combinaison de somme sur orbite ont nécessité aussi 2000 lignes de codes. L'algorithme de calcul des secondaires avec toutes ses optimisations prend environ 1000 lignes de code Sage.

Tous les objets ont été implantés en utilisant largement le nouveau mécanisme Sage des catégories (catégories informatiques fondées sur les catégories et classes apparaissant dans les mathématiques). En effet, c'est un cahier des charges supplémentaires à vérifier pour l'implantation mais il permet ensuite de bénéficier de fonctionnalités comme des constructions fonctorielles.

FIGURE 8.1 – Schéma d'implantation du module sur les invariants



8.2 L'anneau abstrait des invariants et ses différentes représentations

Le point d'entrée pour l'utilisateur est le constructeur `InvariantRingPermutationGroup`. Ce dernier demande deux arguments ; un groupe de permutations et un corps. Pour le moment, le corps attendu est le corps des rationnels. À terme, nous souhaiterons améliorer ce module pour qu'il puisse aussi traiter les cas modulaires. Ainsi, construire l'anneau abstrait des invariants se fait comme il suit :

```
sage: G = TransitiveGroup(6,8); G
Transitive group number 8 of degree 6
sage: G.gens()
[(1,3,5)(2,4,6), (1,4)(2,5), (1,5)(2,4)(3,6)]
sage: I = InvariantRingPermutationGroup(G, QQ); I
Abstract algebra of invariant polynomial under the action of Transitive
group number 8 of degree 6 over Rational Field
sage: I.category()
Category of graded algebras over Rational Field
```

Cette algèbre est abstraite (ceci est plus ou moins une convention informatique fondée sur des arguments mathématiques) dans `Sage` dans le sens que de nombreuses fonctionnalités sont accessibles mais, par exemple, il n'y a pas de base connue pour cette algèbre. Sans base, nous

n'avons ni combinaison linéaire, ni structure de données pour les éléments. C'est dans différentes réalisations ou représentations de cette algèbre que nous allons pouvoir faire des calculs explicites.

```
sage: A = I.representation_as_orbit_sum(); A
Algebra of invariant polynomials under the action of Transitive group
number 8 of degree 6 over Rational Field view as combination of orbit
sum of monomials
sage: A.category()
Category of graded algebras with basis over Rational Field
sage: A.zero()
0
sage: A.one()
o(0, 0, 0, 0, 0, 0)
sage: P = A.an_element(); P
3*o(1, 1, 0, 0, 0, 0) + 2*o(2, 0, 0, 0, 0, 0)
sage: Q = 3*P - A.basis()[(5,4,3,2,1,0)]; Q
9*o(1, 1, 0, 0, 0, 0) + 6*o(2, 0, 0, 0, 0, 0) - o(5, 4, 3, 2, 1, 0)
sage: Q*Q
Traceback (most recent call last):
...
NotImplementedError:
```

La représentation de l'algèbre des invariants comme combinaison linéaire de sommes sur orbite possède une base explicite (indexée par les vecteurs d'entiers canoniques sous l'action du groupe) et une structure de données pour ces éléments. Cette algèbre graduée possède un neutre pour l'addition et la multiplication. Il est possible de faire des combinaisons linéaires mais pas de produit de manière générale. Même si nous comptons implanter ce produit à terme, l'approche par évaluation veut justement éviter de tels calculs dans cette représentation qui sont forts coûteux dû à la dimension très importante des composantes homogènes. C'est dans l'espace des évaluations que nous ferons les produits.

```
sage: E = I.representation_as_evaluation(); E
Algebra of invariant polynomials under Transitive group number 8 of degree 6
as vector of evaluations
sage: E.category()
Category of vector spaces over Cyclotomic Field of order 6 and degree 2
sage: E.dimension()
30
sage: E.zero()
(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)
sage: E.one()
(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1)
sage: P = E.an_element(); P
(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)
sage: Q = 3*P + E.one(); Q
```

```
(4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1)
sage: E.product(Q,Q)
(16, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1)
```

Voici la représentation du quotient de l'algèbre des invariants par l'idéal $\langle e_1, \ldots, e_{n-1}, e_n - \epsilon \rangle$. Comme attendu, c'est un espace vectoriel sur un corps cyclotomique de dimension $\frac{6!}{|G|} = 30$. On vérifie visuellement la consistance des deux neutres vis à vis du produit point par point.

Entre les différentes réalisations de l'algèbre des invariants abstraite, il existe des passerelles, notamment le morphisme d'évaluation Φ, un des objets centraux de cette thèse. Évaluons donc les sommes de puissance aux racines de l'unité par Φ.

```
sage: A = I.representation_as_orbit_sum(); A
Algebra of invariant polynomials under the action of Transitive group
number 8 of degree 6 over Rational Field view as combination of orbit
sum of monomials
sage: A._element_basis_to_evaluation((0,0,0,0,0,0))
(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1)
sage: A._element_basis_to_evaluation((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)
sage: A._element_basis_to_evaluation((2,0,0,0,0,0))
(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)
sage: A._element_basis_to_evaluation((3,0,0,0,0,0))
(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)
sage: A._element_basis_to_evaluation((4,0,0,0,0,0))
(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)
sage: A._element_basis_to_evaluation((5,0,0,0,0,0))
(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)
sage: A._element_basis_to_evaluation((6,0,0,0,0,0))
(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
 6, 6, 6, 6, 6)
```

Ces résultats étaient attendus, évaluons maintenant des bons candidats pour être invariants secondaires.

```
sage: A._element_basis_to_evaluation((1,1,0,0,0,0))
(0, -1, zeta6, 2*zeta6 - 2, 2*zeta6 - 2, 3*zeta6 - 3, -zeta6 + 1, -2*zeta6,
 2, 2*zeta6 - 2, 0, 3*zeta6 - 3, 2, -2*zeta6, 3, -1, 0, 2*zeta6 - 2, 2,
 -zeta6 + 1, 3, 0, 2, zeta6, -2*zeta6, -3*zeta6, 0, -3*zeta6, 0, -2*zeta6)
sage: A._element_basis_to_evaluation((2,1,0,0,0,0))
```

```
(0, 0, 0, -2*zeta6 + 1, 2*zeta6 - 1, 0, 0, 2*zeta6 - 1, 2*zeta6 - 1,
 -2*zeta6 + 1, 6*zeta6 - 3, 0, -2*zeta6 + 1, 2*zeta6 - 1, 0, 0, 6*zeta6 - 3,
 2*zeta6 - 1, -2*zeta6 + 1, 0, 0, 0, 2*zeta6 - 1, 0, -2*zeta6 + 1, 0,
 -6*zeta6 + 3, 0, -6*zeta6 + 3, -2*zeta6 + 1)
sage: A._element_basis_to_evaluation((3,2,1,0,0,0))
(0, -2, 10, 7, 1, 12*zeta6 - 6, -2, 7, 1, 1, -9, -12*zeta6 + 6, 7, 1,
 -12*zeta6 + 6, 10, 9, 7, 1, 10, 12*zeta6 - 6, 0, 7, -2, 7, -12*zeta6 + 6, 9,
 12*zeta6 - 6, -9, 1)
```

Les points d'évaluations, peuvent être récupérés comme il suit :

```
sage: from sage.combinat.invariant_ring_perm_gps.evaluation import
Evaluation_points
sage: G = TransitiveGroup(6,8); G
Transitive group number 8 of degree 6
sage: Points = Evaluation_points(G); Points
Evaluation points for the invariant polynomials under the action of
Transitive group number 8 of degree 6
sage: Points.category()
Category of finite enumerated sets
sage: Points.cardinality()
30
sage: Points.list()
[(1, zeta6, zeta6 - 1, -1, -zeta6, -zeta6 + 1),
 (1, zeta6, zeta6 - 1, -1, -zeta6 + 1, -zeta6),
 (1, zeta6, zeta6 - 1, -zeta6, -1, -zeta6 + 1),
 ...
 ... (Il y a bien 30 points en tout)
 ...
 (1, zeta6 - 1, -zeta6, zeta6, -zeta6 + 1, -1),
 (1, zeta6 - 1, -zeta6 + 1, zeta6, -1, -zeta6),
 (1, zeta6 - 1, -zeta6 + 1, zeta6, -zeta6, -1)]
```

Revenons à l'algèbre abstraite, c'est dans cette dernière que les algorithmes évolués sont implantés. On peut ainsi récupérer le système des invariants primaires et aussi le polynôme énumérateur des degrés des invariants secondaires.

```
sage: I = InvariantRingPermutationGroup(G, QQ); I
Abstract algebra of invariant polynomial under the action of Transitive
group number 8 of degree 6 over Rational Field
sage: I.primary_invariants()
{1: [o(1, 0, 0, 0, 0, 0)],
 2: [o(2, 0, 0, 0, 0, 0)],
 3: [o(3, 0, 0, 0, 0, 0)],
 4: [o(4, 0, 0, 0, 0, 0)],
 5: [o(5, 0, 0, 0, 0, 0)],
 6: [o(6, 0, 0, 0, 0, 0)]}
```

```
sage: I.secondary_invariants_series()
3*z^12 + z^11 + 4*z^10 + 4*z^9 + 5*z^8 + 2*z^7 + 5*z^6 + z^5 + 2*z^4
+ z^3 + z^2 + 1
```

Ceci décrit à peu près toute la technologie préliminaire avant de pouvoir construire un système d'invariants secondaires associé aux polynômes symétriques.

8.3 Calcul des invariants secondaires

Pour obtenir les invariants secondaires associés au système de primaires formé par les polynômes symétriques, on appelle la méthode `Secondary_invariants`.

```
sage: I
Abstract algebra of invariant polynomial under the action of Transitive
group number 8 of degree 6 over Rational Field
sage: I.secondary_invariants()
{0: [[(0, 0, 0, 0, 0, 0)]],
 1: [],
 2: [[(1, 1, 0, 0, 0, 0)]],
 3: [[(2, 1, 0, 0, 0, 0)]],
 4: [[(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0)], [(3, 1, 0, 0, 0, 0)]],
 5: [[(1, 1, 0, 0, 0, 0), (2, 1, 0, 0, 0, 0)]],
 6: [[(1, 1, 0, 0, 0, 0), (3, 1, 0, 0, 0, 0)],
    [(2, 1, 0, 0, 0, 0), (2, 1, 0, 0, 0, 0)],
    [(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0)],
    [(3, 2, 1, 0, 0, 0)], [(2, 2, 1, 1, 0, 0)]],
 7: [[(2, 1, 0, 0, 0, 0), (3, 1, 0, 0, 0, 0)], [(4, 2, 1, 0, 0, 0)]],
 8: [[(1, 1, 0, 0, 0, 0), (3, 2, 1, 0, 0, 0)],
    [(1, 1, 0, 0, 0, 0), (2, 2, 1, 1, 0, 0)],
    [(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0), (3, 1, 0, 0, 0, 0)],
    [(3, 1, 0, 0, 0, 0), (3, 1, 0, 0, 0, 0)], [(5, 2, 1, 0, 0, 0)]],
 9: [[(1, 1, 0, 0, 0, 0), (4, 2, 1, 0, 0, 0)],
    [(2, 1, 0, 0, 0, 0), (3, 2, 1, 0, 0, 0)],
    [(2, 1, 0, 0, 0, 0), (2, 2, 1, 1, 0, 0)], [(5, 3, 1, 0, 0, 0)]],
 10: [[(1, 1, 0, 0, 0, 0), (5, 2, 1, 0, 0, 0)],
     [(2, 1, 0, 0, 0, 0), (4, 2, 1, 0, 0, 0)],
     [(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0), (3, 2, 1, 0, 0, 0)],
     [(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0), (2, 2, 1, 1, 0, 0)]],
 11: [[(1, 1, 0, 0, 0, 0), (5, 3, 1, 0, 0, 0)]],
 12: [[(2, 1, 0, 0, 0, 0), (5, 3, 1, 0, 0, 0)],
     [(1, 1, 0, 0, 0, 0), (1, 1, 0, 0, 0, 0), (5, 2, 1, 0, 0, 0)],
     [(1, 1, 0, 0, 0, 0), (2, 1, 0, 0, 0, 0), (4, 2, 1, 0, 0, 0)]]}
```

La méthode retourne ici un dictionnaire `Python` dans lequel les clés sont des degrés (des entiers) et les entrées sont des listes de listes de vecteurs d'entiers canoniques. Ainsi en degré 0, la première liste a un seul élément donc il y aura qu'un seul secondaire. L'élément de cette liste est

une liste contenant uniquement le vecteur canonique $(0, 0, 0, 0, 0, 0)$, cela signifie que ce secondaire est juste la somme sur orbite de la constante 1. Pour le degré 4, la liste contient deux éléments, le premier est une liste composée de deux fois le même vecteur d'entiers canonique $(1, 1, 0, 0, 0)$. Cela signifie que le carré de la somme sur orbite du monôme x_1x_2 est un invariant secondaire (nécessairement non irréductible). L'autre liste comporte un seul vecteur : $(3, 1, 0, 0, 0, 0)$, ainsi la somme sur orbite du monôme dont les exposants sont donnés par les coordonnées de ce vecteur constitue le second invariant secondaire pour ce degré 4.

On peut rendre la fonction verbale en définissant l'argument optionnel `verbose` comme étant vrai `True`. La fonction affiche alors au fur et à mesure les calculs entrepris par l'algorithme. Les détails techniques de l'algorithme expliqués dans la partie théorique de cette thèse peuvent alors être observée en action.

```
sage: I.secondary_invariants(verbose=True)
Initialiation of secondary of degree 0
  ------
Secondary invariants of degree 1 :
  We must search 0 secondary invariants
  ------
Secondary invariants of degree 2 :
  We must search 1 secondary invariants
      Research of products of secondaries of degree smaller
      Research now to complete with new irreducible secondaries
        New irreducible [2]
  ------
Secondary invariants of degree 3 :
  We must search 1 secondary invariants
      Research of products of secondaries of degree smaller
      Research now to complete with new irreducible secondaries
        New irreducible [3]
  ------
Secondary invariants of degree 4 :
  We must search 2 secondary invariants
      Research of products of secondaries of degree smaller
        Add product [2, 2]
      Research now to complete with new irreducible secondaries
        New irreducible [4]
  ------
Secondary invariants of degree 5 :
  We must search 1 secondary invariants
      Research of products of secondaries of degree smaller
        Add product [2, 3]
  ------
Secondary invariants of degree 6 :
  We must search 5 secondary invariants
    Correction by adding the space spanned by secondaries of degree 0
      Research of products of secondaries of degree smaller
        Add product [2, 4]
        Add product [3, 3]
```

```
      Add product [2, 2, 2]
    Research now to complete with new irreducible secondaries
      (5, 1, 0, 0, 0, 0) is not a good secondary invariant
      (5, 0, 0, 1, 0, 0) is not a good secondary invariant
      (4, 2, 0, 0, 0, 0) is not a good secondary invariant
      (4, 1, 1, 0, 0, 0) is not a good secondary invariant
      (4, 1, 0, 1, 0, 0) is not a good secondary invariant
      (4, 1, 0, 0, 1, 0) is not a good secondary invariant
      (4, 0, 0, 2, 0, 0) is not a good secondary invariant
      (3, 3, 0, 0, 0, 0) is not a good secondary invariant
      New irreducible [5]
      (3, 2, 0, 1, 0, 0) is not a good secondary invariant
      (3, 2, 0, 0, 1, 0) is not a good secondary invariant
      (3, 1, 1, 1, 0, 0) is not a good secondary invariant
      (3, 1, 1, 0, 1, 0) is not a good secondary invariant
      (3, 1, 0, 2, 0, 0) is not a good secondary invariant
      (3, 1, 0, 1, 1, 0) is not a good secondary invariant
      (2, 2, 2, 0, 0, 0) is not a good secondary invariant
      New irreducible [6]
  ------
Secondary invariants of degree 7 :
  We must search 2 secondary invariants
    Correction by adding the space spanned by secondaries of degree 1
      Research of products of secondaries of degree smaller
        Add product [3, 4]
        Register new relation : [2, 2, 3]
      Research now to complete with new irreducible secondaries
        (5, 2, 0, 0, 0, 0) is not a good secondary invariant
        (5, 1, 1, 0, 0, 0) is not a good secondary invariant
        (5, 1, 0, 1, 0, 0) is not a good secondary invariant
        (5, 1, 0, 0, 1, 0) is not a good secondary invariant
        (5, 0, 0, 2, 0, 0) is not a good secondary invariant
        (4, 3, 0, 0, 0, 0) is not a good secondary invariant
        New irreducible [7]
  ------
Secondary invariants of degree 8 :
  We must search 5 secondary invariants
    Correction by adding the space spanned by secondaries of degree 2
      Research of products of secondaries of degree smaller
        Add product [2, 5]
        Add product [2, 6]
        Add product [2, 2, 4]
        Add product [4, 4]
        Register new relation : [2, 3, 3]
        Register new relation : [2, 2, 2, 2]
      Research now to complete with new irreducible secondaries
        (5, 3, 0, 0, 0, 0) is not a good secondary invariant
        New irreducible [8]
  ------
```

```
Secondary invariants of degree 9 :
  We must search 4 secondary invariants
    Correction by adding the space spanned by secondaries of degree 3
      Research of products of secondaries of degree smaller
        Add product [2, 7]
        Add product [3, 5]
        Add product [3, 6]
        Register new relation : [2, 3, 4]
        Register new relation : [3, 3, 3]
        Use of the relation [2, 2, 2, 3]
      Research now to complete with new irreducible secondaries
        (5, 4, 0, 0, 0, 0) is not a good secondary invariant
        New irreducible [9]
  ------
Secondary invariants of degree 10 :
  We must search 4 secondary invariants
    Correction by adding the space spanned by secondaries of degree 4
      Research of products of secondaries of degree smaller
        Add product [2, 8]
        Add product [3, 7]
        Add product [2, 2, 5]
        Add product [2, 2, 6]
  ------
Secondary invariants of degree 11 :
  We must search 1 secondary invariants
    Correction by adding the space spanned by secondaries of degree 5
      Research of products of secondaries of degree smaller
        Add product [2, 9]
  ------
Secondary invariants of degree 12 :
  We must search 3 secondary invariants
    Correction by adding the space spanned by secondaries of degree 6
    Correction by adding the space spanned by secondaries of degree 0
      Research of products of secondaries of degree smaller
        Add product [3, 9]
        Add product [2, 2, 8]
        Register new relation : [4, 8]
        Add product [2, 3, 7]
...
...  affiche ici de nouveau la longue liste des invariants secondaires
...
```

On peut aussi demander uniquement les secondaires irréductibles.

```
sage: I.irreducible_secondary_invariants()
{0: [[(0, 0, 0, 0, 0, 0)]],
 1: [],
 2: [[(1, 1, 0, 0, 0, 0)]],
 3: [[(2, 1, 0, 0, 0, 0)]],
```

```
 4: [[(3, 1, 0, 0, 0, 0)]],
 5: [],
 6: [[(3, 2, 1, 0, 0, 0)], [(2, 2, 1, 1, 0, 0)]],
 7: [[(4, 2, 1, 0, 0, 0)]],
 8: [[(5, 2, 1, 0, 0, 0)]],
 9: [[(5, 3, 1, 0, 0, 0)]],
 10: [],
 11: [],
 12: []}
```

MuPAD-Combinat, avec son module PermutationGroupInvariantRing utilisant les bases de SAGBI-Gröbner, retourne exactement les mêmes invariants secondaires irréductibles.

```
 +---+
 | T |             MuPAD-Combinat 1.3.2 (stable)
 +---+---+
 | A | K |         an open source MuPAD package for
 +---+---+---+
     | I | N |    research in Algebraic Combinatorics
     +---+---+

This package provides or extends the following libraries:
  combinat, examples, Dom, Cat, output, experimental, IPC, operators

For quick information on a particular library, please type:
  info(library)   or   ?library   (requires MuPAD >= 4.0.0)

For the full html documentation, please browse through:
  http://mupad-combinat.sf.net/ (project web page)
  file:/opt/MuPAD-4.0.6/packages/Combinat/index.html (local mirror)

>> G := Dom::PermutationGroup(6, [[[1,3,5],[2,4,6]],[[1,4],[2,5]],
                                  [[1,5],[2,4],[3,6]]]):
>> IG := Dom::PermutationGroupInvariantRing(Dom::Rational, G)

>> IG::irreducibleSecondaryInvariants()

table(
  9 = [o([5, 3, 1, 0, 0, 0])],
  8 = [o([5, 2, 1, 0, 0, 0])],
  7 = [o([4, 2, 1, 0, 0, 0])],
  6 = [o([2, 2, 1, 1, 0, 0]), o([3, 2, 1, 0, 0, 0])],
  4 = [o([3, 1, 0, 0, 0, 0])],
  3 = [o([2, 1, 0, 0, 0, 0])],
  2 = [o([1, 1, 0, 0, 0, 0])]
)
```

Bibliographie

[Abd99] Ines Abdeljaouad. Calculs d'invariants primitifs de groupes finis. *Theor. Inform. Appl.*, 33(1) :59–77, 1999.

[Abd00] Ines Abdeljaouad. *Théorie des Invariants et Applications à la Théorie de Galois effective*. PhD thesis, Université Paris 6, 2000.

[BBT11] François Bergeron, Nicolas Borie, and Nicolas M. Thiéry. Deformed diagonal harmonic polynomials for complex reflection groups. *FPSAC 2011*, February 2011. Accepted, 12 pages, arXiv :1011.3654 [math.CO].

[Ber09] François Bergeron. *Algebraic combinatorics and coinvariant spaces*. CMS Treatises in Mathematics. Canadian Mathematical Society, Ottawa, ON, 2009.

[BGW10] F. Bergeron, A. Garsia, and N. Wallach. Harmonics for Deformed Steenrod Operators. *DMTCS proc.*, AN(01) :497–508, 2010. arXiv :0812.3566 [math.CO].

[BT11] Nicolas Borie and Nicolas M. Thiéry. An evaluation approach to computing invariants rings of permutation groups. In *Proceedings of MEGA 2011*, March 2011. Accepted, 8 pages.

[Col97a] Antoine Colin. Solving a system of algebraic equations with symmetries. *J. Pure Appl. Algebra*, 117/118 :195–215, 1997. Algorithms for algebra (Eindhoven, 1996).

[Col97b] 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.

[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.

[DM10] M. D'Adderio and L. Moci. On a conjecture of Hivert and Thiéry about Steenrod operators. *Arxiv preprint arXiv :1010.4985*, 2010.

[DSW09] Xavier Dahan, Éric Schost, and Jie Wu. Evaluation properties of invariant polynomials. *J. Symbolic Comput.*, 44(11) :1592–1604, 2009.

[FR09] J.C. Faugère and S. Rahmany. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. In *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, pages 151–158. ACM, 2009.

[GAP08] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008.

[Gat90] K. Gatermann. *Symbolic solution of polynomial equation systems with symmetry*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1990.

[GK00] Katharina Geissler and Jürgen Klüners. Galois group computation for rational polynomials. *J. Symbolic Comput.*, 30(6) :653–674, 2000. Algorithmic methods in Galois theory.

[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.

[GST06] Pierrick Gaudry, Éric Schost, and Nicolas M. Thiéry. Evaluation properties of symmetric polynomials. *Internat. J. Algebra Comput.*, 16(3) :505–523, 2006.

[Hai03] Mark Haiman. Combinatorics, symmetric functions, and Hilbert schemes. In *Current developments in mathematics, 2002*, pages 39–111. Int. Press, Somerville, MA, 2003.

[HE71] M. Hochster and John A. Eagon. Cohen-Macaulay rings, invariant theory, and the generic perfection of determinantal loci. *Amer. J. Math.*, 93 :1020–1058, 1971.

[Her03] Patricia Hersh. A partitioning and related properties for the quotient complex $\Delta(B_{lm})/S_l \wr S_m$. *J. Pure Appl. Algebra*, 178(3) :255–272, 2003. With an appendix by Vic Reiner.

[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*, 116(4) :844–863, 2009.

[HT04] 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].

[HT09] Florent Hivert and Nicolas M. Thiéry. The Hecke group algebra of a Coxeter group and its representation theory. *J. Algebra*, 321(8) :2230–2258, 2009.

[Hum90] James E. Humphreys. *Reflection groups and Coxeter groups*, volume 29 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1990.

[Kas05] Masaki Kashiwara. Level zero fundamental representations over quantized affine algebras and Demazure modules. *Publ. Res. Inst. Math. Sci.*, 41(1) :223–250, 2005.

[Kem93] Gregor Kemper. The *invar* package for calculating rings of invariants. IWR Preprint 93-94, University of Heidelberg, 1993.

[Kin07a] S. King. Minimal generating sets of non-modular invariant rings of finite groups. *Arxiv math/0703035*, 2007.

[Kin07b] S.A. King. Fast Computation of Secondary Invariants. *Arxiv math/0701270*, 2007.

[Knu73] Donald E. Knuth. *The art of computer programming. Volume 3.* Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973. Sorting and searching, Addison-Wesley Series in Computer Science and Information Processing.

[KV97] Axel Kohnert and Sébastien Veigneau. Using Schubert basis to compute with multivariate polynomials. *Adv. in Appl. Math.*, 19(1) :45–60, 1997.

[Las08] A. Lascoux. Schubert and macdonald polynomials, a parallel. *preprint avaible online : www-igm. univ-mlv. fr/ al/ARTICLES/MsriExp. pdf*, 2008.

[LS82] Alain Lascoux and Marcel-Paul Schützenberger. Polynômes de Schubert. *C. R. Acad. Sci. Paris Sér. I Math.*, 294(13) :447–450, 1982.

[McK98] Brendan D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26(2) :306–324, 1998.

[MN08] C. Monico and M.D. Neusel. Counting special monomials. *Recall (from, eg, Page 34 of [6])*, 10 :0, 2008.

[Pro00] Vincent Prosper. Factorization properties of the q-specialization of Schubert polynomials. *Ann. Comb.*, 4(1) :91–107, 2000.

[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].

[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.

[Rea78] Ronald C. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Ann. Discrete Math.*, 2 :107–120, 1978. Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).

[S+09] W. A. Stein et al. *Sage Mathematics Software (Version 3.3)*. The Sage Development Team, 2009. `http://www.sagemath.org`.

[SCc08] The Sage-Combinat community. Sage-Combinat : enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, 2008. `http://combinat.sagemath.org`.

[Ser65] Jean-Pierre Serre. *Algèbre locale. Multiplicités*, volume 11 of *Cours au Collège de France, 1957–1958, rédigé par Pierre Gabriel. Seconde édition, 1965. Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1965.

[Ser03] Ákos Seress. *Permutation group algorithms*, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2003.

[Sta79] Richard P. Stanley. Invariants of finite groups and their applications to combinatorics. *Bull. Amer. Math. Soc. (N.S.)*, 1(3) :475–511, 1979.

[Stu93] Bernd Sturmfels. *Algorithms in invariant theory*. Springer-Verlag, Vienna, 1993.

[Thi00] 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].

[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., 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.

[Val89] Annick Valibouze. Fonctions symétriques et changements de bases. In *EUROCAL '87 (Leipzig, 1987)*, volume 378 of *Lecture Notes in Comput. Sci.*, pages 323–332. Springer, Berlin, 1989.

[Vei97] Sébastien Veigneau. SP, a package for Schubert polynomials realized with the computer algebra system MAPLE. *J. Symbolic Comput.*, 23(4) :413–425, 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.

Index des notations

$\mathfrak{S}_n$: le groupe symétrique d'ordre n
$\sigma := \langle v \rangle$: permutation σ associée au code de Lehmer v
$v := \mathcal{C}(\sigma)$: code de Lehmer v de la permutation σ
$\langle \sigma, \tau \rangle$: groupe engendré par les permutations σ et τ
$\langle S \rangle_{\mathbb{K}}$: $\mathbb{K}$-espace vectoriel engendré par les éléments de S
$\langle \Theta_1, \ldots, \Theta_n \rangle$: idéal dans $\mathbb{K}[\mathbf{x}]$ engendré par les polynômes $\Theta_1, \ldots, \Theta_n$
$\langle \Theta_1, \ldots, \Theta_n \rangle^G$: idéal dans $\mathbb{K}[\mathbf{x}]^G$ engendré par les polynômes invariants $\Theta_1, \ldots, \Theta_n$
$GL_n(\mathbb{K})$: groupe des matrices inversibles de taille n à coefficient dans le corps $\mathbb{K}$
$GL(V)$: groupe linéaire de l'espace vectoriel V (i.e. ensemble des endomorphismes inversibles de l'espace vectoriel V)
$\beta(\mathbb{K}[\mathbf{x}]^G)$: degré maximal d'un système minimal de générateurs de l'algèbre $\mathbb{K}[\mathbf{x}]^G$
$H(A, z)$: série de Hilbert de l'algèbre A en la variable z
$S(\mathbb{K}[\mathbf{x}]^G, z)$: polynôme énumérateur des degrés des invariants secondaires lorsque les invariants primaires ont été choisis de manière explicite
A_+ : sous-algèbre de A formé des éléments de degré strictement positif
A_d : désigne le sous espace des éléments de degré d de l'algèbre graduée A
$A_{\leqslant d}$: désigne le sous espace des éléments de degré au plus d de l'algèbre graduée A
X_σ : polynôme de Schubert indexé par la permutation σ
Y_v : polynôme de Schubert indexé par le code de Lehmer v
$\mathring{W}$: groupe de Coxeter fini
W : groupe de Coxeter affine construit à partir de la version fini $\mathring{W}$
$\mathbb{C}[W]$: algèbre du groupe W
$H(W)$: algèbre de Hecke du groupe de Coxeter W
$H\mathring{W}$: groupe algèbre de Hecke du groupe de Coxeter fini $\mathring{W}$

Index

0-Hecke algèbre, 89

action
  au niveau 0, 93
  sur les polynômes, 25
  sur les vecteurs d'entiers, 39
algèbre
  de (Iwahori)-Hecke, 91
  de Cohen-Macaulay, 34
  de groupe, 89
  de Hecke, 89
  graduée, 22
  graduée connexe, 23
anneau
  des invariants, 26
  des polynômes, 22

base
  de Gröbner, 15, 57
  de SAGBI-Gröbner, 15, 57
borne sur le degré des générateurs, 31

code
  de Lehmer, 20
  dominant, 75

décomposition de Hironaka, 33
degré d'un groupe de permutations, 20
différence divisée, 74
dimension
  de Krull, 22

ensemble
  minimal de générateurs, 32

forme normale de Smith, 96

graphe
  de Cayley, 90
groupe
  de Coxeter, 89, 90
  de permutations, 20
  de réflexions complexes, 21
  fini de matrices, 21
  linéaire $GL_n(\mathbb{K})$, 21
  symétrique, 19
Groupe algèbre de Hecke, 92
groupe algèbre de Hecke, 89

hauteur d'une racine, 94

invariant
  primaire, 31
  primitif, 29
  secondaire, 34
itérable, 117

lemme
  de Nakayama gradué, 24

monôme
  sous l'escalier, 28
morphisme d'évaluation Φ, 59

opérateur de Reynolds, 26
orbite d'un vecteur d'entiers, 40
ordre
  lexicographique, 41
  lexicographique partiel, 45

partie algébriquement indépendante, 22
permutation, 19
  vexillaire, 78
polynôme
  énumérateur des cycles, 30
  de Schubert, 75
  de Vandermonde, 35
  invariant, 26
  symétrique, 27
  symétrique élémentaire, 27

réflexion complexe, 21
règle de Monk, 76

série
  de Hilbert, 23
  de Molien, 29
  des invariants secondaires, 34

sigma-invariant, 34
somme
    de puissances, 31
    sur orbite, 27
stabilisateur
    d'un polynôme, 28
système
    d'invariants primaires, 31
    d'invariants secondaires, 34
    de générateurs forts, 45
    de paramètres homogènes, 22

théorème
    de Hilbert, 31
    de Molien, 29
    de Noether, 31
    fondamental des polynômes symétriques, 28
transversale d'un sous-groupe, 22
type de Cartan, 89

vecteur canonique, 41

Calcul des invariants de groupes de permutations par transformée de Fourier

Résumé :

Cette thèse porte sur trois problèmes en combinatoire algébrique effective et algorithmique.

Les premières parties proposent une approche alternative aux bases de Gröbner pour le calcul des invariants secondaires des groupes de permutations, par évaluation en des points choisis de manière appropriée. Cette méthode permet de tirer parti des symétries du problème pour confiner les calculs dans un quotient de petite dimension, et ainsi d'obtenir un meilleur contrôle de la complexité algorithmique, en particulier pour les groupes de grande taille. L'étude théorique est illustrée par de nombreux bancs d'essais utilisant une implantation fine des algorithmes. Un prérequis important est la génération efficace de vecteurs d'entiers modulo l'action d'un groupe de permutation, dont l'algorithmique fait l'objet d'une partie préliminaire.

La quatrième partie cherche à déterminer, pour un certain quotient naturel d'une algèbre de Hecke affine, quelles spécialisations des paramètres aux racines de l'unité donne un comportement non générique.

Finalement, la dernière partie présente une conjecture sur la structure d'une certaine q-déformation des polynômes harmoniques diagonaux en plusieurs paquets de variables pour la famille infinie de groupes de réflexions complexes.

Tous ces chapitres s'appuient fortement sur l'exploration informatique, et font l'objet de multiples contributions au logiciel `Sage`.

Mots clés :

Théorie des invariants effective, Combinatoire algébrique, Calcul formel, Groupes de permutations, Groupes de Coxeter, Algèbres de Hecke affine, Polynômes harmoniques, Génération à un isomorphisme près, Exploration informatique.

Calculate invariants of permutation groups by Fourier Transform

Abstract :

This thesis concerns algorithmic approaches to three challenging problems in computational algebraic combinatorics.

The firsts parts propose a Gröbner basis free approach for calculating the secondary invariants of a finite permutation group, proceeding by using evaluation at appropriately chosen points. This approach allows for exploiting the symmetries to confine the calculations into a smaller quotient space, which gives a tighter control on the algorithmic complexity, especially for large groups. The theoretical study is illustrated by extensive benchmarks using a fine implementation of algorithms. An important prerequisite is the generation of integer vectors modulo the action of a permutation group, whose algorithmic constitute a preliminary part of the thesis.

The fourth part of this thesis is determining for a certain interesting quotient of an affine Hecke algebra exactly which root-of-unity specialization of its parameter lead to non-generic behavior.

Finally, the last part presents a conjecture on the structure of certain q-deformed diagonal harmonics in many sets of variables for the infinite family of complex reflection groups.

All chapters proceed widely by computer exploration, and most of established algorithms constitute contributions of the software `Sage`.

Keywords :

Computational Invariant Theory, Algebraic Combinatorics, Computer Algebra, Permutation Groups, Coxeter Groups, Affine Hecke algebras, Harmonic Polynomials, Combinatorial Generation up to an Isomorphism, Computer Exploration.
