Fonctions symétriques et changements de bases

Annick Valibouze
LITP,Université P. et M. Curie, 4 Place Jussieu
75252 Paris Cedex 05
Unité associée au CNRS N.248 et GRECO de Calcul Formel N.60
UUCP: ... mcvax!inria!litp!avb

Abstract

This paper describes change of basis algorithms for symmetric polynomials. We consider below the three usual following bases : monomial forms, symmetric elementary and Newton polynomials. The originality consists in retaining only one representative of the orbit to make the computations. It is a crucial point if we realize that one orbit can contain commonly hundreds of terms. We implemented these algorithms in FRANZLISP with an interface MACSYMA.

INTRODUCTION

Une fonction est dite symétrique, si elle est invariante par toute permutation de ses variables.
Une difficulté apparente de manipulation des fonctions symétriques est le caractère exponentiel des groupes symétriques ($n!$ éléments).
L'algèbre des polynômes symétriques possède entre autres bases les fonctions symétriques élémentaires et les fonctions puissances.
Nous présentons ici des algorithmes de décomposition d'un polynôme symétrique dans chacune de ces deux bases qui évitent ce caractère exponentiel. Pour aboutir à cela il a fallu travailler sur des représentations contractées des polynômes, consistant à ne conserver qu'un élément par orbite.
L'ensemble de ces algorithmes, implantés en Franzlisp, constituent un sous-module de `MACSYMA` que j'ai nommé `SYM`.
Comme les coefficients d'un polynôme sont des fonctions symétriques de ses racines, ces algorithmes peuvent déboucher sur la manipulation des racines d'un polynôme et éviter l'explosion des calculs.
Les algorithmes sont suffisamment performants pour pouvoir calculer des résultants en des temps du même ordre de grandeur que la méthode classique.

1 Définitions et notations

Soient k un corps, R_n l'anneau $k[x_1, x_2, ..., x_n]$ des polynômes à coefficients sur k en les variables $x_1, x_2, ..., x_n$ et X la multivariable $(x_1, x_2, ..., x_n)$.

Pour tout élément σ de S_n (le groupe des permutations d'ordre n) et toute suite T, $(t_1, t_2, ..., t_n)$, on notera $\sigma(T)$ la suite $(t_{\sigma(1)}, t_{\sigma(2)}, \ldots, t_{\sigma(n)})$.

Un polynôme P de R_n est dit *symétrique* , si pour tout élément σ de S_n, $P(X)$ est égal à $P(\sigma(X))$. L'algèbre des polynômes symétriques de R_n apparaît alors comme l'ensemble des invariants $R_n^{S_n}$.

Introduisons maintenant quelques notations pour les fonctions symétriques (voir [Macdonald]) :
Soit I une suite finie $(i_1, i_2, \ldots, i_n)$ d'entiers positifs ou nuls. On définit le monôme X^I comme le

produit $x_1^{i_1}x_2^{i_2}\dots x_n^{i_n}$. Si $i_1 \geq i_2 \cdots \geq i_n$, I est appelée une *partition.* Cette notation des partitions est indépendante du nombre de zeros. Les *parts* de I sont les i_k non nuls. La *longueur* $lg(I)$ de I est le nombre de ses parts. Si $a_1, \dots, a_q$ sont les parts distinctes de I avec $a_1 > a_2 > \cdots > a_q$ et $m_{a_1}, \dots, m_{a_q}$ leur *multiplicité* (i.e. leur nombre d'occurences dans I) respective dans I, on notera également la partition I : $(a_1^{m_{a_1}} a_2^{m_{a_2}} \dots a_q^{m_{a_q}})$. A une partition I on associe la *forme monomiale* $M_I(X)$, somme des monômes de l'orbite de X^I sous l'action de S_n :

$$M_I(X) = \sum_{\sigma \in S_n/G(I)} X^{\sigma(I)}$$

où $G(I)$ désigne le stabilisateur de I sous l'action de S_n. Si la partition I est de longueur strictement supérieure à n, le nombre de variables, on conviendra que $M_I(X)$ est nulle.
Les formes monomiales forment trivialement une base de $R_n^{S_n}$.

La forme monomiale $M_{(1^i)}(X)$ est la $i^{ième}$ *fonction symétrique élémentaire* . On la note $e_i(X)$, avec les conventions $e_0 = 1$ et $e_i = 0$ pour $i > n$. Les fonctions symétriques élémentaires forment une base de l'algèbre $R_n^{S_n}$. Si l'on considère un polynôme unitaire en x de degré n, et X l'ensemble de ses racines, le coefficient de x^i est $(-1)^i e_i(X)$.

Les formes monomiales $p_0 = n$ et $p_i(X) = \sum_{x \in X} x^i$, où i est un entier strictement positif, sont les *fonctions puissances* sur X. Elles forment également une base de l'algèbre $R_n^{S_n}$.

2 Le problème

Nous voulons réaliser les opérations élémentaires de l'algèbre $R_n^{S_n}$ et donner deux algorithmes décomposant un polynôme symétrique sur la base des fonctions symétriques élémentaires et sur celle des fonctions puissances. Confrontés à un problème d'espace, nous coderons les polynômes symétriques pour ne travailler qu'avec un seul monôme par forme monomiale (cf. paragraphe 3). Le point fondamental consiste alors en la formule permettant de calculer le produit de deux formes monomiales (lemme du produit, voir paragraphe 5). Le passage des fonctions symétriques élémentaires aux fonctions puissance, et sa réciproque, sont donnés par les formules classiques de Girard-Newton [Girard]. On les trouve également, ainsi que d'autres, dans [Macdonald, ch.2].

3 Représentation des données

Un polynôme symétrique s'écrit naturellement comme combinaison linéaire sur k de formes monomiales. Il peut donc être codé par une liste de partitions adjointes d'un coefficient élément de k. Soit I une partition. Les deux représentations, $(i_1, \dots, i_n)$ et $(a_1^{m_{a_1}} \dots a_q^{m_{a_q}})$ de I induisent deux codages. Avec la première on dira que le polynôme est *partitionné de type* 1. Par exemple le polynôme $p(X) = 3\, M_{(2,0,0)}(X) + 5\, M_{(1,1,0)}(X)$ est codé par la liste ((3, 2, 0, 0), (5, 1, 1, 0)). Avec la seconde, I est codé par la liste $(a_1, m_{a_1}, a_2, m_{a_2}, \dots, a_n, m_{a_n})$. Notre polynôme p sera alors codé avec la liste ((3, 2, 1), (5, 2, 1)). Un polynôme ainsi codé sera dit *partitionné de type* 2. C'est la représentation partitionnée de type 2 qui est utilisée pour l'implémentation des algorithmes de changements de bases `ELEM` et `PUI` que nous décrirons ici.

On peut ainsi obtenir ainsi un gain de place pouvant aller jusqu'à l'exponentielle de l'ordre n du groupe symétrique.

4 Ordres

Nos algorithmes, `ELEM` et `PUI`, font intervenir des décroissances d'ordres pour, d'une part, ne jamais faire le même calcul deux fois, et d'autre part, s'assurer de leur convergence. Pour cela nous considérons deux suites finies d'entiers $u = (u_i)_{1 \leq i \leq n}$ et $v = (v_i)_{1 \leq i \leq n}$.

4.1 Ordre lexicographique : $<$

Il sera associé à la décomposition en les fonctions symétriques élémentaires. On dit que u est inférieur à v pour l'*ordre lexicographique*, noté ici $u < v$, s'il existe un entier k compris entre 1 et n tel que u_k soit strictement inférieur à v_k et que pour tout entier p inférieur à k on ait $u_p = v_p$.
Cet ordre définit naturellement la notion de suite ordonnée lexicographiquement pour les suites produits $u \times v = ((u_p, v_p))_{1 \leq p \leq n}$.
Restreint aux partitions de $\mathbf{N}^n$ il induit également un ordre total sur les formes monomiales de $R_n^{S_n}$: $M_I < M_J$ si $I < J$.

Remarque : Les plus petites formes monomiales pour cet ordre sont les fonctions symétriques élémentaires avec : $e_1 < e_2 < \cdots < e_n$.

4.2 Ordre des longueurs : $<_{lg}$

Il sera associé à la décomposition en les fonctions puissances. On dira que u est inférieur à v pour l' *ordre des longueurs* si $lg(u) < lg(v)$ ou si $lg(u) = lg(v)$ et $u < v$. On notera alors $u <_{lg} v$.

Remarque : Les plus petites formes monomiales pour cet ordre sont les fonctions puissances avec : $p_1 <_{lg} p_2 <_{lg} \cdots <_{lg} p_n <_{lg} \ldots$

5 Produit de deux formes monomiales

C'est la formule calculant ce produit qui va permettre de réaliser nos manipulations en conservant le codage, que nous avons appelé partitionné, évitant ainsi l'explosion exponentielle des développements.
Soient $u = (u_i)_{1 \leq i \leq n}$ et $v = (v_i)_{1 \leq i \leq n}$ deux n-uplets.
Le partition engendrée par une permutation de u sera notée $P(u)$. Désignons par $u \times v$ et $u + v$ les suites $((u_i, v_i))_{1 \leq i \leq n}$ et $(u_i + v_i)_{1 \leq i \leq n}$ respectivement. On dira que $u \times v$ est dans l'*ordre lexicographique* sur $(\mathbf{N}^2)^n$ si pour tout $1 \leq i < n$ ou $u_i > u_{i+1}$, ou bien $u_i = u_{i+1}$ et $v_i \geq v_{i+1}$.
On définit l'action de S_n sur les n-uplets de couples de la manière suivante :

$$\begin{array}{rcl} S_n \times (\mathbf{N}^2)^n & \longrightarrow & (\mathbf{N}^2)^n \\ (\sigma \; , \; (c_1, \ldots, c_n)) & \longmapsto & (c_{\sigma(1)}, \ldots, c_{\sigma(n)}) \end{array}$$

où pour $c_i = (a_i, b_i)$ on a $c_{\sigma(i)} = (a_{\sigma(i)}, b_{\sigma(i)})$.

Le lemme du produit : *Soit X la multivariable $(x_1, \ldots, x_n)$. Pour toute partition J et K on a :*

$$M_J(X) M_K(X) = \sum_{L \in Ad(J,K)} c_L \; M_{P(J+L)}(X) \tag{1}$$

où $Ad(J,K) = \{\sigma(K) \mid \sigma \in S_n \; , \; J \times \sigma(K)$ est dans l'ordre lexicographique$\}$ et c_L est un entier qui peut se calculer de deux manières :

- *Soit $P(J+L) = (\lambda_1^{m_{\lambda_1}} \lambda_2^{m_{\lambda_2}} \ldots \lambda_q^{m_{\lambda_q}})$. Soit i compris entre 1 et q. Les m_{λ_i} parts λ_i de $P(J+L)$ résultent de la somme d'une sous-suite $u_i(J)$ de J et d'une sous-suite $u_i(L)$ de L. En désignant* *par $c_i(J)$ (resp. $c_i(L)$) le cardinal de l'orbite de $u_i(J)$ (resp. $u_i(L)$) sous l'action de $S_{m_{\lambda_i}}$, il vient :*

$$c_L = \prod_{i=1}^{q} c_i(J) = \prod_{i=1}^{q} c_i(L). \tag{2}$$

- *La seconde égalité est donnée par le quotient de deux cardinaux de stabilisateurs :*

$$c_L = \frac{\#G(J+L)}{\#G(J \times L)}. \tag{3}$$

L'identité (2) mieux adaptée à l'algorithme `ELEM` nécessitera une représentation partitionnée de type 2 et la (3) proposée par Daniel LAZARD est utilisée dans le module `SYM` pour le produit de deux polynômes symétriques.

Remarques : K est dans l'ensemble d'admissibilité $Ad(J,K)$ puisque $J \times K$ est ordonnée lexicographiquement et $P(J+K)$ est égale à $J+K$ puisque J et K sont des partitions.

Propriété 5.1 *$P(J+L) <_{lex} J+K$ pour tout L dans $Ad(J,K)$.*

En effet, J et K étant deux partitions, on peut utiliser le lemme suivant :

Lemme : *Soient deux partitions $I = (i_1, ..., i_n)$ et $J = (j_1, ..., j_n)$, et s et t deux permutations de S_n, alors $P(I+J)$ est supérieur à $P(s(I)+t(J))$ relativement à l'ordre lexicographique.*

Démonstration par récurrence sur n : Pour $n=1$, c'est clair. Sinon on regarde le maximum des $i_{s(k)} + j_{t(k)}$ pour k variant entre 1 et n, atteint disons pour l'indice q. De deux choses l'une : Ou bien ce maximum est strictement inférieur à $i_1 + j_1$, et alors la conclusion est assurée ; ou bien il est égal à $i_1 + j_1$, et alors $i_1 = i_{s(q)}$ et $j_1 = j_{t(q)}$. On considère alors les deux suites ordonnées $(i_2, ..., i_n)$ et $(j_2, ..., j_n)$, et leurs permutations : $i_{s(1)}, ..., i_{s(n)}$ et $j_{t(1)}, ..., j_{t(n)}$ où on a enlevé respectivement $i_{s(q)}$ et $j_{t(q)}$. Par hypothèse de récurrence on conclut.

Propriété 5.2 *Le coefficient c_K associé à K est égal à 1.*

On prend $J = (j_1, \ldots, j_n)$ et $K = (k_1, \ldots, k_n)$. Comme le coefficient associé à J est le quotient de $\#G(J+K)$ avec $\#G(J \times K)$, et que $\#G(J+K) \geq \#G(J \times K)$, pour qu'il soit égal à 1 on doit montrer l'inégalité inverse, ou ce qui revient au même, que l'égalité de deux parts de $J+K$ entraine l'égalité des deux parts de $J \times K$ qui les ont engendrées. Si on a $j_q + k_q = j_r + k_r$ avec $q < r$, ou bien $k_r = k_q$ ou bien $k_q > k_r$ puisque K est une partition. Le premier cas implique que $j_q = j_r$. Et le deuxième viendrait de $j_q < j_r$, ce qui est exclu puisque J est une partition. On voit ainsi que $(j_q, k_q) = (j_r, k_r)$.

Propriété 5.3 *J , $K <_{lex} P(J+L)$ pour toute suite L de $Ad(J,K)$.*

Ceci est totalement évident pour J, et aussi pour K par symétrie.

6 Décomposition en les symétriques élémentaires

Soit P un polynôme de $R_m^{S_m}$ de degré d. Notons p le plus petit des deux nombres d et m. Il s'agit de construire le polynôme f de p variables sur k, tel que :

$$P(x_1, x_2, \ldots, x_m) = f(e_1(X), ..., e_p(X))$$

6.1 Principe de la décomposition

Pour réaliser cette décomposition on regarde le lemme du produit énoncé précédemment. On en déduit ainsi les formules (4) et (5) permettant de réécrire une forme monomiale avec d'autres strictement plus petites pour l'ordre lexicographique. Or on obtient exactement la règle dont on a besoin puisque toute fonction symétrique élémentaire minore pour cet ordre toute forme monomiale non symétrique élémentaire.

Soit I une partition de longueur non nulle. Pour toute décomposition de I en somme non triviale $J + K$ de partitions de longueurs non nulles, on a d'après le lemme du produit et la propriété 5.2 :

$$M_I = M_K\, M_J - \sum_{L \in Ad^*(J,K)} c_L\, M_{P(J+L)} \tag{4}$$

où $Ad^*(J,K) = Ad(J,K) \backslash \{K\}$ et dont tout élément L vérifie, d'après les propriétés 5.1 et 5.3 :

$$J <_{lex} P(J+L) <_{lex} I. \tag{5}$$

A partir de la formule (4) on voit se dessiner au moins deux stratégies. La première que nous décrivons ci-dessous est une méthode linéaire et la seconde serait une méthode dichotomique où chaque part i_p de I serait divisée comme suit :

$$\begin{aligned} j_p = k_p &= \frac{i_p}{2} && \text{si } i_p \text{ paire} \\ j_p = k_p - 1 &= \lfloor \frac{i_p}{2} \rfloor && \text{si } i_p \text{ impaire} \end{aligned}$$

Dans les deux cas il est essentiel de ne décomposer chaque partition qu'une seule fois.

6.2 L'algorithme linéaire

Considérons le polynôme symétrique P que l'on désire décomposer. Pour éviter de décomposer plusieurs fois la même forme monomiale, on ordonne P dans l'ordre lexicographique décroissant sur les formes monomiales qui le constituent. Soit M_I sa forme monomiale dirigeante affectée de son coefficient c dans k. Notons n la longueur de la partition I : $(i_1, i_2, ..., i_n, 0, 0, ..., 0)$. Si M_I n'est pas déjà symétrique élémentaire, on peut l'écrire comme la somme de $J = (i_1 - 1, i_2 - 1, ..., i_n - 1)$ et de $K = (1^n)$ (i.e $M_K = e_n$). La décomposition (4) devient alors :

$$M_I = e_n M_J - \sum_{L \in Ad^*(J,K)} c_L\, M_{P(J+L)} \tag{6}$$

où le second membre, $Q(M_I)$, est considéré comme un polynôme de $\mathbf{k}[e_n][x_1, x_2, ..., x_m]^{S_m}$. On remplace P par $P1$, somme de $P - cM_I$ avec $cQ(M_I)$, que l'on ordonne comme P. Les inégalités (5) assurent que les formes monomiales de $P1$ sont strictement plus petites que M_I. Il suffit maintenant de réitérer ce processus en considérant que les coefficients sont dans l'anneau $k[e_1, \ldots, e_p]$. L'ordre lexicographique étant un bon ordre le processus s'arrête lorsque la forme monomiale de tête est une fonction symétrique élémentaire (voir la remarque sur l'ordre lexicographique).

Remarque : Dans ce cas particulier le calcul du produit d'une forme monomiale avec une fonction symétrique élémentaire est réalisable par un algorithme, `MULTELEM` (voir 6.2.2), qui

- ne commence jamais à construire une solution qui s'avèrerait ne pas appartenir à l'ensemble admissible $Ad(J,K)$ et qui serait alors éliminée en cour de calcul ;
- permet de ramener le polynôme, membre droit de (4), déjà ordonné dans l'ordre lexicographique avec I en tête qu'il sera alors aisé de retirer.

6.2.1 Algorithme ELEM

On suppose que e_i prend la valeur `ei`. Le symbole $\star$ entre un élément d de $\mathbf{k}[e_1, ..., e_p]$ et un polynôme symétrique exprimé sur la base des formes monomiales signifie que l'on multiplie tout les coefficient du polynôme par d.

définition : ELEM
Départ :
LIRE(P)
ORDONNER(P) dans l'ordre lexicographique décroissant.
DECOMPOSER(P)

définition : DECOMPOSER(P)
P est un polynôme symétrique trié dans l'ordre lexicographique décroissant.
M_I est sa forme monomiale de tête,et c son coefficient dans $\mathbf{k}[e_1, ..., e_p]$.
SI M_I est une fonction symétrique élémentaire
ALORS RENDRE P en remplaceant chaque M_I par $e_{lg(I)}$.
SINON
Q := DECOMP(M_I)$\star$ c
P := SOMME(Q,P-c$\star M_I$, lexico)
DECOMPOSER(P)

définition : SOMME(P1,P2,ordre)
P1 et P2 étant 2 polynômes rangés suivant l'ordre décroissant,
on ramène leur somme de nouveau rangée dans l'ordre décroissant.

définition : DECOMP(M_I)
Soient J la partition obtenue en enlevant 1 à chaque part de I,
n la longueur de I.
R := MULTELEM(M_J,n)
R' := M_I-R
RAMENER(R' + en$\star M_J$)

D'après la remarque précédente, R est ordonné dans l'ordre lexicographique décroissant. Comme pour chaque L dans $Ad^*(J, K)$ on a $P(J + L) <_{lex} I$, M_I est en tête de R, et R' est facile alors à récupérer. Les inégalités $J <_{lex} P(J + L)$ permettent de constater qu'il suffit de mettre M_J en queue de R' pour que $Q(M_I)$ soit ordonné dans l'ordre lexicographique décroissant.

6.2.2 Description de MULTELEM

Soient J et K les partitions de l'algorithme linéaire. Nous raffinons, tout d'abord, la contrainte d'admissibilité du lemme du produit, dans ce cas où $M_K = e_n$.
Etendons aux suites la notation $(a_1^{m_1} \dots a_r^{m_r})$, réservée habituellement aux partitions, qui signifiera ici que les m_1 premiers éléments sont égaux à a_1, que les m_2 suivants sont égaux à a_2 et ainsi de suite.

Propriété 6.1 *Soit L une permutation de K où M_K est une fonction symétrique élémentaire. La suite L est admissible (i.e. dans $Ad(J, K)$) si et seulement si $P(J + L)$ est égale à $J + L$.*

En effet si $J = (a_1^{m_1} \dots a_q^{m_q})$, comme les parts de K sont 0 ou 1, les suites admissibles sont celles du type :

$$(1^{n_1} 0^{m_1 - n_1} 1^{n_2} 0^{m_2 - n_2} \dots 1^{n_i} 0^{m_i - n_i} \dots 1^{n_q} 0^{m_q - n_q} 1^{n - (n_1 + \dots + n_q)}),$$

tel que n_i soit un entier naturel compris entre 0 et m_i et que $n-(n_1+\cdots+n_q)$ soit positif ou nul. Ainsi chaque part j_r+l_r d'une suite $J+L$ où L est admissible, ne peut-être qu'inférieure ou égale à la part $j_{r-1}+l_{r-1}$ qui la précède, si elle existe. Soit $P(J+L)=J+L$. Inversement on ne peut avoir $P(J+L)=J+L$ si L n'a pas la forme que nous venons de donner.

Maintenant pour réaliser le produit e_nM_J, on construit les diagrammes représentatifs [Macdonald, p.1] de toutes les partitions $P(J+L)$, où $L\in Ad(J,K)$, de la manière suivante : on rajoute n cases dans le diagramme de J sans jamais en mettre deux sur la même ligne (toutes les parts de K valent 1 ou 0) et de sorte que le diagramme final représente bien une partition qui est ici la contrainte d'admissibilité dans $Ad(J,K)$.
En remplissant d'abord les lignes représentant les plus grands exposants, on obtient les plus grandes partitions $J+L$ en premier et on peut tronquer les calculs afin de ne pas dépasser m le nombre de variables. Ceci justifie le fait que le polynôme symétrique `R` de `DECOMP` soit ordonné dans l'ordre lexicographique décroissant.
Les coefficients c_L sont alors calculés au fur et à mesure de la construction à l'aide de l'égalité (2) du lemme du produit.

6.2.3 Exemple : réduction d'une fonction puissance

Décomposons le polynôme symétrique $P(X)=p_3(X)$ en suivant l'algorithme `ELEM`. Pour plus de clarté nous restons sur la base des formes monomiales au lieu de prendre des représentations partitionnées de type 2 et nous évitons les troncatures dues au dépassement de cardinalité en supposant que X a au moins 3 variables. Au départ

$$P(X)=M_{(3)}(X).$$

Avec `MULTELEM` appliquée à $J=(2)$ et e_1 on obtient $M_{(3)}=e_1M_{(2)}-M_{(2,1)}$. Ce qui donne pour `Q` $-M_{(2,1)}+e_1M_{(2)}$ et `P` est remplacé par :

$$-M_{(2,1)}+e_1M_{(2)}.$$

La forme monomiale de tête de P est à présent $M_{(2,1)}$ et son coefficient est -1. En appliquant `MULTELEM` à $J=(1)$ et e_2 on obtient : $M_{(2,1)}=e_2M_{(1)}-3M_{(1,1,1)}$. Ainsi `Q` est égal à $3\,M_{(1,1,1)}-e_2M_{(1)}$ et `P` est remplacé par :

$$e_1M_{(2)}+3\,M_{(1,1,1)}-e_2M_{(1)}.$$

On continue : Comme $M_{(2)}=e_1M_{(1)}-2M_{(1,1)}$, `P` est remplacé par :

$$3M_{(1,1,1)}-2e_1M_{(1,1)}+(-e_2+e_1^2)M_{(1)}$$

Comme $M_{(1,1,1)}$ est une fonction symétrique élémentaire les autres formes monomiales à décomposer dans `P` le sont également. En remplaçant chaque M_I par $e_{lg(I)}$ dans `P` on obtient finalement :

$$p_3=3e_3-2e_1e_2+(-e_2+e_1^2)e_1=e_1^3-3e_1e_2+3e_3$$

6.2.4 Comparaison avec la méthode de Waring

Nous comparons ici la fonction `ELEM` et celle liée à la méthode de Waring [Dubrueil] écrite en Macsyma dans le livre [S,D,T] portant sur un alphabet de 3 lettres. La méthode de Waring utilise la forme développée des polynômes symétriques et non sa représentation partitionnée. Une des raisons de l'écart de temps entre les deux méthodes est le fait que même si par la méthode de Waring on trouve dès le départ le monôme de $k[e_1,\ldots,e_p]$ qui comporte la plus grande forme monomiale du polynôme symétrique que l'on réduit, une fois construite on la développe pour la soustraire à ce polynôme. On remarque que ce monôme de $k[e_1,\ldots,e_p]$ serait celui obtenu si la fonction `DECOMP` de `ELEM` ne rendait que e_nM_J.
La comparaison porte sur les fonctions puissances par variation de d, la puissance :

d	**WARING**	**ELEM**
12	2 mn 37 s	24 s
15	8 mn 18 s	55 s
18	18 mn 48 s	2 mn
20	31 mn	3 mn
21	39 mn	3 mn 42

Si le cardinal de l'alphabet augmente l'écart de temps s'amplifie avec le caractère exponentiel de son groupe symétrique.
La place prise par les polynômes dans Waring empêche la résolution de certains problèmes comme celui proposé par P. Cartier (voir plus loin) faisant intervenir des orbites de cardinalité 7!.

7 Décomposition en les fonctions puissances

Soit $R = \mathbf{Z}[x_1, \ldots, x_m]$ et $P \in R^{S_m}$ de degré d. Il s'agit de construire le polynôme F de $\mathbf{Q}[x_1, x_2, ..., x_d]$ tel que :

$$P(X) = F(p_1(x), p_2(X), ..., p_d(X))$$

Dans la pratique si $p = inf(m, d)$, P s'écrit en fonction des p premières fonctions puissances. Ceci résulte de la dépendance algébrique des fonctions puissances d'ordres strictement supérieures à m en fonction des m premières.

7.1 Principe sur les formes monomiales

Le procédé, identique à celui utilisé pour les fonctions symétriques élémentaires, est fondé sur la décroissance suivant l'ordre des longueurs dont les formes monomiales minorantes sont les fonctions puissances (cf. paragraphe 4.2), et sur l'utilisation du produit de deux formes monomiales.

Soit I une partition $(i_1, \ldots, i_m)$ de longueur n non nulle. On désigne par σ_p la transposition entre le premier et le $p^{ième}$ élément. Soit r un entier quelconque compris entre 1 et n. Nous utilisons le lemme du produit avec K et J les partitions (i_r) (i.e. $M_K = p_{i_r}$) et $(i_1, \ldots, i_{r-1}, i_{r+1}, \ldots, i_m)$ respectivement, obtenant ainsi $I = P(J + \sigma_m(K))$ avec $c_{\sigma_m(K)} = m_{i_r}(I)$ et donc :

$$M_I = \frac{p_{i_r}\, M_J - \sum c_{\sigma_q(K)}\, M_{P(J+\sigma_q(K))}}{m_{i_r}(I)}, \tag{7}$$

où la sommation est étendue à tout les entiers q compris entre 1 et $m-1$ tels que $i_{q-1} > i_q$, qui est ici la contrainte d'admissibilité.
Comme pour tout entier q inférieur à $m-1$ on a $lg(I)-1 = lg(J) = lg(P(J+\sigma_q(K)))$, la décroissance stricte suivant l'ordre des longueurs est assurée :

$$J <_{lg} P(J + \sigma_q(K)) <_{lg} I.$$

Propriété 7.1 *: Si $r = 1$ alors pour tout entier q de la sommation on a : $c_{\sigma_q(K)} = 1$.*

Preuve : Par convention 0! vaut 1. Nous utilisons l'égalité (3) du lemme du produit donnant les coefficients $c_{\sigma(K)}$. Dans le cas où $r = 1$, pour chaque permutation σ_q de la sommation on a :
$P(J + \sigma_q(K)) = (i_1 + i_q, i_2, i_3, \ldots, i_{q-1}, i_{q+1}, \ldots, i_m)$. Comme la part $i_1 + i_q$ est strictement plus grande que les autres et que $i_{q-1} > i_q$, le cardinal du stabilisateur de $P(J + \sigma(K))$ est donné par :

$$\#G(P(J + \sigma_q(K))) = 1!(m_{i_1} - 1)!m_{(i_1-1)}!m_{(i_1-2)}! \ldots (m_{i_{q-1}})!(m_{i_q} - 1)!(m_{(i_q-1)})! \ldots m_0!.$$

Par ailleurs, comme $J + \sigma_q(K) = ((i_2, 0), (i_3, 0), \ldots, (i_{q-1}, 0), (i_q, i_1), (i_{q+1}, 0), \ldots, (i_m, 0))$, on conclut en constatant l'égalité de $\#G(P(J + \sigma_q(K)))$ et de $\#G(P(J \times \sigma(K)))$ qui est égal à

$$(m_{i_1} - 1)!m_{i_1-1}! \ldots (m_{i_{q-1}})!1!(m_{i_q} - 1)!(m_{(i_q-1)})! \ldots m_0!.$$

7.2 Algorithme PUI

Comme pour la réduction en les fonctions symétriques élémentaires on réalise le produit d'une forme monomiale par une fonction puissance avec un algorithme, que je nomme `MULTPUI`, adapté à ce cas particulier de produit de formes monomiales. Il est implanté dans le cas $r = 1$ pour lequel tout les coefficients sont égaux à 1 (cf. propriété 7.1) et qui permet d'obtenir aisément les partitions $P(J+L)$ admissibles rangées suivant l'ordre des longueurs décroissant.
L'algorithme `PUI` décomposant un polynôme symétrique en les fonctions puissances découle de ce qui précède et se calque sur celui donné pour les fonctions symétriques élémentaires en :

- substituant l'ordre des longueurs à l'ordre lexicographique
- substituant `MULTPUI` à `MULTELEM`
- remplaçant le test d'arrêt sur une fonction symétrique élémentaire par celui sur une fonction puissance
- remplaçant `Q := DECOMP(`M_I`)`$\star c$ par `Q := DECOMP(`M_I`)`$\star c/m_{i_1}$.

8 Applications

Une première application est la solution d'un problème proposé par Cartier. La seconde découle d'un autre proposé par Barrucand. Nous ne donnons pas ici les méthodes utilisées pour réaliser ce type de calculs. Elles feront l'objet d'un autre article relatif aux problèmes d'élimination. Les exécutions ont été réalisées sur le VAX 780 du LITP, en utilisant `MACSYMA` et des programmes écrits en `FRANZLISP` inclus dans le module de manipulation de fonctions symétriques, `SYM`.

8.1 Problème de Cartier

On se donne le polynôme $x^7 - 7x + 3$ dont les racines sont $x_1, x_2, ..., x_7$ et on cherche le polynôme de degré 35 dont les racines sont les sommes 3 à 3 des x_i.

Ce problème fait intervenir des polynômes symétriques de degré 35 à réduire ou bien avec `ELEM` ou bien avec `PUI`. Les temps d'éxécutions sont de 2mn 30 avec `PUI` et de 27 mn avec `ELEM`.

Remarque: L'utilisation des fonctions puissances s'est avérée bien plus efficace que celle des fonctions symétriques élémentaires. Et ceci, aussi bien pour ce qui est de la complexité en temps et en espace, que pour la combinatoires liée à ce problème. Il semblerait que cette constatation soit d'ordre général.

8.2 Problème de Barrucand

Le calcul d'un discriminant dépendait du problème suivant : Etant donnés les polynômes

$$\begin{aligned} P(x) &= x^5 - e_1\, x^4 + e_2\, x^3 - e_3\, x^2 + e_4\, x - e_5 \\ Q(x) &= -55x^4 + 52e_1x^3 + (-40e_2 - 2e_1^2)x^2 + (64e_3 - 16e_1e_2 + 4e_1^3)x \\ &\quad - 64e_4 + 16e_2^2 - 8e_1^2e_2 + e_1^4, \end{aligned}$$

on cherche à calculer le produit $S = Q(x_1)Q(x_2)Q(x_3)Q(x_4)Q(x_5)$, où x_1, x_2, x_3, x_4, x_5 sont les racines de P.
On peut obtenir S comme le résultant de P et Q en x. On peut également le voir comme la cinquième fonction symétrique élémentaire en les $Q(x_1), Q(x_2), Q(x_3), Q(x_4), Q(x_5)$. Ce calcul peut se faire en passant par les fonctions puissances et nécessite 78 secondes dont 7 de "garbage collector", alors que l'utilisation du résultant de MACSYMA donne le résultat en 77 secondes dont 13 de "garbage

collector".

Ce calcul permet de constater :

1. l'efficacité des fonctions puissances, même pour obtenir les fonctions symétriques élémentaires
2. l'efficacité des algorithmes de manipulations des fonctions symétriques qui n'est pas très loin de celle du résultant.

Références

[Dubreuil], P. Dubreuil, Algèbre , Cahiers scientifiques, fascicule XX, *Gauthier - Villard.*

[Girard],(1629), Invention Nouvelle en Algèbre, Amsterdam.

[Macdonald], I.G. Macdonald,(1979), Symmetric functions and Hall polynomials , *Clarendon Press*, Oxford.

[S,D,T], Y. Siret, J.H. Davenport et E. Tournier, (1986), Calcul Formel, Systèmes et algorithmes de manipulations algébriques, *Masson.*