Gröbner Bases, Ideal Computation and Computational Invariant Theory

Supervised by Kazuhiro Yokoyama

Tristan Vaccon

September 10, 2010

Contents

Preamble

My internship took place at Rikkyo University (Ikebukuro, Tokyo, Japan) between the first of June and the 31st of July 2010. It was supervised by Prof. Kazuhiro Yokoyama. I had my desk at the graduate students's room and twice a week, I had to give a one-hour presentation in front of Prof. Yokoyama and sometime some of his students and post-docs.

My path to Computational Invariant Theory during the internship was really close to the one given in this report.

Introduction

We will indeed first recall some basic knowledge about algebraic geometry, and then, some basic knowledge about Gröbner bases. The main highlight of those two first parts should be the very elementary proof of Hilberts Nullstellensatz we will give.

After that, we will study Computational Ideal Theory, *id est* how operations on ideals can be computed (with the help of Gröbner bases), and finally, Computational Invariant Theory. We will also consider implementation of the algorithms we will study. All the implementation I made were in Magma, and they all can be found in the Annex.

Remark. Unlike what might be used outside France, in what follows, $\mathbb{N}$ will denote the set of natural numbers, and $\mathbb{N}^*$ the set of positive natural numbers.

1 Algebraic Geometry

In this first section, our main goal would be to prove Hilbert's Nullstellensatz, and present the links between algebra and geometry. Since it is very basic knowledge on algebraic geometry, many proofs (either obvious or very classical) will not be given. Yet, most of them can be found in *Cox, Little & O'Shea* [1].

1.1 Affine Varieties

1.1.1 Definition

Definition 1. Let k be a field and let $F \subset k[x_1, \ldots, x_n]$ ($n \in \mathbb{N}^*$). Then we call

$$V(F) = \{(a_1, \ldots, a_n) \in k^n / \forall f \in F, F(a_1, \ldots, a_n) = 0\}$$

the **affine variety** defined by F.

Remark. We have $V(F) = V(I(F))$ where $I(F)$ is the ideal spanned by F.

1.1.2 Ideals and Affine Varieties

Definition 2. Let I be an ideal of $k[x_1, \ldots, x_n]$, then $\{f_1, \ldots, f_s\} \subset I$ is called a **basis** of I if $\langle f_1, \ldots, f_s \rangle = I$.

Definition 3. Let A be a commutative ring. A is called **noetherian** if it satisfies the ascending chain condition : given any chain of ideals of A, $I_1 \subset \cdots \subset I_k \subset I_{k+1} \subset \ldots$, then there exists $n \in \mathbb{N}^*$ such that $I_n = I_{n+1} = \ldots$. It is equivalent to the fact that every non-empty set of ideals of A has a maximal element.

Proposition 1. *A is noetherian if and only if all ideals of A are finitely generated.*

Theorem 1 (Hilbert's Theorem)**.** *If A is noetherian, then $A[X]$ is noetherian. Thus, $\forall n \in \mathbb{N}^*$, $A[X_1, \ldots, X_n]$ is noetherian.*

Remark. A field is noetherian (it contains only two ideals ...), thus if k is a field, then $\forall n \in \mathbb{N}^*$, $k[X_1, \ldots, X_n]$ is noetherian.

Definition 4. Let $n \in \mathbb{N}^*$ and $V \subset k^n$. We set

$$I(V) = \{f \in k[X_1, \ldots, X_n] \diagup \forall (a_1, \ldots, a_n) \in V, f(a_1, \ldots, a_n) = 0\}$$

Lemma 1. *$I(V)$ is an ideal of $k[X_1, \ldots, X_n]$, and is called the ideal generated by V.*

Lemma 2. *If $I \subset J$ are ideals of $k[X_1, \ldots, X_n]$ then $V(I) \supset V(J)$.*

Lemma 3. *If $V \subset W \subset k^n$, then $I(V) \supset I(W)$.*

Lemma 4. *If J is an ideal of $k[X_1, \ldots, X_n]$ then $J \subset I(V(J))$.*

Now that we have recalled those basic definitions, we shall give a proof of Hilbert's Nullstellensatz, which might be the shortest basic proof (*id est* that does not need some more advanced mathematics) currently known that stands for any algebraically closed field.

1.2 Hilbert's Nullstellensatz

1.2.1 Preliminary work

Definition 5. Let A be a ring. B is a *finitely generated A-algebra* if there exists $n \in \mathbb{N}^*$ and an ideal I of $A[X_1, \ldots, X_n]$ so as $B \simeq A[X_1, \ldots, X_n]/I$. If so, B is called *integral* if any element of B is integral over A, *id est* for all $x \in B$, there exists a monic $f \in A[X]$ with $f(x) = 0$.

Lemma 5. *Let A and B be two rings with B integral over A. Then if B is a field, A is also a field.*

Proof. Let $x \in A \setminus \{0\}$. $y = x^{-1} \in B$ is integral over A, so there exists a monic $f \in A[X]$ with $f(y) = 0$, *id est* there exists $n \in \mathbb{N}^*$ and $a_0, \ldots, a_{n-1}$ with $y^n + a_{n-1}y^{n-1} + \cdots + a_0 = 0$. Multiplying by x^{n-1} we obtain $y = -(a_{n-1} + \cdots + a_0 x^{n-1})$, and thus, $y \in A$. □

Proposition 2. *Let k be a field, and B a k-algebra finitely generated. If B is a field, then B is a finite algebraic extension of k.*

Proof. We proceed by induction. For $n \in \mathbb{N}$, let $P(n)$ be the proposition "If k is a field and B a k-algebra generated by n elements, and which is a field, then B is a finite algebraic extension of k."

$P(0)$ is true, since in that case $B = k$. If $n = 1$, then we have $B = k[x_1]$ for some x_1 in B. If x_1 is transcendental over A, then $B \simeq k[X]$, which is not a field. So, x_1 is algebraic over A, and then, B is a finite algebraic extension of k.

Now, we assume that $n > 1$ and $P(n-1)$ is true. Let B be a k-algebra generated by n elements, so $B = k[x_1, \ldots, x_n]$ for some $x_1, \ldots, x_n$ in B. Let $A = k[x_1]$ and let K be the fraction ring of A. Since B is a field and $A \subset B$, then $K \subset B$. Thus, B is a K-algebra generated by $n-1$ elements. With $P(n-1)$, B is a finite algebraic extension of K. So $x_2, \ldots, x_n$ are algebraic over K. Hence, there exist $P_i \in K[X]$, monics, so that $P_i(x_i) = 0$, $2 \leq i \leq n$. Let $f \in k[x_1]$ be the product of the denominators of the coefficients of the P_i. Then $x_2, \ldots, x_n$ are integral over $A_f = A[1/f]$, and since its generator are integral over A_f, so is B. Since B is a field, $A_f \subset B$. With the previous lemma, A_f is a field.

If x_1 is transcendental over k, then $A_f = k[X][1/P]$ for some $P \in k[X]$, non-zero. Yet, $1 - XP$ is a non-zero polynomial in $k[X]$ and thus in A_f, but it has no inverse in A_f : if there exists $Q \in A_f$ so as $Q(1 - XP) = 1$, then the evaluation in $X = 1/P$ leads to an absurdity. So, $1 - XP$ has no inverse in A_f while A_f is a field, which is absurd.

Thus, x_1 is algebraic over k. It follows that K is is a finite algebraic extension of k, and since B is a finite algebraic extension of K, we can deduce the result. □

1.2.2 Weak form

Theorem 2. *Let $n \in \mathbb{N}^*$ and I be a maximal ideal of $k[X_1, \ldots, X_n]$, where k is an algebraically closed field. Then there exists $(a_1, \ldots, a_n) \in k^n$ such that $I = \langle X_1 - a_1, \ldots, X_n - a_n \rangle$.*

Proof. First, we shall prove that the ideals of the form $J = \langle X_1 - a_1, \ldots, X_n - a_n \rangle$, with $(a_1, \ldots, a_n) \in k^n$, are maximals. Obviously, $\phi : k[x_1, \ldots, x_n] \to k$, defined

by $\phi(P) = P(a_1, \ldots, a_n)$ is a surjective ring homomorphism. If $P \in k[X_1, \ldots, X_n]$ and if we apply the euclidean division of P by $X_1 - a_1$ in $k[X_2, \ldots, X_n][X_1]$, and then the euclidean division of the remainder (which lies in $k[X_2, \ldots, X_n]$) by $x_2 - a_2$, in $k[X_3, \ldots, X_n][X_2]$, ..., we obtain $P = (X - a_1)Q_1 + \cdots + (X_n - a_n)Q_n + P(a_1, \ldots, a_n)$, with $Q_j \in k[X_j, \ldots, X_n]$. So if $P \in \text{Ker}\phi$, then $P \in J$, and $\text{Ker}\phi \subset J$. Of course, $J \subset \text{Ker}\phi$, and $\text{Ker}\phi = J$, $k[X_1, \ldots, X_n]/J = k$ and J is a maximal ideal.

Now, let M be a maximal ideal of $k[X_1, \ldots, X_n]$, then $k[X_1, \ldots, X_n]/M$ is a finitely generated k-algebra which is a field. With the previous proposition, it is an algebraic finite extension of k, and since k is algebraically closed, $k[X_1, \ldots, X_n]/M = k$. So for $i \in \{1, \ldots, n\}$, there exists $a_i \in k$ such that $X_i - a_i \in M$. Then $\langle X_1 - a_1, \ldots, X_n - a_n \rangle \subset M$, and we have already seen that $\langle X_1 - a_1, \ldots, X_n - a_n \rangle$ is maximal. Hence the result is proven.

□

Theorem 3 (The Weak Nullstellensatz)**.** *Let k be an algebraically closed field and let $I \subset k[X_1, \ldots, X_n]$ be an ideal so as $V(I) = \varnothing$. Then $I = k[X_1, \ldots, X_n]$.*

Proof. With the fact that $k[X_1, \ldots, X_n]$ is noetherian, if $I \neq k[X_1, \ldots, X_n]$ then there exist a maximal ideal M of $k[X_1, \ldots, X_n]$ such that $I \subset M$. With the previous proposition, there exists $(a_1, \ldots, a_n) \in k^n$ such that $\langle X_1 - a_1, \ldots, X_n - a_n \rangle \subset M$. Yet, $I \subset M$ so $(a_1, \ldots, a_n) \in V(M) \subset V(I)$ and $V(I) \neq \varnothing$ which is absurd. So $I = k[X_1, \ldots, X_n]$. □

Remark. It is a little bit different but we can also show that if k is an infinite field (which is the case for an algebraically closed field), and if $P \in k[X_1, \ldots, X_n]$, with $P \neq 0$, then there exists $x_1, \ldots, x_n$ such that $P(x_1, \ldots, x_n) \neq 0$.

Proof. Let us show by induction on n that if $P \in k[X_1, \ldots, X_n]$, with $\forall x \in k^n, P(x) = 0$, then $P = 0$.

When $n = 1$, the result is well-known. We assume that $n \geq 2$ and that the result is proven for $n-1$. Let $P \in k[X_1, \ldots, X_n]$, with $\forall x \in k^n, P(x) = 0$. We can write $P = \sum_{i=1}^{N} A_i X_n^i$, with $A_i \in k[X_1, \ldots, X_{n-1}]$. If $(x_1, \ldots, x_{n-1}) \in k^{n-1}$, then we have for all $x_n \in k$, $P(x_1, \ldots, x_n) = 0 = \sum_{i=1}^{N} A_i(x_1, \ldots, x_{n-1}) x_n^i$. So with the case $n = 1$, the polynomial in X_n, $\sum_{i=1}^{N} A_i(x_1, \ldots, x_{n-1}) X_n^i$, equals the zero polynomial. Hence, If $(x_1, \ldots, x_{n-1}) \in k^{n-1}$, then $A_i(x_1, \ldots, x_{n-1}) = 0$. With the case $n - 1$, $A_i = 0$, and thus, $P = 0$. Finally, the result is proven. □

1.2.3 Strong form

Definition 6. Let I be an ideal of $k[X_1, \ldots, X_n]$. We define its **radical ideal** $\sqrt{I}$ by

$$\sqrt{I} = \{f \in k[X_1, \ldots, X_n] \,/\, \exists m \in \mathbb{N}^*, f^m \in I\}.$$

An ideal I such that $I = \sqrt{I}$ is called radical.

Theorem 4 (Hilbert's Nullstellensatz)**.** *Let k be an algebraically closed field. Let J be an ideal of $k[X_1, \ldots, X_n]$. Then*

$$\sqrt{J} = I(V(J)).$$

Proof. First, let $P \in \sqrt{J}$, then there exist $r \in \mathbb{N}^*$ such that $P^r \in J$. Then, if $x \in V(J)$, then $P^r(x) = 0$ and k is a field so $P(x) = 0$. Thus, $P \in I(V(J))$ and $\sqrt{J} \subset I(V(J))$.

Conversely, let $P \in I(V(J)$. In $k[X_1, \ldots, X_n, T]$, let J' be the ideal generated by J and $1 - TP$. If $V(J') \neq \varnothing$, let $z = (x_1, \ldots, x_n, t) = (x, t) \in V(J')$. Then $(1-PT)(z) = 0$ and also $x \in V(J)$, so $P(x) = 0$. Thus $(1-PT)(z) = 1-P(x)t = 1 \neq 0$ which is absurd. So $V(J') = \varnothing$.

With the previous theorem, $J' = k[X_1, \ldots, X_n, T]$, so there exist $m \in \mathbb{N}^*$, $Q_0, \ldots, Q_m$ in $k[X_1, \ldots, X_n, T]$, $(P_1, \ldots, P_j) \in J^m$ such that

$$(1 - TP)Q_0 + \sum_{j=1}^{m} Q_j P_j = 1.$$

In $k(X_1, \ldots, X_n)$, we can substitute T by $\frac{1}{P}$ and we find

$$\sum_{j=1}^{m} Q_j(X_1, \ldots, X_n, \frac{1}{P}) P_j(X_1, \ldots, X_n) = 1.$$

With r large enough (for instance, larger than the maximum over j of the maximum of the degree in T of the Q_j), we have the equality in $k[X_1, \ldots, X_n]$: $P^r = \sum_{j=1}^{m} P^r Q_j(X_1, \ldots, X_n, \frac{1}{P}) P_j(X_1, \ldots, X_n)$. Hence, $P^r \in J$, and the result is proven. □

With this theorem, it is easy to deduce a first correspondence between geometry (varieties) and algebra (ideals) :

Definition 7. A topological space is called irreducible if it cannot be expressed as the union of two proper closed subset. A nonempty subset of a topological space is called irreducible if it is an irreducible topological space for the induced topology. The empty set is not considered irreducible.

Corollary 1. *There is a one-to-one inclusion-reversing correspondence between affine varieties of k^n and radical ideals, given by $V \mapsto I(V)$ and $I \mapsto V(I)$. There is also a one-to-one correspondence between irreducible affine varieties and prime ideals.*

Remark. k^n is irreducible for the Zariski topology since $\langle 0 \rangle$ is prime ($k[X_1, \ldots, X_n]$ being indeed an integral domain).

1.3 Links between Algebra and Geometry

1.3.1 Zariski Topology

Let $n \in \mathbb{N}^*$. We can define a topology on k^n by taking the closed sets as the affine varieties (defined with ideals of $k[X_1, \ldots, X_n]$), which is called the **Zariski topology** on k^n.

Proposition 3. *The Zariski topology on k^n is well defined.*

Proof.

- $\varnothing = V(k[X_1, \ldots, X_n])$;
- $k^n = V(0)$;
- let I and J be two ideals of $k[X_1, \ldots, X_n]$. Then $V(I) \cup V(J) = V(IJ)$, and every finite union of affine varieties is an affine variety ;
- if $(I_\alpha)_{\alpha \in I}$ are some ideals of $k[X_1, \ldots, X_n]$, then $\bigcap_{\alpha \in I} V(I_\alpha) = V(+_{\alpha \in I} I_\alpha)$.

Thus, the Zariski topology is a a topology on k^n. □

Now, we can state what happens with the applications $V \mapsto I(V)$ and $I \mapsto V(I)$ when we do not deal with necessarily affine varieties :

Lemma 6. *If $A \subset k^n$, then $V(I(A)) = \overline{A}$.*

Proof. Let $f \in I(A)$ and $a \in A$. Then $f(a) = 0$ and thus $a \in V(I(A))$, and $V(I(A)) \supset A$. Since $V(I(A))$ is closed, $V(I(A)) \supset \overline{A}$. By definition, there exists some ideal J such that $\overline{A} = V(J)$. Then $V(I(A)) \supset V(J) \supset A$, thus $\sqrt{(I(A))} \subset \sqrt{(J)} \subset A$. Hence, $V(\sqrt{J}) \supset V(I(A))$ and $V(\sqrt{J}) = V(J) = \overline{A}$. Finally, $V(I(A)) = \overline{A}$.

□

1.3.2 Operations on Ideals and Varieties

Definition 8. If I and J are ideals of $k[X_1, \ldots, X_n]$, then we define $I : J$ with

$$I : J = \{f \in k[X_1, \ldots, X_n] \,/\, \forall g \in J, fg \in I\},$$

and $I : J$ is called the **ideal quotient**, or the colon ideal, of I by J.

Remark. If I and J are ideals of $k[X_1, \ldots, X_n]$, then $I : J$ is indeed an ideal of $k[X_1, \ldots, X_n]$.

Proposition 4. *If k is algebraically closed and I and J are ideals in $k[X_1, \ldots, X_n]$, then :*

- $V(I+J) = V(I) \cap V(J)$;
- $V(IJ) = V(I \cap J) = V(I) \cup V(J)$, *with* $IJ = \langle fg, (f,g) \in I \times J \rangle$;
- $\sqrt{I \cap J} = \sqrt{I} \cap \sqrt{J}$;
- $V(I:J) \supset \overline{V(I) - V(J)}$ *(with* $A - B = A \cap B^c$*), and if in addition,* I *is a radical ideal, then* $V(I:J) = \overline{V(I) - V(J)}$.

Proof. We wil only prove the last property. First, we show that $I : J \subset I(V(I) - V(J))$. If $V(I) \subset V(J)$, the result is obvious.

Let $f \in I : J$ and $x \in V(I) - V(J)$, then $fg \in I$ for all $g \in J$. Since $x \in V(I)$, we have $f(x)g(x) = 0$ for all $g \in J$. $x \notin V(J)$ so there exists $g \in V(J)$ such that $g(x) \neq 0$. Hence, with $f(x)g(x) = 0$, $f(x) = 0$, and then $I : J \subset I(V(I) - V(J))$. Hence, $V(I:J) \supset V(I(V(I) - V(J))) = \overline{V(I) - V(J)}$.

Then, with I a radical ideal, let $x \in V(I:J)$. Let $h \in I(V(I) - V(J))$. If $g \in J$, then g vanishes on $V(J)$, h on $V(I) - V(J)$ so hg vanishes on $V(I)$. Hence $hg \in \sqrt{I} = I$. So, if $h \in I(V(I) - V(J))$, if $g \in J$, $hg \in I$, and thus $h \in I : J$. Hence, $I(V(I) - V(J)) \subset I : J$. Thus, $I : J = I(V(I) - V(J))$, and then $V(I:J) = \overline{V(I) - V(J)}$. □

Now that the theoretical basis needed are recalled, we can wonder how computations can be made with ideals and such operations (radical, intersection, quotient,...), and hence come Gröbner bases.

2 Gröbner Bases

Before being able to define what a Gröbner Basis is, we shall present some preliminary results and definitions. Again in this section, some proofs will not be given, and we refer for them to *Cox, Little & O'Shea* [1].

2.1 Preliminary Works

2.1.1 Monomial Ordering

Definition 9. A **monomial ordering** on $\mathbb{N}^n$ ($n \in \mathbb{N}^*$) is an ordering $>$ satisfying :

- $>$ is a total ordering on $\mathbb{N}^n$;
- if $\alpha > \beta$ and $\gamma \in \mathbb{N}^n$, then $\alpha + \gamma > \beta + \gamma$;
- $>$ is a well-ordering on $\mathbb{N}^n$, which means that every nonempty subset of $\mathbb{N}^n$ has a smallest element for $>$.

Remark. $>$ is a *well-ordering* on $\mathbb{N}^n$ if and only if there is no strictly decreasing sequence (for $>$) in $\mathbb{N}^n$.

Remark. A monomial on $\mathbb{N}^n$ order clearly induce a total order on the monomials of $k[X_1, \ldots, X_n]$.

Definition 10 (Lexicographical order)**.** We define the **lexicographical order** $>_{lex}$ on $\mathbb{N}^n$ with : if $(\alpha, \beta) \in \mathbb{N}^n \times \mathbb{N}^n$, then $\alpha >_{lex} \beta$ if the left-most nonzero coordinate of $\alpha - \beta$ is positive.

Definition 11 (Graded Lex Order)**.** We define the **graded lexicographical order** $>_{grlex}$ on $\mathbb{N}^n$ with : if $(\alpha, \beta) \in \mathbb{N}^n \times \mathbb{N}^n$, then $\alpha >_{grlex} \beta$ if $|\alpha| > |\beta|$, or $|\alpha| = |\beta|$ and $\alpha >_{lex} \beta$.

Definition 12 (Graded reverse Lex Order)**.** We define the **graded reverse lexicographical order** $>_{grevlex}$ on $\mathbb{N}^n$ with : if $(\alpha, \beta) \in \mathbb{N}^n \times \mathbb{N}^n$, then $\alpha >_{grevlex} \beta$ if $|\alpha| > |\beta|$, or $|\alpha| = |\beta|$ and the right-most nonzero coordinate of $\alpha - \beta$ is negative.

Proposition 5. *All of the three orders above are monomial orders.*

Hence, we can define what is the multidegree of a polynomial, its leading coefficient, monomial, ...

Definition 13. Let $f = \sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $k[x_1, \ldots, x_n]$, and let $>$ be a monomial order. Then :

- The **multidegree** of f is $multidegree(f) = \max(\alpha \in \mathbb{N}^n / a_\alpha \neq 0)$ (maximum taken with respect to $>$).
- The **leading coefficient** of f is $LC(f) = a_{multideg(f)} \in k$.
- The **leading monomial** of f is $LM(f) = x^{multideg(f)}$.
- The **leading term** of f is $LT(f) = LC(f) \times LM(f)$.

2.1.2 Division Algorithm

We can generalize the euclidean division algorithm of $k[x]$ to $k[x_1, \ldots, x_n]$.

Theorem 5. *Being given a monomial order $>$ on $\mathbb{N}^n$ and $F = (f_1, \ldots, f_s)$ be an ordered s-tuple of polynomials in $k[X_1, \ldots, X_n]$, if $f \in k[X_1, \ldots, X_n]$, then f can be written $f = a_1 f_1 + \ldots a_s f_s + r$ where all a_i and r are in $k[X_1, \ldots, X_n]$, and either $r = 0$ or none of its monomials is divisible by any of the $LT(f_1), \ldots, LT(f_s)$. r is called a remainder of f on division by F (there is no unicity), and if $a_i f_i \neq 0$ then $multideg(f) \geq multideg(a_i f_i)$ for all i. The a_i can be computed by the algorithm given below.*

The idea of this algorithm is really the same than the one for $k[x]$: trying to eliminate the leading term (with respect to the order given) possible of f with the leading terms of F (F being ordered), and if it is not possible, this leading term goes to the remainder.

Algorithm 1 Division algorithm in $k[X_1, \ldots, X_n]$ of f by $(f_1, \ldots, f_s)$

```
r:=0; p:=f;
for i in [1..s] do
  a_i := 0;
end for
while p ≠ 0 do
  i:=1;
  divisionoccured:=false;
  while i ≤ s & divisionoccured == false do
    if LT(f_i) divides LT(p) then
      a_i := a_i + LT(p)/LT(f_i);
      p := p - (LT(p)/LT(f_i)) f_i;
      divisionoccured:=true;
    else
      i:=i+1
    end if
  end while
  if divisionoccured==false then
    r:=r+LT(p);
    p:=p-LT(p);
  end if
end while
return a_1, ..., a_s, r
```

2.1.3 Monomial Ideals and Dickson's lemma

Since in the division algorithm, it is mainly leading terms from F that are operating, we should consider the so-called monomials ideals :

Definition 14. An ideal $I \subset k[x_1, \ldots, x_n]$ is a **monomial ideal** if it can be generated by a set of monomials : thus if $I = \langle x^\alpha, \alpha \in A \rangle$ for some $A \subset \mathbb{N}^n$.

Lemma 7. *Let $I = \langle x^\alpha, \alpha \in A \rangle$ be a monomial ideal. Then x^β, for some $\beta \in \mathbb{N}^n$, lies in I if and only if x^β is divisible by x^α, for some $\alpha \in A$.*

Lemma 8. *Let I be a monomial ideal, and $f \in k[x_1, \ldots, x_n]$, then $f \in I$ if and only if every term of f lies in I, if and only f is a k-linear combination of the monomials in I.*

Lemma 9. *Two monomials ideals are the same if and only if they contains the same monomials.*

Then, the main result for monomial ideals is Dickson's lemma :

Theorem 6 (Dickson's Lemma)**.** *A monomial ideal $I = \langle x^\alpha, \alpha \in A \rangle \subset k[x_1, \ldots, x_n]$ can be written in the form $I = \langle x^{\alpha_1}, \ldots, x^{\alpha_s} \rangle$, for some $\alpha_i \in A$.*

Proof. We proceed by induction on the number of variables, n.

If $n = 1$, then I is genererated by the x^α, $\alpha \in A \subset \mathbb{N}$. If β is the smallest element of A, then x^β divides all the x^α with $\alpha \in A$, and $I = \left\langle x^\beta \right\rangle$

Now, we assume that the theorem is true for $n-1$, for some $n \in \mathbb{N}$, $n \geq 2$. Suppose that I is a monomial ideal of $k[x_1, \ldots, x_n, y]$. We will write the monomials of $k[x_1, \ldots, x_n, y]$ as $x^\alpha y^m$ with $\alpha \in \mathbb{N}^{n-1}$ and $m \in \mathbb{N}$.

Let J be the ideals in $k[x_1, \ldots, x_n]$ defined by the monomials x^α for which there exists $m \in \mathbb{N}$ such that $x^\alpha y^m \in I$. J is a monomial ideal and by the inductive hypothesis, we can write $J = \left\langle x^{\alpha(1)}, \ldots, x^{\alpha(s)} \right\rangle$. For $i \in \{1, \ldots, s\}$, there exists $m_i \in \mathbb{N}$ such that $x^{\alpha(i)} y^{m_i} \in I$. Let m be the maximum of the m_i.

Now, for each $k \in \{0, \ldots, m-1\}$, we consider J_k the ideal in $k[x_1, \ldots, x_n]$ defined by the monomials x^β such that $x^\beta y^k \in I$.

Again, $J_k = \left\langle x^{\alpha_k(1)}, \ldots, x^{\alpha_k(s_k)} \right\rangle$.

Finally, let us show that I is generated by the $x^{\alpha(1)} y^m, \ldots, x^{\alpha(s)} y^m$ (coming from J), and all the $x^{\alpha_k(1)} y^k, \ldots, x^{\alpha_k(s_k)} y^k$ coming from J_k, for each $k \in \{0, \ldots, m-1\}$. Of course, by definition, all those monomials are in I.

If $x^\alpha y^p$ is a monomial of I, then :

- if $p \geq m$, then by definition of J, $x^\alpha y^p$ is divisible by some $x^{\alpha(i)} y^m$.
- elsewhere, $p \leq m-1$, then $x^\alpha y^p$ is divisible by some $x^{\alpha_p(i)} y^p$.

Hence, those monomials generate an ideals having the same monomials as I, and thus by a previous lemma, I is generated by those monomials.

Finally, we shall prove that those generators can be chosen in A. If $I = \langle x^\alpha, \alpha \in A \rangle$, we have seen that we can write $I = \left\langle x^{\beta_1}, \ldots, x^{\beta_p} \right\rangle$. For $i \in \{1, \ldots, p\}$, since $x^{\beta_i} \in I$, x^{β_i} can be divided by some x^{α_i}, $\alpha_i \in A$. Hence, $I = \langle x^{\alpha_1}, \ldots, x^{\alpha_p} \rangle$. □

But what is more impressive is that this property somehow still holds for any ideal, and here will lie the idea of Gröbner bases.

2.2 Definition and Properties

2.2.1 Definition

Definition 15. Let $I \subset k[x_1, \ldots, x_n]$ be an ideal, different from $\{0\}$. Given a monomial order, we define $LT(I) = \{LT(f) / f \in I\rangle$. We will naturally note $\langle LT(I)\rangle$ the ideal generated by $LT(I)$.

Lemma 10. *$\langle LT(I)\rangle$ is a monomial ideal, and thus, there exists $g_1, \ldots, g_t$ in I such that $\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_t)\rangle$.*

Definition 16. Given a monomial order, a finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal I is called a **Gröbner Basis** if $\langle LT(g_1), \ldots, LT(g_t)\rangle = \langle LT(I)\rangle$.

Proposition 6. *Given a monomial order, every ideal $I \subset k[x_1, \ldots, x_n]$ has a Gröbner basis, and any Gröbner basis of I generates I.*

Proof. The previous lemma show the first part of the proposition. For the second part, if $g_1, \ldots, g_t$ is a Gröbner basis of an ideal I, then, if $f \in I$, the division of f by $g_1, \ldots, g_t$, with the division algorithm we have seen previously, gives us $f = a_1 g_1 + \cdots + a_t g_t + r$, with no term of r divisible by one of the $LT(g_i)$.

Hence, $r = 0$ because by construction, $r \in I$, so

$$LT(r) \in \langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_t)\rangle$$

and thus, r should be divisible by some $LT(g_i)$, which is absurd. Finally, $f \in \langle g_1, \ldots, g_t\rangle$, and the result is proven. □

2.2.2 Division and Gröbner Bases

Proposition 7. *Let $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis of an ideal $I \subset k[x_1, \ldots, x_n]$, and let $f \in k[x_1, \ldots, x_n]$, then there is a unique $r \in k[x_1, \ldots, x_n]$ such that no term of r is divisible by any of the $LT(g_1), \ldots, LT(g_t)$, and there is $g \in I$ with $f = g + r$.*

In particular, r is the remainder of the division of f by G, no matter of how the elements of G are listed for the division algorithm, and $f \in I$ if and only if the remainder of the division of f by G is 0.

Proof. With the division algorithm, the existence of such a f is provided.

Considering uniqueness, if $f = g^{(1)} + r_1 = g^{(2)} + r_2$ with $r_1 \neq r_2$, then $r_1 - r_2 = g^{(2)} - g^{(1)} \in I$, and thus, $LT(r_1 - r_2)$ is divisible by some $LT(g_i)$, which is impossible since no term of r_1 nor r_2 is divisible by any of the $g_1, \ldots, g_t$. Thus, $r_2 = r_1$ and the uniqueness is proven. □

Definition 17. Let $f, g \in k[x_1, \ldots, x_n]$ be two nonzero polynomials.

- If $multideg(f) = \alpha$ and $multideg(g) = \beta$, let $\gamma = (\gamma_1, \ldots, \gamma_n)$, where $\gamma_i = \max(\alpha_i, \beta_i)$ for each i. Then x^γ is called the *least common multiple* of $LM(f)$ and $LM(g)$: $x^\gamma = LCM(LM(f), LM(g))$.
- The **S-polynomial** (from syzygy polynomial) of f and g is

$$S(f,g) = \frac{x^\gamma}{LT(f)} f - \frac{x^\gamma}{LT(g)} g$$

Theorem 7. *Let I be an ideal of $k[x_1, \ldots, x_n]$, then a basis $G = \{g_1, \ldots, g_t\}$ for I is a Gröbner basis for I if and only if for all i, j in $\{1, \ldots, t\}$, the remainder of the division of $S(g_i, g_j)$ by G is zero.*

2.3 How to Compute a Gröbner Bases

2.3.1 Buchberger's Algorithm

The previous theorem gives us an algorithm to compute a Gröbner basis of an ideal, given a basis of that ideal : this is the Buchberger's algorithm.

Algorithm 2 Buchberger's algorithm : computes a Gröbner basis of the ideal $I = \langle F \rangle$, $F = (f_1, \ldots, f_s)$

```
G := F;
repeat
   G := G';
   for (p, q) ∈ G'^2, p ≠ q do
      S := overline(S(p, q))^{G'}
      if S ≠ 0 then
         G := G ∩ {S}
      end if
   end for
until G == G'
return G
```

2.3.2 Minimal and Reduced Gröbner Bases

The results given by Buchberger's algorithm can be very big in size, and some of its elements can be somehow "useless". What follow will explain this idea.

Lemma 11. *Let G be a Gröbner basis for the polynomial ideal I, and let $p \in G$ be a polynomial such that $LT(p) \in \langle LT(G \setminus \{p\}) \rangle$. Then $G \setminus \{p\}$ is also a Gröbner basis for I.*

Definition 18. A **minimal Gröbner basis** for a polynomial ideal I is a Gröbner basis G for I such that :

- $\forall p \in G,\ LC(p) = 1$
- $\forall p \in G,\ LT(p) \notin \langle LT(G \setminus \{p\})\rangle$

Definition 19. A **reduced Gröbner basis** for a polynomial ideal I is a Gröbner basis G for I such that :

- $\forall p \in G,\ LC(p) = 1$
- For all $p \in G$, no monomials of p lies in $\langle LT(G \setminus \{p\})\rangle$

Proposition 8. *Let $I \neq \{0\}$ be a polynomial ideal. Then, for a given monomial ordering, I has a unique reduced Gröbner basis.*

Yet, we have chosen not to look for an algorithm that gives minimal or reduced, and instead, we have prefered jumping directly to the applications of Gröbner basis to the ideal computation, and then, computational invariant theory. This choice will reveal mostly harmless. Still, some of the more advanced algorithm might suffer a little from the use of non-reduced and non-minimal Gröbner basis in their implementation.

3 Ideal Computation with Gröbner Bases

Gröbner basis are the tool "par excellence" of any computation involving ideals. In this section, we will particularly see how Gröbner basis can be used to compute operations on ideals : intersection, ideal quotient,...

3.1 Elimination Ideal

The first operation that we will need to compute is that of elimination ideal, and to begin with, we shall of course define what an elimination ideal is :

Definition 20. Let $I \subset k[X_1, \ldots, X_n]$ be an ideal, and $U \subset \{X_1, \ldots, X_n\}$. Then the ideal $I \cap k[U]$ in $k[U]$ is called an elimination ideal. If $U = \{X_i, \ldots, X_n\}$, then the ideal $I \cap k[U]$ in $k[U]$ is called the i-th elimination ideal.

Elimination ideals, and the fact that they can be very easily computed, will prove very useful for more complicated and avanced computation. The proposition that follows explain how they can indeed be computed.

Proposition 9. *Let $I \subset k[X_1, \ldots, X_n]$ be an ideal, and $U \subset \{X_1, \ldots, X_n\}$. Let G be a Gröbner basis of I with respect to a monomial order such that for all $s \in k[U]$, $t \in k[U^c]$ non constant, $u \in k[U]$, $s < ut$. Then $G \cap K[U]$ is a Gröbner basis of the elimination ideal $I \cap k[U]$, with respect to the order induced on $k[U]$.*

Proof. The first thing that we can notice is that with such a monomial ordering, and with G a Gröbner basis with respect to that order, if we take $P \in G \cap k[U]^c$, then necessarily, its leading monomial would have nonzero exponents in U^c since if no term of P has nonzero exponents in U^c, then $P \in k[U]$, and with the fact that $\forall s \in k[U], \forall t \in k[U^c], \forall u \in k[U]$, t non constant, $s < ut$, then any such term is greater than any term in $k[U]$, and thus the leading monomial of P can not lie in $k[U]$.

Then, if $x \in I \cap k[U]$, and if we perform the division of x by G, then the leading monomial of the elements of $G \cap k[U]^c$ can not divide any monomial of $x \in k[U]$ and thus, they do not take part in the division. Hence, dividing $x \in k[U]$ by G is exactly the same thing than dividing x by $G \cap k[U]$.

It then came naturally that $G \cap k[U]$ generates $I \cap k[U]$ since G generates I.

In the same way, if we take $S(f, g)$ with $f, g \in G \cap k[U]$, then $S(f, g) \in k[U]$, and the division of $S(f, g)$ by $G \cap k[U]$ is the same than the division by G. Hence the remainder is 0, and we have directly that $G \cap k[U]$ is a Gröbner basis of $I \cap k[U]$. □

Remark. If $U = \{k, \ldots, n\}$ for some $k \in \{1, \ldots, n\}$, then we can use the lexicographical order. And if $U = \{1, \ldots, k\}$, we can also use the lexigraphical order, with the order in the variables taken backward. For the algorithms below, we will only need those two cases.

3.2 Intersection of Ideals

With the ability of computing elimination ideals, we now can look at how intersection of ideals can be computed.

3.2.1 A formula

The idea of the computation we give lies in the following propostion :

Proposition 10. *Let I and J be two ideals of $k[X_1, \ldots, X_n]$.*

Let $L = t \times I + (1 - t) \times J$ in $k[X_1, \ldots, X_n, t]$ (thus, we add a new variable, t), with the products formed by multiplying generators by t or $1 - t$, respectively. Then $I \cap J = L \cap k[X_1, \ldots, X_n]$.

Proof. If $x \in I \cap J$, then $x = tx + (1-t)x$ with $x \in I$ and $x \in J$, so $x \in L$, and by definition $x \in k[X_1, \ldots, X_n]$, thus $x \in L \cap k[X_1, \ldots, X_n]$. Conversely, if $x \in L \cap k[X_1, \ldots, X_n]$, then $x \in L$, thus $x = t*i+(1-t)*j$ for some $i \in I$, $j \in J$. Yet, $x \in k[X_1, \ldots, X_n]$, so we can evaluate $x = t*i+(1-t)*j$ in $t = 0$ and $t = 1$ to find $x = i = j$ and $x \in I \cap J$. Thus the result is proven. □

Remark. We may notice that $L \cap k[X_1, \ldots, X_n]$ is an elimination ideal.

If theoritically, the formula seems easy to implemant, we still have to *add a new variable*, and not all computer algebra system handle that easily... In what follows, we will explain how it can be done in Magma.

3.2.2 How to add a new variable, coercion in Magma

Magma can easily coerce polynomial from a polynomial ring to another, as soon as those two polynomial rings have the same rank ... Hence, no automatic coercion is available between, say $k[X_1, \ldots, X_n]$ and $k[Y_1, \ldots, Y_n, t]$, or even with the use of so-called "global polynomial ring" of different rank...

That is why, to implement successfully the intersection of ideals, we have to implement our own coercion functions. For that, hopefully, we can use Magma's homomorphisms :

```
function expand(R,l)
n:=Rank(R);
K:=BaseRing(R);
R2:=PolynomialRing(K,n+l);
Z:=[R2.i : i in [1..n]];
f:=hom<R->R2 | Z>;
return f,R2;
end function;

function proj(R,l)
n:=Rank(R);
K:=BaseRing(R);
R2:=PolynomialRing(K,l);
Z:=[R2.i : i in [1..l]] cat [0 : i in [(l+1)..n]];
f:=hom<R->R2 | Z>;
return f,R2;
end function;
```

The first function, given a polynomial ring $R = k[X_1, \ldots, X_n]$, and a number l of variables to add, returns the canonical injection : $f : k[X_1, \ldots, X_n] \to k[Y_1, \ldots, Y_{n+l}]$, $P \mapsto P$, and a polynomial ring $R2 = k[Y_1, \ldots, Y_{n+l}]$.

The second function, given a polynomial ring $R = k[X_1, \ldots, X_n]$, and a number $l \leq n$ of variables as a goal, returns the canonical surjection : $fk[X_1, \ldots, X_n] \to k[Y_1, \ldots, Y_l]$, $X_i \mapsto Y_i$ if $i \in \{1, \ldots, l\}$ and $X_i \mapsto 0$ elsewhere, and a polynomial ring $R2 = k[Y_1, \ldots, Y_l]$.

Finally, if we have a polynomial ring R of rank n, if $t \in \mathbb{N}^*$, expand will give us R_1 of rank $n + t$ and $f : R \to R_1$, canonical injection. Then, proj(R_1,n) will give us R_2 of rank n and $g : R_1 \to R_2$ canonical surjection. Since R_2 is of rank n, we can use automatic coerction from R_2 to R. It will be performed by the command $R!x$ which will coerce x in R, if possible.

Hence, with f, g, and automatic coercion, we can indeed add t variables to $R = k[X_1, \ldots, X_n]$ when working with $R_1 = k[Y_1, \ldots, Y_{n+t}]$.

3.2.3 An Algorithm in Magma

So, with those two functions, we can now write an algorithm in Magma to compute the intersection of two ideals :

```
function intersection(F1,F2,R)
n:=Rank(R);
n1:=#F1;
n2:=#F2;
f,R2:=expand(R,1);
F3:=[f(F1[i])*R2.(n+1) : i in [1..n1]]
cat [f(F2[i])*(1-R2.(n+1)) : i in [1..n2]];
G:=grobner(F3,R2,lexord);
G2:=eliminationset(G,{n+1},R2);
g,R3:=proj(R2,n);
s:=#G2;
G3:=[R!(g(G2[i])) : i in [1..s]];
return G3;
end function;
```

3.3 Ideal Quotient and Saturation

3.3.1 Ideal Quotient

With the previous algorithm to compute intersection of ideals, and with the two properties given below, we can easily deduce an algorithm to compute quotients of ideals.

Proposition 11. *Let I, J be two ideals of $k[X_1, \ldots, X_n]$, and let us write $J = \langle f_1, \ldots, f_k \rangle$, and let $f \in k[X_1, \ldots, X_n]$. Then we have :*

- $I : J = \bigcap_{i=1}^{k} I : f_i$
- $I : f = (I \cap (f))f^{-1}$

The $I : f = (I \cap (f))f^{-1}$ means the ideal of the elements of $I \cap (f)$, divided by f. It is then rather straightforward to prove the proposition by just recalling the definitions... The algorithm follows directly the proposition.

3.3.2 Saturation

Another interesting operation on ideal is that of saturation. Here is the definition of the saturation ideal of I with respect to f :

Definition 21. $I : f^{\infty} = \bigcup_{i \in \mathbb{N}} I : f^i$ is the saturation ideal of I with respect to f.

Since the $I : f^i$ defines an ascending ideal chain, and with the fact that $k[X_1, \ldots, X_n]$ is noetherian, we can find $k \in \mathbb{N}$ such that $I : f^{\infty} = I : f^k$

The proposition below allows us to find a faster algorithm to compute saturation ideals than finding such a k.

Proposition 12. *Let I be an ideal of $k[X_1, \ldots, X_n]$, and let $f \in k[X_1, \ldots, X_n]$. We define $J =< I, tf - 1 >$ an ideal of $k[X_1, \ldots, X_n, t]$. Then $I : f^{\infty} = J \cap k[X_1, \ldots, X_n]$.*

Proof. Let $g \in J \cap k[X_1, \ldots, X_n]$. Then $g = q_1 i + q_2(1 - tf)$, with $q_1, q_2 \in k[X_1, \ldots, X_n, t]$. In $k(X, Y)$, we can substitute Y by $\frac{1}{f}$ and then multiply the equation by f^d where $d = \deg_t q_1$ to obtain $f^d g = qi$, where $q \in k[X_1, \ldots, X_n]$, thus $g \in I : f^{\infty}$. Conversely, let $g \in I : f^{\infty}$. Then there exists $d \in \mathbb{N}$ such that $gf^d \in I$. By definition of J, $1 = tf + j$ with $j \in J$. Thus, $1 = (tf)^d + j'$ with $j' \in J$. Thus, $g = gf^d t^d + gj' \in J$, and we have proved the result. □

Again, the algorithm is just the application of the formula.

3.4 The Dimension of an Ideal

In this section, we will explain how the dimension of an ideal can be computed. First, we will need some definitions.

3.4.1 Definitions and first properties

Different definitions can be given to the dimension of an ideal. We will see that they somehow all corresponds.

Definition 22.
- The Krull dimension of a ring is the supremum of the length n of strictly ascending chain $I_0 \subsetneq I_1 \cdots \subsetneq I_n$ of prime ideals of the ring.

- The Krull dimension of an ideal $I \subset k[X_1, \ldots, X_n]$ is the Krull dimension of the quotient ring $k[X_1, \ldots, X_n]/I$.
- If X is a topological space, we define the dimension of X as the supremum of the integers n such that there exists a strictly increasing chain $Z_0 \subsetneq \cdots \subsetneq Z_n$ of irreducible closed subset of X. We define the dimension of an affine variety as its dimension as a topological space.

Proposition 13. *If Y is an affine variety, then its dimension is equal to the dimension of $I(Y)$.*

Proof. The correspondance between prime ideals containing $I(Y)$ and closed irreducible subsets of Y shows us the correspondance between chains of prime ideals of $k[X_1, \ldots, X_n]/I$ and chains of closed irreducible subsets of Y, and hence, the equality of the dimensions. □

Definition 23.
- If $k \subset L$ are two fields, and if $S \subset L$, then S is called **algebraically independant** over k if for all $\{a_1, \ldots, a_n\} \subset S$ (no two the same), if $P \in k[X_1, \ldots, X_n]$ is such that $P(a_1, \ldots, a_n) = 0$, then $P = 0$.
- The **transcendence degree** of L over k is the cardinal of the largest algebraically independant subset of L over k.
- The **transcendence degree** of L a k-algebra which is an integral domain is the transcendance degre of its fraction field.
- L is called a **purely transcendental** extension of k if there exists an algebraically independant over k $S \subset L$ such that $L = k(S)$

We will need to admit the following theorem (related to Noether's Normalisation lemma). Its proof can be found in any good book about algebraic geometry.

Theorem 8. *If I is a prime ideal of $k[X_1, \ldots, X_n]$, then the dimension of I is equal to the transcendence degree of $k[X_1, \ldots, X_n]/I$ over k.*

We shall also admit the link between the dimension of an ideal and that of its monomial ideal.

Definition 24. A monomial ordering is called graded if $deg(f) > deg(g)$ implies that $f > g$, with deg the total degree.

Proposition 14. *If I is an ideal of $k[X_1, \ldots, X_n]$, and if we take a graded monomial ordering, then $dim I = dim LT(I)$.*

3.4.2 Heuristics

If k is an algebraically closed field, if we take G a Gröbner basis of an ideal I, with respect to a graded monomial ordering, then we can write $I = \langle g_1, \ldots, g_s \rangle$, and $LT(I) = \left\langle X_1^{\alpha_{1,1}} \ldots X_n^{\alpha_{1,n}}, \ldots, X_1^{\alpha_{s,1}} \ldots X_n^{\alpha_{s,n}} \right\rangle$ with $LM(g_j) = X_1^{\alpha_{j,1}} \ldots X_n^{\alpha_{j,n}}$. Thus, with basic properties of operations on varieties,

$$\begin{aligned} V(LT(I)) &= \bigcap_{i \in \{1,\ldots,s\}} V\left(X_1^{\alpha_{i,1}} \ldots X_n^{\alpha_{i,n}}\right) \\ &= \bigcap_{i \in \{1,\ldots,s\}} \bigcup_{j \in \{1,\ldots,n\}} V(X_j^{\alpha_{i,j}}) = \bigcap_{i \in \{1,\ldots,s\}} \bigcup_{j \in \{1,\ldots,n\}, \alpha_{i,j} \neq 0} V(X_j) \\ &= \bigcup_{J \in \{1,\ldots,n\}^s} \bigcap_{i \in \{1,\ldots,s\}, \alpha_{i,J_i} \neq 0} V(X_j) = \bigcup_{M \in L} V(M), \end{aligned}$$

where $L \in P(\{X_1, \ldots, X_n\})$, using the distributivity of $\cap$ with $\cup$.

We shall admit the following proposition :

Proposition 15. *If V and W are affine varieties, then $dim\,(V \cup W) = \max\,(dimV, dimW)$.*

Besides that, if $M \subset \{X_1, \ldots, X_n\}$, then $\langle M \rangle$ is prime ($k\,[\{X_1, \ldots, X_n\} \setminus M]$ being an integral domain) and

$$tr\deg k[X_1, \ldots, X_n] / \langle M \rangle = n - \sharp M.$$

Hence, $dimV(M) = n - \sharp M$.

Thus, if $V(LT(I)) = \bigcup_{M \in L} V(M)$, then $dimI = \max_{M \in L} \{n - \sharp M\}$.

We can notice that $V(M) \subset V(LT(I))$ if and only if every generator of $LT(I)$ involves at least one variable X_i lying in M. Thus all the M can be combinatorialy computed, and thus the dimension.

We can also notice that $M' = \{X_1, \ldots, X_n\} \setminus M$ is such that $V(M) \subset V(LT(I))$ if and only if any $LM(g_j)$ involves at least one variable not in M', and then the dimension of I would be the maximum of the such $\sharp M'$.

This can be rephrase as $M' = \{X_1, \ldots, X_n\} \setminus M$ is such that $V(M) \subset V(LT(I))$ if and only if for all j, $LM(g_j) \notin k[M']$.

3.4.3 An algorithm

This is how the algorithm we give for the computation of the dimension of an ideal works : it reduces to the monomial ideal and computes all the possible M' recursively starting with $\emptyset$, and finally it gives the largest one and its cardinal. Those algorithms are purely combinatorial, and we will only refer to Becker and Weispfenning [6] (pages 449 to 451) for a proof.

Algorithm 3 Computation of the dimension of the ideal generated by F in R.

```
G:=grobner(F,R,grevlexord);
s:=♯G;
for i in [1..s] do
  l:=normemultidegree(G[i],R,grevlexord);
  if l eq 0 then
    return -1; break i;
  else if l < 0 then
    Remove( G,i);
  end if
end for
return  dim(G,R,grevlexord);
```

Algorithm 4 dim function

```
M:=dimrec(LTens(G,R,ord),1,{},{},R);
d,U:=maxcardens(M);
return  M,d,U;
```

3.5 The Radical of an Ideal

In this section, we will present how we can compute the radical of an ideal in any characteristic. We will first treat the case of the zero characteristic, and then that of the positive characteristic. Since those questions are not entirely easy, we will mostly only give few explanations about the algorithms, leaving the reader to look at the references for more information.

3.5.1 Zero-dimensional Radical in Zero Characteristic

The algorithm we present for computing radical in zero characteristic relies first on the computation of radical of zero-dimensional ideals, which is simpler, and enough to deduce the computation for higher-dimensional ideals.

The main idea for the computation comes from the following proposition :

Proposition 16. *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal ($n \geq 1$). If $I \cap k[x_i]$ contains a separable polynomial for each $i \in \{1, \ldots, \}$, then $I = \sqrt{I}$.*

For a proof, we refer to Becker and Weispfenning [6] (lemmea 8.13).

Then, the following algorithm, due to Faugère, allows us to compute univariate polynomials in a zero-dimensional ideal (here for the variable i) :

Algorithm 5 function dimrec(S,k,U,M,R)

M2:=M; n:=Rank(R);
for i in [k..n] **do**
  if $S \cap k\,[\{X_1, \ldots, X_n\} \setminus (U \cup \{X_i\})]$ **then**
    M2:=dimrec(S,i+1,U join i,M2,R);
  end if
end for
if No element of M2 already contains U **then**
  M2:=M2 join U;
end if
return M2;

Algorithm 6 function univariate(F,R,i)

G:=grobner(F,R,grevlexord); n:=Rank(R);
d:=0; t,L_d:=division($R.i^d$,G,R,grevlexord); {L is the remainder of the division}
while $L_0, \ldots, L_d$ is linearly independant **do**
  d:=d+1;
  t,L_d:=division($R.i^d$,G,R,grevlexord);
end while
return $\sum_{j=0}^{d} \alpha_j x_i^j$ where $\sum_{j=0}^{d} \alpha_j L_i = 0$;

It is easy to see that if $\sum_{j=0}^{d} \alpha_j L_i = 0$, then $\sum_{j=0}^{d} \alpha_j x_i^j \in I$, with the "linearity" of the remainder of the division by G.

With the proposition and the algorithm, we can deduce an algorithm to compute the radical of a zero-dimensional ideal in caracteristic zero.

Algorithm 7 function zerodimradical(F,R)

n:=Rank(R);
for i in $\{1, \ldots, n\}$ **do**
  f_i:=univariate(F,R,i);
  $g_i := \frac{f_i}{GCD(f_i, f_i')}$; {with the derivative with respect to x_i}
end for
return F cat $[g_1, \ldots, g_n]$

In zero characteristic, any of the g_i is separable, and is of course in $\sqrt{I}$, thus with the proposition, it is easy to see that $I + \langle g_1, \ldots, g_n \rangle = \sqrt{I}$, and it corresponds to the F cat $[g_1, \ldots, g_n]$ returned.

3.5.2 Higher-dimensional Radical in Zero Characteristic

Before we can present the algorithm for any dimension radical, we shall present two proposition to explain how the algorithm works.

Proposition 17. *Let $L = k(x_{r+1}, \ldots, x_n)$ be a rational function field and J be an ideal of $L[x_1, \ldots, x_r]$. If G is a Gröbner basis of J with respect to any monomial ordering, such that $G \subset k[x_1, \ldots, x_n]$, which can easily be achieved for any Gröbner basis by multiplying by the LCM of the denominators of the coefficients. Let $f := LCM\{LC(g), g \in G\}$, with the LCM taken in $k[x_{r+1}, \ldots, x_n]$. Let I be the ideal in $k[x_1, \ldots, x_n]$ generated by G, then $J \cap k[x_1, \ldots, x_n] = I : f^\infty$.*

Proposition 18. *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal. Let G be a Gröbner basis with respect to the lexicographical ordering. Let $f = LCM\{LC(\overline{g}), g \in G\}$ where $\overline{g}$ is considering g as an element of $k(x_{r+1}, \ldots, x_n)[x_1, \ldots, x_r]$, and taking the leading coefficient with respect to the lexicographical ordering. Then the contraction ideal of $J = Ik(x_{r+1}, \ldots, x_n)[x_1, \ldots, x_r]$ is $J \cap k[x_1, \ldots, x_n] = I : f^\infty$.*

Furthermore, $J = Ik(x_{r+1}, \ldots, x_n)[x_1, \ldots, x_r]$ is zero-dimensional, with a k such that $I : f^\infty = I : f^k$, then $I = (I + (f^k)) \cap (I : f^\infty)$.

For a proof, we refer again to Becker and Weispfenning [6] (lemma 8.91, 8.94 and 8.95).

With those two propositions, we can provide an algorithm to compute the radical ideal of an ideal in zero characteristic :

Algorithm 8 function radical0(F,R)

n:=Rank(R);
M,d,U:=dimension(F,R);
if d = -1 **then**
  return F; $\{\langle F\rangle = k[x_1, \ldots, x_n]\}$
else
  if d=0 **then**
    return zerodimradical(F,R);
  else
    Renumber the variables such that $M = \{1, \ldots, r\}$
    $L := k(x_{r+1}, \ldots, x_n)$;
    with the previous proposition, find $f \in k[x_{r+1}, \ldots, x_n]$ such that $I = (I + (f^k)) \cap (IL[x_1, \ldots, x_r] \cap k[x_1, \ldots, x_n]$ for some k
    J:=zerodimradical($IL[x_1, \ldots, x_r]$,$L[x_1, \ldots, x_r]$);
    with the previous proposition, compute $J^c = J \cap k[x_1, \ldots, x_n]$
    I2:=radical0(I cat[f],R);
    return $I2 \cap J^c$;
  end if
end if

3.5.3 Radical in positive characteristic

Thanks to Matsumoto [8], there is a simpler algorithm for computing the radical of a polynomial ideal over a perfect positive-characteristic field. Yet, we will only refer to the article for proofs and explanations.

Algorithm 9 function radicalp(F,q,R), where q is a power of p, the characteristic of the base ring of R

n:=Rank(R);
k:=BaseRing(R);
B:=grobner(F,R,grevlexord);
trouve:=false;
repeat
$B_0 := \{\phi(B_i), i \in \{1, \ldots, \sharp B\}\}$; {where $\phi : \sum_\alpha a_\alpha x^\alpha \mapsto \sum_\alpha a_\alpha^{1/q} x^\alpha$ which is well defined (perfect field)}
B':=grobner(B_0 cat $\{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$,$k\,[X_1, \ldots, X_n, Y_1, \ldots, Y_n]$,lexord);
B_0':=elimination(B',n+1, $k\,[X_1, \ldots, X_n, Y_1, \ldots, Y_n]$); $\{B' \cap k[Y_1, \ldots, Y_n]\}$
B":=B_0' with substitution of the Y_i by the X_i;
if $\langle B \rangle = \langle B'' \rangle$ **then**
trouve:=true
else
B:=B";
end if
until trouve

3.6 Computation Results and Examples

Here, we will discuss a few exemple of use of our implementation of the algorithms previously given.

3.6.1 Basic operations

First, we will see that $\langle X^2Y \rangle \cap \langle XY^2 \rangle = \langle X^2Y^2 \rangle$:

```
> R:=PolynomialRing(RationalField(),2);
> intersection([X^2*Y],[X*Y^2],R);
[
X^2*Y^2
]
```

Then, we can compute $\langle XY(Z+2)^2, (X+3)Y^2 \rangle : \langle XYZ, X(Y-1)^2Z \rangle$:

```
> R:=PolynomialRing(RationalField(),3);
> colon([X*Y*(Z+2)^2,(X+3)*Y^2],[X*Y*Z,X*(Y-1)^2*Z],R);
[
X*Y*Z^3 + 8*X*Y*Z^2 + 20*X*Y*Z + 16*X*Y,
Y*Z^3 + 4*Y*Z^2 + 4*Y*Z,
```

```
-4*X*Y*Z^2 - 16*X*Y*Z - 16*X*Y,
Y*Z^2 + 4*Y*Z + 4*Y,
12*X*Y^2*Z + 16*X*Y^2 + 36*Y^2*Z + 48*Y^2,
4/3*X*Y^2 + 4*Y^2
]
```

We can also compute $\langle X^2 + Z, Y^2 + Z\rangle : (X - Y)^\infty$:

```
> R:=PolynomialRing(RationalField(),3);
> saturation([X^2+Z,Y^2+Z],X-Y,R);
[
X^2 + Z,
Y^2 + Z,
X + Y
]
```

3.6.2 About Dimension Computation

Now we will take a quick look at what our implementation for the computation of the dimension of an ideal is able to do :

```
> R:=PolynomialRing(RationalField(),3);
> F:=[Y^2*Z^3,X^5*Z^4,X^2*Y*Z^3];
> G:=grobner(F,R,grevlexord);
> dimension(G,R);
{
{ 3 },
{ 1, 2 }
}
2 { 1, 2 }
```

The answers given are, in that order :

- The list of all maximal independant subset of $\{X_1, \ldots, X_n\}$;
- The dimension of the ideal ;
- A maximal independant subset of maximal cardinal.

Yet, if the polynomials generating the ideal grow in degree (say, total degree), we will see how our implementation is limited :

```
> dimension(grobner([(X-1)^2*Y*Z,X*(Y+2)^2*(Z+3),
X*(Y+5)*(Z+2)],R,grevlexord),R);
{
{ 2 },
{ 3 }
}
1 { 3 }
> dimension(grobner([(X-1)^2*Y*Z,X*(Y+2)^2*(Z+3),
(X+3)*(Y+5)*(Z+2)],R,grevlexord),R);
{
{}
}
0 {}

> dimension(grobner([(X-1)^2*Y^3*Z^4,X*(Y+2)^2*(Z+3)^3,
(X+3)^2*(Y+5)^3*(Z+2)],R,grevlexord),R);
```

And the last one would not give any answer in one hour...

3.6.3 About Radical Computation

Here is an exemple of the computation of the radical of a radical ideal. We see that indeed, even if we get a more complicated set of generators as an answer (due to Gröbner basis computation), we indeed get the same ideal :

```
> R:=PolynomialRing(RationalField(),3);
> I:=ideal;
> F0:=[x*y*z,(x+1)*(y+1)*(z+1),(x+2)*(y+2)*(z+2)];
> radical(F0,R);
[
x*y*z,
x*y*z + x*y + x*z + x + y*z + y + z + 1,
x*y*z + 2*x*y + 2*x*z + 4*x + 2*y*z + 4*y + 4*z + 8,
-x*y - x*z - x - y*z - y - z - 1,
-2*x - 2*y - 2*z - 6,
z^3 + 3*z^2 + 2*z,
-y^2*z - y*z^2 - 3*y*z,
-y^2 - y*z - 3*y - z^2 - 3*z - 2,
1/2*x^3 + 3/2*x^2 + x,
1/2*y^3 + 3/2*y^2 + y,
1/2*z^3 + 3/2*z^2 + z
```

```
]
> A:=radical(F0,R);
> equal(A,F0,R);
true
> IsRadical(I);
true
```

We can consider the computation of the radical of more complicated ideals :

```
> R:=PolynomialRing(RationalField(),2);
> radical([x^2*y^2],R);
[
x^2*y^2,
x*y^2,
x^2*y,
x*y
]
> radical([x^2*y^2,(x+1)^3*(y-1)],R);
[
x^2*y^2,
x^3*y - x^3 + 3*x^2*y - 3*x^2 + 3*x*y - 3*x + y - 1,
x^3 + 3*x^2 - 3*x*y^2 + 3*x - y^2 + 1,
3*x*y^4 - 3*x*y^2 + y^4 - y^2,
3*x*y^3 - 3*x*y^2 + y^3 - y^2,
1/9*y^4 - 1/9*y^2,
1/9*y^3 - 1/9*y^2,
x^2 + x,
-y^2 + y
]
> I:=ideal<R|x^2*y^2,(x+1)^3*(y-1)>;
> Radical(I);
Ideal of Polynomial ring of rank 2 over Rational Field
Lexicographical Order
Variables: x, y
Dimension 0, Radical
Groebner basis:
[
x - y + 1,
y^2 - y
]
> J:=radical([x^2*y^2,(x+1)^3*(y-1)],R);
```

```
> equal(J,[x-y+1,y^2-y],R);
true
> I:=ideal<R|x^2*y^2,(x+1)^3*y>;
> Dimension(I);
1 [ 1 ]
> radical([(x*y)^2,(x+1)^3*y],R);
[
    x^2*y^2,
    3*x*y^2 + y^2,
    x^3*y + 3*x^2*y + 3*x*y + y,
    -1/9*y^2,
    -3*x^2*y - 3*x*y - y,
    9*x*y + 5*y,
    y
]
> J:=radical([(x*y)^2,(x+1)^3*y],R);
> Radical(I);
Ideal of Polynomial ring of rank 2 over Rational Field
Lexicographical Order
Variables: x, y
Radical
Groebner basis:
[
    y
]
> equal(J,[y],R);
true
```

Finally, in positive characteristic, we can see that Matsumoto's algorithm seems very efficient :

```
> T<u,x,y,z>:=PolynomialRing(GF(7),4);
> F:=[z^7-x*y*u^5,y^4-x^3*u];
> radical(F,T);
[
    6*u*x + y*z,
    u*y^2 + 6*x*z^2,
    6*u^2*y + z^3,
    x^2*z + 6*y^3
]
> I:=ideal<T|z^7-x*y*u^5,y^4-x^3*u>;
```

```
> I;
Ideal of Polynomial ring of rank 4 over GF(7)
Lexicographical Order
Variables: u, x, y, z
Basis:
[
    6*u^5*x*y + z^7,
    6*u*x^3 + y^4
]
> Radical(I);
^C
[Interrupted]
```

Indeed, Matsumoto's algorithm, as I have implemented it on Magma, can compute radical of ideals in positive characteristic that even Magma can not compute ! Well at least, its version V2.11-10 don't seem to be able to compute any positive-characteristic radical...

4 Computational Invariant Theory

4.1 Symmetric Polynomials

Invariant Theory is a very anciant topic, beginning with symmetric polynomials : everyone has heard of the fundamental theorem of symmetric polynomials (while not everyone has seen a proof of this theorem...), but we will see that the theory is far much deeper, going to recent developpements. In this section, we will specially consider Computational Invariant Theory, and some of his connection with Gröbner basis. We will indeed begin with symmetric polynomials, and give a proof the fundamental theorem of symmetric polynomials.

4.1.1 Some definitions

Definition 25. We will note, if $n \in \mathbb{N}$, $A \in M_n(k)$, $f \in k[X_1, \ldots, X_n]$, $f(A.X)$ for $f\left(\sum_{k=1}^n a_{1,k}X_k, \ldots, \sum_{k=1}^n a_{n,k}X_k\right)$, $f(A.X) \in k[X_1, \ldots, X_n]$.

Definition 26. We define the elementary symmetric functions $\sigma_1, \ldots, \sigma_n$ of $k[x_1, \ldots, x_n]$ by : $\sigma_1 = x_1 + \cdots + x_n$, ... , $\sigma_r = \sum_{i_1 < \cdots < i_r} x_{i_1} \ldots x_{i_r}$, ..., $\sigma_n = x_1 \ldots x_n$.

We will need a little lemma, which will soon reveal useful in some of the proofs below :

Lemma 12. *Let $\sigma_i^{(n)}$ be the ith elementary symmetric function in variables $x_1, \ldots, x_n$, with $n \in \mathbb{N}^*$ and $i \in \{1, \ldots, n\}$. We set $\sigma_0^{(n)} = 1$ (for all $n \in \mathbb{N}^*$) and $\sigma_i^{(n)} = 0$ if $i < 0$ or $i > n$ ($i \in \mathbb{Z}$).*

Then $\forall n \in \mathbb{N}^*$, $\forall i \in \mathbb{Z}$, $\sigma_i^{(n)} = \sigma_i^{(n-1)} + x_n \sigma_{i-1}^{(n-1)}$.

Proof. Let $n \in \mathbb{N}^*$ and $r \in \mathbb{Z}$.

If $r \leq 0$ or $r > n$, then we indeed have $\sigma_r^{(n)} = \sigma_r^{(n-1)} + x_n \sigma_{r-1}^{(n-1)}$.

If $r = n$, then $\sigma_n^{(n)} = x_1 \ldots x_n$, $\sigma_n^{(n-1)} = 0$ and $x_n \sigma_{n-1}^{(n-1)} = x_1 \ldots x_{n-1} x_n$, so the equality still holds.

If $r = 1$, then $\sigma_1^{(n)} = x_1 + \cdots + x_n$, $\sigma_1^{(n-1)} = x_1 + \cdots + x_{n-1}$ and $x_n \sigma_0^{(n-1)} = x_n$ and again, there is no problem.

Finally, if $r < n$ then :

$$\sigma_r^{(n)} = \sum_{i_1 < \cdots < i_r} x_{i_1} \ldots x_{i_r}.$$

$$\sigma_r^{(n-1)} = \sum_{i_1 < \cdots < i_r < n} x_{i_1} \ldots x_{i_r}.$$

$$x_n \sigma_{k-1}^{(n-1)} = \sum_{i_1 < \cdots < i_{r-1} < n} x_{i_1} \ldots x_{i_{r-1}} x_n.$$

So,

$$\begin{aligned} \sigma_r^{(n-1)} + x_n \sigma_{r-1}^{(n-1)} &= \sum_{i_1 < \cdots < i_r} x_{i_1} \ldots x_{i_r} \\ &= \sigma_r^{(n)} \end{aligned}$$

and the result is proven. □

We set $(-1)^0 = 1$. We are now able to prove this first very famous result :

Proposition 19. *It is well-known that $(X - x_1) \ldots (X - x_n) = \sum_{i=0}^{n} (-1)^i \sigma_i X^{n-i}$, $n \in \mathbb{N}^*$ (with here $\sigma_i = \sigma_i^{(n)}$).*

Proof. If $n = 1$, with $(-1)^0 = 1$, $(-1)^0 * \sigma_0^{(1)} * X + (-1)^1 * \sigma_1^{(1)} * X^0 = X - x_1$ and the result is proven.

We assume that the result holds for $n - 1$ ($n > 1$). Then with the previous lemma :

$$
\begin{aligned}
(X-x_1)\dots(X-x_{n-1})(X-x_n) &= \left(\sum_{i=0}^{n-1}(-1)^i\sigma_i^{(n-1)}X^{n-i-1}\right)(X-x_n)\\
&= \sum_{i=0}^{n-1}(-1)^i\sigma_i^{(n-1)}X^{n-i-1} + \sum_{i=0}^{n-1}(-1)^{i+1}x_n\sigma_i^{(n-1)}X^{n-i-1}\\
&= X^n + \sum_{i=1}^{n-1}(-1)^i\sigma_i^{(n-1)}X^{n-i} + \sum_{i=1}^{n-1}(-1)^i x_n\sigma_{i-1}^{(n-1)}X^{n-i}\\
&= \sum_{i=0}^{n}(-1)^i\sigma_i X^{n-i}
\end{aligned}
$$

and the result is proven. □

Definition 27. A polynomial is called symmetric if for any permutation matrix A, $f(A.X) = f(X)$.

4.1.2 The fundamental theorem of symmetric polynomials

Now, we are able to prove the celebrated fundamental theorem of symmetric polynomials :

Theorem 9. *Every symmetric polynomial in $k[x_1,\dots,x_n]$ can be written uniquely as a polynomial in the elementary symmetric functions $\sigma_1,\dots,\sigma_n$.*

Proof. We consider the lex monomial ordering on $k[x_1,\dots,x_n]$. Let $f \in k[x_1,\dots,x_n]$ be a symmetric polynomial, let $LT(f) = a * x^\alpha$, $a \in k$, we note $\alpha = (\alpha_1,\dots,\alpha_n)$, and we have $\alpha_1, \geq \dots \geq \alpha_n$. Indeed, if $\alpha_i < \alpha_{i+1}$ for some i, we write $\beta = (i \quad i+1)(\alpha)$ (we permute α_i and α_{i+1}), and since x^α is a monomial of f and f is symmetric, x^β is a monomial of f, but $x^\beta >_{lex} x^\alpha$, which is absurd since $x^\alpha = LM(f)$.

Let $h = \sigma_1^{\alpha_1-\alpha_2}\sigma_2^{\alpha_2-\alpha_3}\dots\sigma_{n-1}^{\alpha_{n-1}-\alpha_n}\sigma_n^{\alpha_n}$. Since $LT(\sigma_r) = x_1x_2\dots x_r$ for $1 \leq r \leq n$, we find that $LT(h) = x_1^{\alpha_1}\dots x_n^{\alpha_n} = x^\alpha$.

We set $f_1 = f - ah$. If $f_1 = 0$, then we have the result for f, elsewhere, we have $multideg(f_1 = f - ah) < multideg(f)$, and f_1 is symmetric. Then again we can set $a_1 \in k$ and h_1 which is a polynomial in the σ_i such that $f_2 = f_1 - a_1h_1$, and either $f_2 = 0$ or $multideg(f_2) < multideg(f_1)$.

Thus, we produce a sequence of polynomials $f, f_1, \dots$ with $multideg(f) > multideg(f_1) > multideg(f_2) > \dots$ and since the lex ordering is a weel-ordering, the process of producing f_i must terminate, and the only way it can terminate is with $f_{t+1} = 0$ for some t.

With that, $f = ah + a_1 * h_1 + \dots + a_th_t$, and we have the result for f.

Now, we can consider uniqueness : if $f \in k[x_1, \ldots, x_n]$ is a symmetric polynomial, and if we assume that f can be written $f = g_1(\sigma_1, \ldots, \sigma_n) = g_2(\sigma_1, \ldots, \sigma_n)$ with g_1 and g_2 polynomials in some $k[y_1, \ldots, y_n]$. Let $g = g_1 - g_2$, we have $g(\sigma_1, \ldots, \sigma_n) = 0$ in $k[x_1, \ldots, x_n]$. If we write $g = \sum_\beta a_\beta y^\beta$ and $g_\beta = a_\beta \sigma_1^{\beta_1} \ldots \sigma_n^{\beta_n}$. Like before, we have $LT(g_\beta) = a_\beta x_1^{\beta_1 + \cdots + \beta_n} \ldots x_n^{\beta_n}$. Since the map $(\beta_1, \ldots, \beta_n) \mapsto (\beta_1 + \cdots + \beta_n, \beta_2 + \cdots + \beta_n, \ldots, \beta_n$ is of course injective, the g_β have all distincts leading terms and it is easy to see that with lex order, if $a_\beta y^\beta = LT(g)$ then $LT(g(\sigma_1, \ldots, \sigma_n)) = LT(g_\beta) = a_\beta x_1^{\beta_1 + \cdots + \beta_n} \ldots x_n^{\beta_n}$, and thus if $g(\sigma_1, \ldots, \sigma_n) = 0$, then $g = 0$, and the uniqueness is proven.

□

Given this result, we may wonder how an expression of a symmetric polynomial as a polynomial in the elementary symmmetric functions can be found. An answer lies in Gröbner basis.

Proposition 20. *We consider the ring $k[x_1, \ldots, x_n, y_1, \ldots, y_n]$ with a monomial ordering such that any monomial involving one of the x_i is greater than all the polynomial of $k[y_1, \ldots, y_n]$ (for instance, the lexicographical ordering). Let G be Gröbner basis of $I = \langle \sigma_1 - y_1, \ldots, \sigma_n - y_n \rangle \subset k[x_1, \ldots, x_n, y_1, \ldots, y_n]$. Let $f \in k[x_1, \ldots, x_n]$, let g be the remainder of the division of f by G, then f is symmetric if and only if $g \in k[y_1, \ldots, y_n]$ and then, $f = g(\sigma_1, \ldots, \sigma_n)$ is the unique expression of f as a polynomial in the elementary symmetric functions.*

4.1.3 Miscellaneous

Definition 28. A polynomial $f \in k[x_1, \ldots, x_n]$ is homogeneous of degree k if any term appearing in f is of total degree k. For such a polynomial f, if $x \in k^n$ and $\lambda \in k$, then $f(\lambda x) = \lambda^k f(x)$.

We can always write $f \in k[x_1, \ldots, x_n]$ as $f = \sum_{k=1}^n f_k$ where n is the total degree of f and all f_k are homogeneous of degree k. We may call f_k the k-th homogeneous components of f. Then we have the proposition :

Proposition 21. *A polynomial $f \in k[x_1, \ldots, x_n]$ is symmetric if and only if all of its homogeneous components are symmetric.*

The proof is very straightforward, based on the fact that the action send an homogeneous polynomial of degree l to an homogeneous polynomial of degree l.

Remark. We may notice that the action of the symmetric group provides, for any of its member, a linear morphism of the vector space of homogeneous polynomial of a given degree.

Another object related to symmetric polynomials is Newton's sums.

Definition 29 (Newton's sums)**.** Let $n \in \mathbb{N}^*$ and $k \in \mathbb{N}^*$. Then we define s_k, or $s_k^{(n)}$ if there is an ambiguity, by $s_k = x_1^k + \cdots + x_n^k$.

Newton's identities will give a connection between Newton's sums and elementary symmetric functions :

Proposition 22 (Newton's identities)**.** *Let $n \in \mathbb{N}^*$ and $k \in \mathbb{N}$. Then*

$$s_k - \sigma_1 s_{k-1} + \cdots + (-1)^{k-1}\sigma_{k-1}s_1 + (-1)^k k\sigma_k = 0$$

for $1 \leq k \leq n$, and for $k > n$:

$$s_k - \sigma_1 s_{k-1} + \cdots + (-1)^{n-1}\sigma_{n-1}s_{k-n+1} + (-1)^n \sigma_n s_{k-n} = 0.$$

Proof. We should first notice that if we set, like we did previously, $\sigma_0 = 1$, and $\sigma_i = 0$ if $i < 0$ or $i > n$, then if $k > n$, $\sigma_{n+1}^{(n)} = \cdots = \sigma_k^{(n)}$ and $(-1)^{n+1}\sigma_{n+1}^{(n)}s_{k-n-1}^{(n)} + \cdots + (-1)^{k-1}\sigma_{k-1}^{(n)}s_1^{(n)} + (-1)^k k\sigma_k^{(n)} = 0$. Hence, the second Newton's identity, $s_k - \sigma_1 s_{k-1} + \cdots + (-1)^{n-1}\sigma_{n-1}s_{k-n+1} + (-1)^n\sigma_n s_{k-n} = 0$ if $k > n$, can be rewritten in $s_k^{(n)} - \sigma_1^{(n)}s_{k-1}^{(n)} + \cdots + (-1)^{k-1}\sigma_{k-1}^{(n)}s_1^{(n)} + (-1)^k k\sigma_k^{(n)} = 0$, which has the same expression than the first of Newton's identities.

So, all we have to prove is $\forall n \in \mathbb{N}^*$, $\forall k \in \mathbb{N}^*$:

$$s_k^{(n)} - \sigma_1^{(n)}s_{k-1}^{(n)} + \cdots + (-1)^{k-1}\sigma_{k-1}^{(n)}s_1^{(n)} + (-1)^k k\sigma_k^{(n)} = 0.$$

We proceed by induction on n. We shall keep in mind the fact that for all $n \in \mathbb{N}$, $n \geq 2$, and for all $k \in \mathbb{N}$:

$$\sigma_k^{(n)} = \sigma_i^{(n-1)} + x_n\sigma_{i-1}^{(n-1)},$$

$$s_k^{(n)} = s_k^{(n-1)} + x_n^k.$$

For $n = 1$, $\sigma_1^{(1)} = x_1$, and $s_k^{(1)} = x_1^k$, so if $k \in \mathbb{N}^*$, we indeed have $s_k^{(1)} - \sigma_1^{(1)}s_{k-1}^{(1)} = 0$ and the other terms are 0. So the identity is true for $n = 1$ and for all $k \in \mathbb{N}^*$.

Now, we assume that we have for some $n \in \mathbb{N}^*$, $n \geq 2$:

$$\forall k \in \mathbb{N}^* : s_k^{(n-1)} - \sigma_1^{(n-1)}s_{k-1}^{(n-1)} + \cdots + (-1)^{k-1}\sigma_{k-1}^{(n-1)}s_1^{(n-1)} + (-1)^k k\sigma_k^{(n-1)} = 0$$

Let $k \in \mathbb{N}^*$. If $k = 1$, then we indeed have $s_1^{(n)} = \sigma_1^{(n)}$, and the result holds for $k = 1$. Now, we assume that $k > 1$.

We set $A = \sum_{i=0}^{k-1}(-1)^i\sigma_i^{(n)}s_{k-i}^{(n)} + (-1)^k\sigma_k^{(n)}$. Then :

$$
\begin{aligned}
A &= \sum_{i=0}^{k-1}(-1)^i(\sigma_i^{(n-1)} + x_n\sigma_{i-1}^{(n-1)})(s_{k-i}^{(n-1)} + x_n^{k-i}) + (-1)^k\sigma_k^{(n-1)} + (-1)^k k x_n \sigma_{k-1}^{(n-1)} \\
&= \sum_{i=0}^{k-1}(-1)^i\sigma_i^{(n-1)}s_{k-i}^{(n-1)} + (-1)^k k\sigma_k^{(n-1)} + (-1)^k k x_n\sigma_{k-1}^{(n-1)} \\
&\quad + \sum_{i=0}^{k-1}(-1)^i\sigma_i^{(n-1)}x_n^{k-i} + \sum_{i=0}^{k-1} x_n\sigma_{i-1}^{(n-1)}(-1)^i s_{k-i}^{(n-1)} + \sum_{i=0}^{k-1}(-1)^i x_n^{k-i+1}\sigma_{i-1}^{(n-1)}.
\end{aligned}
$$

We can use our assumption for k and $n-1$, and

$$
\sum_{i=0}^{k-1}(-1)^i\sigma_i^{(n-1)}s_{k-i}^{(n-1)} + (-1)^k k\sigma_k^{(n-1)} = 0.
$$

From now on and until we have proven $A = 0$, since we will only deal with Newton sums and elementary symmetric functions involving the $n-1$ first variables, we will only write s_k and σ_k (instead of $s_k^{(n-1)}$ and $\sigma_k^{(n-1)}$).

$$
\begin{aligned}
A &= (-1)^k k x_n\sigma_{k-1} + \sum_{i=0}^{k-1}(-1)^i\sigma_i x_n^{k-i} + \sum_{i=0}^{k-1}(-1)^i x_n\sigma_{i-1}s_{k-i} + \sum_{i=0}^{k-1}(-1)^i x_n^{k-i+1}\sigma_{i-1} \\
&= x_n\left((-1)^k k\sigma_{k-1} + \sum_{i=0}^{k-1}(-1)^i\sigma_{i-1}s_{k-i}\right) \\
&\quad + \sum_{i=0}^{k-1}(-1)^i\sigma_i x_n^{k-i} + \sum_{i=0}^{k-1}(-1)^i x_n^{k-i+1}\sigma_{i-1}
\end{aligned}
$$

Since $\sum_{i=0}^{k-1}(-1)^i x_n^{k-i+1}\sigma_{i-1} = -\sum_{j=0}^{k-2}(-1)^j x_n^{k-j}\sigma_{i-1}$ ($\sigma_{-1} = 0$), then $\sum_{i=0}^{k-1}(-1)^i\sigma_i x_n^{k-i} + \sum_{i=0}^{k-1}(-1)^i x_n^{k-i+1}\sigma_{i-1} = (-1)^{k-1}\sigma_{k-1}x_n$.

Hence,

$$
\begin{aligned}
A &= x_n\left((-1)^k k\sigma_{k-1} + \sum_{i=0}^{k-1}(-1)^i\sigma_{i-1}s_{k-i} + (-1)^{k-1}\sigma_{k-1}\right) \\
&= -x_n\left((-1)^{k-1}(k-1)\sigma_{k-1} + \sum_{i=0}^{k-1}(-1)^{i-1}\sigma_{i-1}s_{k-i}\right).
\end{aligned}
$$

$\sigma_{-1} = 0$, so $\sum_{i=0}^{k-1}(-1)^{i-1}\sigma_{i-1}s_{k-i} = \sum_{i=0}^{k-2}(-1)^i\sigma_i s_{k-i}$.

We can use our previous assumption for $k-1$ and $n-1$ and

$$
\sum_{i=0}^{k-2}(-1)^i\sigma_i s_{k-i} + (-1)^{k-1}(k-1)\sigma_{k-1} = 0.
$$

Hence, finally :

$$\begin{aligned} A &= -x_n \times \left(\sum_{i=0}^{k-2} (-1)^i \sigma_i s_{k-i} + (-1)^{k-1}(k-1)\sigma_{k-1} \right) \\ &= -x_n \times 0 \\ &= 0, \end{aligned}$$

and the result is proven. □

Theorem 10. *If k is a zero-characteristic field, then every symmetric polynomial in $k[x_1, \ldots, x_n]$ can be written as a polynomial in the s_r, where $s_r = x_1^r + \cdots + x_n^r$, for $r \in \{1, \ldots, n\}$.*

Proof. We have $s_1 = \sigma_1$. If we assume that that for some $l \in \mathbb{N}$, $l > 1$, any σ_t, $t \in \{1, \ldots, l-1\}$, can be written as a polynomial in the s_r, where $s_r = x_1^r + \cdots + x_n^r$, for $r \in \{1, \ldots, n\}$, then from Newton's identities :

$$\sigma_r = (-1)^{r-1} \frac{1}{r} (s_r - \sigma_1 s_{r-1} + \cdots + (-1)^{r-1} \sigma_{r-1} s_1),$$

which can be written since $char(k) = 0$, and σ_r is a polynomial in the s_r, where $s_r = x_1^r + \cdots + x_n^r$, for $r \in \{1, \ldots, n\}$. Hence, by induction, any of the σ_r, $r \in \mathbb{N}^*$, is a polynomial in the s_r, where $s_r = x_1^r + \cdots + x_n^r$, for $r \in \{1, \ldots, n\}$, and finally, with the fundamental theorem of symmetric polynomials, it is also true for any symmetric polynomial. □

4.2 Ring of Invariants under the action of finite matrix groups

4.2.1 Some definitions

Now, we shall generealize our study about the action of the symmetric group of order n to the action of any finite subgroup of $M_n(k)$.

Proposition 23. *A finite subset $G \subset GL_n(k)$ is a group if and only if it is non-empty and closed under matrix multiplication.*

Definition 30. Let $G \subset GL_n(k)$ be a finite matrix group, then $f \in k[X_1, \ldots, X_n]$ is invariant under G if for all $A \in G$, $f(X) = f(A.X)$.

The set of all invariant polynomials is denoted $k[X_1, \ldots, X_n]^G$, and is naturally a subring of $k[X_1, \ldots, X_n]$ containing the constant polynomials, and is called the ring of invariants of $k[X_1, \ldots, X_n]$ under the action of G.

Proposition 24. *Let $G \subset GL_n(k)$ be a finite matrix group, then $f \in k[X_1, \ldots, X_n]$ is invariant under G if and only if all of its homogeneous components are.*

4.2.2 Generators of the ring of invariants

$k[x_1,\dots,x_n]^G$ being defined, we shall study whether $k[x_1,\dots,x_n]^G$ is finitely generated or not, and if we can find some generators for it.

Definition 31. Let $G \subset GL_n(k)$ be a finite matrix group, we define the **Reynolds operator** of G by the map $R_G : k[x_1,\dots,x_n] \to k[x_1,\dots,x_n]$ defined by $R_G(f)(x) = \frac{1}{\sharp G}\sum_{A\in G} f(A.x)$ for all $f \in k[x_1,\dots,x_n]$.

R_G is clearly a k-linear map such that $R_G(k[x_1,\dots,x_n]) \subset k[x_1,\dots,x_n]^G$ and if $f \in k[x_1,\dots,x_n]^G$ then $R_G(f) = f$.

Theorem 11 (Noether)**.** *Let $G \subset GL_n(k)$ be a finite matrix group, with $char(k) = 0$, then $k[x_1,\dots,x_n]^G = k\left[R_G(x^\beta) : |\beta| \leq |G|\right]$. Hence, $k[x_1,\dots,x_n]^G$ is finitely generated by homogeneous invariant polynomials.*

Proof. If $f = \sum_\alpha c_\alpha x^\alpha \in k[x_1,\dots,x_n]^G$, then $f = R_G(f) = R_G\left(\sum_\alpha c_\alpha x^\alpha\right) = \sum_\alpha c_\alpha R_G(x^\alpha)$. Hence, every invariant polynomial is a k-linear combination of the $R_G(x^\alpha)$. We now shall prove that for all α, $R_G(x^\alpha) \in k\left[R_G(x^\beta) : |\beta| \leq |G|\right]$.

We can write that $(x_1 + \dots + x_n)^k = \sum_{|\alpha|=k} a_\alpha x^\alpha$, with the a_α positive natural numbers (and with multinomial formula, we know an expression of the a_α).

If A is a matrix, let A_i denote the i-th row of A. We have by definition $(Ax)^\alpha = (A_1x)^{\alpha_1}\dots(A_nx)^{\alpha_n}$.

We introduce n new variables $u_1,\dots,u_n$, and we now have $(u_1A_1.x + \dots + u_n.A_nx)^k = \sum_{|\alpha|=k} a_\alpha (A.x)^\alpha u^\alpha$. If we sum over all $A \in G$, we get : $\sum_{A\in G}(u_1A_1.x + \dots + u_nA_n.x)^k = \sum_{|\alpha|=k} a_\alpha \left(\sum_{A\in G}(Ax)^\alpha\right) u^\alpha = \sum_{|\alpha|=k} b_\alpha R_G(x^\alpha)u^\alpha$. In fact $b_\alpha = |G|a_\alpha$.

Let $U_A = u_1A_1.x + \dots + u_nA_n.x$, and $S_k = \sum_{A\in G} U_A^k$, then we have $S_k = \sum_{|\alpha|=k} b_\alpha R_G(x^\alpha)u^\alpha$.

With the last theorem of the previous subsection about symmetric polynomials, any polynomial in the U_A and symmetric in the U_A can be written as a polynomial in the S_k, $0 \leq k \leq |G|$.

Hence, if $k \in \mathbb{N}$, $\sum_{|\alpha|=k} b_\alpha R_G(x^\alpha)u^\alpha = S_k = F(S_1,\dots,S_{|G|})$ for some polynomial $F \in k\left[Y_1,\dots,Y_{|G|}\right]$.

Thus, $\sum_{|\alpha|=k} b_\alpha R_G(x^\alpha)u^\alpha = F\left(\sum_{|\beta|=1} b_\beta R_G(x^\beta)u^\beta,\dots,\sum_{|\beta|=|G|} b_\beta R_G(x^\beta)u^\beta\right)$.

We can expand the last expression and we get for the coefficient corresponding to u^α that $b_\alpha R_G(x^\alpha)$ is a polynomial in the $R_G(x^\beta)$ with $|\beta| \leq |G|$.

Since $char(k) = 0$, $|G| \neq 0$ and $b_\alpha = |G|a_\alpha \neq 0$, and we have $R_G(x^\alpha)$ in the desired form. □

Remark. We can also prove the fact that $k[x_1,\dots,x_n]^G$ is finitely generated by homogeneous invariant polynomials in any characteristic, by considering Hilbert's

Basis Theorem, as long as Reynold's operator is well defined, but this proof does not give any bound considering the degrees of generators.

Now we can provide a criteria to decide, given $f_1, \ldots, f_m$, whether a polynomial f is in $k[f_1, \ldots, f_m]$ or not, and if so to find a way to find a polynomial g so as $f = g(f_1, \ldots, f_m)$.

Proposition 25. *We consider the ring $k[x_1, \ldots, x_n, y_1, \ldots, y_m]$ with a monomial ordering such that any monomial involving one of the x_i is greater than all the polynomial of $k[y_1, \ldots, y_m]$. Let G be Gröbner basis of $I = \langle f_1 - y_1, \ldots, f_m - y_m \rangle \subset k[x_1, \ldots, x_n, y_1, \ldots, y_m]$.*

Let $f \in k[x_1, \ldots, x_n]$, let g be the remainder of the division of f by G, then $f \in k[f_1, \ldots, f_m]$ if and only if $g \in k[y_1, \ldots, y_n]$ and then, $f = g(f_1, \ldots, f_n)$ is an expression of f as a polynomial in the $f_1, \ldots, f_m$.

4.3 Hilbert Series and Molien's Formula

4.3.1 Hilbert Series

In this subsection, we will study an important tool for more advanced algorithms to compute generators of an invariant ring, the Hilbert series.

Definition 32. A vector space V which decomposes into a direct sum of the form $V = \bigoplus_{n \in \mathbb{N}} V_d$ is called a **graded vector space** (or a $\mathbb{N}$-graded vector space, since we might consider indexation over another set). If $V = \bigoplus_{d=k}^{+\infty} V_d$ with $k \in \mathbb{Z}$, we also say that V is a graded vector space.

Definition 33. For a graded vector space $V = \bigoplus_{d=k}^{+\infty} V_d$ such that V_d is finite dimensional for all d, we define the **Hilbert series** of V as the formal Laurent series : $H(V, t) = \sum_{d=k}^{+\infty} \dim(V_d) t^d$.

A first easy exemple is the Hilbert Series of $k[x_1, \ldots, x_n]$. We know that there are $\binom{n+d-1}{n-1}$ monomials of total degree d (the number of choices for where to put the parenthesis...), and $k[x_1, \ldots, x_n] = \bigoplus_{d=k}^{+\infty} V_d$ with V_d the subspace of all homogeneous polynomial of degree d. Thus, $H(k[x_1, \ldots, x_n], t) = \sum_{d=0}^{+\infty} \binom{n+d-1}{n-1} t^d$.

With $(1-t)^{-1} = \sum_{d=0}^{+\infty} t^d$, it is easy to see that $(1-t)^{-n} = \sum_{d=0}^{+\infty} \binom{n+d-1}{n-1} t^d$. Hence, $H(k[x_1, \ldots, x_n], t) = (1-t)^{-n}$.

Another easy example is that of $H(k[x], t)$ if x has degree $d > 0$:

Proposition 26. *If x has degree d in $V = k[x]$ seen (naturally) as a graded vector space, then $H(k[x], t) = (1 - t^d)^{-1}$.*

Proof. Indeed, if x has degree d, then $\dim V_l = 0$ if l is not divisible by d, and $\dim V_l = 1$ elsewhere. Thus, $H(k[x], t) = \sum_{k \in \mathbb{N}} (t^d)^k = (1 - t^d)^{-1}$. □

From this example, we can easily deduce the Hilbert Series of $k[x_1, \ldots, x_n]$ where the x_i are of degree $d_i > 0$.

Indeed, if V and W are two graded vector spaces, the their tensor product $V \otimes W$ can naturally be seen as a graded vector space, with $(V \otimes W)_d = \bigoplus_{d_1+d_2=d} V_{d_1} \otimes W_{d_2}$. Then, as obvious Cauchy product, $H(V \otimes W, t) = H(V, t)H(W, t)$.

Therefore, if we consider $k[x_1, \ldots, x_n]$ where x_i is of degree $d_i > 0$, since $k[x_1, \ldots, x_n] = k[x_1] \otimes \ldots k[x_n]$, then $H(k[x_1, \ldots, x_n], t) = \frac{1}{(1-t^{d_1})\ldots(1-t^{d_n})}$.

4.3.2 Molien's Formula

The Hilbert series of an invariant ring can be very easily computed with the so-called Molien's formula. Before enouncing this theorem, we may show a first interesting proposition coming from representation and character theory :

Proposition 27. *Let G be a finite group with a reprentation : a morphism $\rho : G \to GL(V)$, with V a finite dimensional k-vector space, with $char(k) = 0$, then $\dim V^G = \frac{1}{\sharp G} \sum_{g \in G} \chi(g)$, where $\chi : G \to \mathbb{C}$, $g \mapsto tr(\rho(g))$ (χ is a character).*

Proof. Like we did before, we can define $R_G : V \to V$, $v \mapsto \frac{1}{\sharp G} \sum_{g \in G} \rho(g).v$, which is a linear application.

We still have that $Im(R_G) = V^G$ and R_G is the identity on V^G, thus $\dim V^G = tr(R_g)$. Since the trace is a morphism, $\dim V^G = tr(R_G) = \frac{1}{\sharp G} \sum_{g \in G} \chi(g)$ and the result is proven. □

Theorem 12. *Let G be a finite matrix group of $GL_n(k)$ acting on $k[X_1, \ldots, X_n]$, with $char(k) = 0$. Then $H(k[X_1, \ldots, X_n]^G, t) = \frac{1}{\sharp G} \sum_{A \in G} \frac{1}{\det(1-tA)}$.*

Proof. Let $A \in G$. Since G is finite, there exists $n \in \mathbb{N}$ such that $A^n = Id$, and thus, A is diagonaliable, in an algebraic closure of k : with a linear change of coordinates (in this algebraic closure field), we may assume that $A = \begin{bmatrix} a_1 & & O \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$, and thus,

$$\det(Id - tA) = (1 - a_1 t) \ldots (1 - a_n t).$$

Now, if we consider V_d the subspace of the homogeneous polynomial of degree d, with $d \in \mathbb{N}$, and if $\alpha \in \mathbb{N}^n$ with $|\alpha| = d$, then $A.x^\alpha = a_1^{\alpha_1} \ldots \alpha_n^{\alpha_n} x^\alpha$. Hence, we have naturally a basis of eigenvectors, and the aigenvalues are the $a_1^{\alpha_1} \ldots a_n^{\alpha_n}$ with $|\alpha| = d$. Thus, we have that, if we consider the obvious homomorphism : $\rho_d : G \to GL(V_d)$, with the previous proposition, $\chi_d(\rho_d(A)) = \sum_{|\alpha|=d} a_1^{\alpha_1} \ldots a_n^{\alpha_n}$, et $\dim V_d^G = \frac{1}{\sharp G} \sum_{A \in G} \chi_d(\rho_d(A))$.

If we take one $A \in G$, we then obtain that $\sum_{d \in \mathbb{N}} \chi_d(\rho_d(A)) t^d = \sum_{d \in \mathbb{N}} \sum_{|\alpha|=d} a_1^{\alpha_1} \ldots a_n^{\alpha_n} t^d$.

As Cauchy product, we obtain

$$\sum_{d\in\mathbb{N}} \chi_d(\rho_d(A))t^d = \prod_{j=1}^{n}\sum_{k_j=0}^{+\infty}(a_j t)^{k_j} = \prod_{j=1}^{n}\frac{1}{1-a_j t} = \frac{1}{det(Id-tA)}.$$

On the other hand, we can sum on all $A \in G$ to obtain

$$\frac{1}{\sharp G}\sum_{A\in G}\sum_{d\in\mathbb{N}}\chi_d(\rho_d(A))t^d = \sum_{d=0}^{+\infty}\frac{1}{\sharp G}t^d\sum_{A\in G}\chi_d(\rho_d(A)) = \sum_{d=0}^{+\infty}\dim(V_d^G)t^d = H(k\left[X_1,\ldots,X_n\right]^G,t).$$

Finally $H(k\left[X_1,\ldots,X_n\right]^G,t) = \frac{1}{\sharp G}\sum_{A\in G}\frac{1}{\det(1-tA)}$.

□

Remark. This proof is not effective in positive characteristic since we no longer have the fact that the trace of a projector is equal to its rank.

Yet, the result still somehow holds in any characteristic for group whose order is not divisible by the characteristic of the field, with some "lifting to $\mathbb{C}$" morphism to define a determinant and eigenvalues.

Since what we consider is a finite group of matrix, G, of order N. Then the order of any element of G divides N, and therefore, the eigenvalues of the elements of G are all Nth roots of the unity. We can define a group homomorphism between Nth roots of the unity in k and Nth roots of the unity in $\mathbb{C}$.

Then by taking for the trace the sum of the image of the eigenvalues by this morphism, and the determinant to be their product, it can be shown that the result still holds [2].

4.3.3 The Hilbert Series of a Finitely Graded Algebra

The Hilbert series of finitely graded algebra have some very nice, and useful for our computation of generators of invariant ring, properties. It will lead us to define the degree of a finitely graded algebra. We first begin with a lemma :

Lemma 13. *Let* $0 \to^{f_0} E_0 \to^{f_1} \ldots E_{n-1} \to^{f_{n-1}} E_n \to^{f_n} 0 = E_{n+1}$ *be an exact sequence of finite-dimensional vectorial space. Then* $\sum_{i=0}^{n}(-1)^i \dim E_i = 0$.

Proof. We use the rank formula :

$$\dim E_i = \dim\ker f_i + \dim Imf_i = \dim Imf_{i-1} + \dim Imf_i$$

for $i \in \{1,\ldots,n\}$.

Therefore, $\sum_{i=0}^{n}(-1)^i \dim E_i = \dim Imf_0 - (\dim Imf_0 + \dim Imf_1) + \cdots + (-1)^n(\dim Imf_{n-1} + \dim Imf_n) = \dim Imf_0 + (-1)^n \dim Imf_n$ and f_0 and f_n are zero applications, so the result is proven. □

From this, we can deduce this interesting proposition :

Proposition 28. *Let* $0 \to^{f_0} V^{(0)} \to^{f_1} \dots V^{(n-1)} \to^{f_{n-1}} V^{(n)} \to^{f_n} 0$ *be an exact sequence of graded vector space (so, such that all maps respect degree), with* $V_i^{(d)}$ *finite-dimensional for all* i *and* d*. Then*

$$\sum_{i=1}^{n}(-1)^i H(V^{(i)},t)=0.$$

Proof. All maps respect degree, thus we have, for any $d \in \mathbb{N}$, the exact sequence

$$0 \to^{f_0} V_d^{(0)} \to^{f_1} \dots V_d^{(n-1)} \to^{f_{n-1}} V_d^{(n)} \to^{f_n} 0.$$

Hence, $\sum_{i=0}^{n}(-1)^i \dim V_d^{(i)} t^d = 0$. We can sum on all d, and then

$$0=\sum_d\sum_{i=0}^{n}(-1)^i \dim V_d^{(i)} * t^d = \sum_{i=0}^{n}(-1)^i \sum_d \dim V_d^{(i)} = \sum_{i=1}^{r}(-1)^i H(V^{(i)},t).$$

□

We shall admit Hilbert's Syzygy Theorem :

Theorem 13. *Let* k *be a field and* M *be a finitely generated module over the polynomial ring* $R = k[X_1,\dots,X_n]$*. Then* M *admits a free resolution, of length at most* n*, id est there exists* $F_0,\dots,F_r$ $(n \geq r)$ *free* R*-submodules of* M *such that we have the exact sequence :* $0 \to F_r \to \dots F_1 \to F_0 \to M \to 0$*.*

Then, we can demonstrate results about Hilbert Series of a finitely generated algebra.

Theorem 14. *Let* $R = \bigoplus_{d=0}^{\infty}, R_d$ *be a finitely generated graded algebra over a field* $k = R_0$*. Then* $H(R,t)$ *is the power series of a rational function. The radius of convergence of this power series is at least 1.*

If $M = \bigoplus_{d=k}^{\infty} M_d$ *is a finitely generated graded* $R-$*module (same definition as for vector spaces), then* $H(M,t)$ *is the Laurent series of a rational function.*

Proof. Let $A = k[x_1,\dots,x_n]$, such that x_i has degree $d_i > 0$. Then we have already shown that $H(A,t)$ is a rational function, and the radius of convergence of its power series is 1 if $n > 0$ and ∞ if $n = 0$. For any integer e, we can define the A-module $A(e)$ by $A(e) = \bigoplus_{d=-e}^{\infty} A(e)_d$ where $A(e)_d = A_{e+d}$. Clearly, $H(A(e),t) = t^{-e}H(A,t)$ and thus is a rational function. An A-module is free if it is isomorphic to some direct sum $\bigoplus_i A(e_i)$. The Hilbert Series of the direct sum of two graded vector space is the sum of the two Hilbert Series. Therefore a finitely generated free A-module is a rational function.

If M is a finitely generated A-module, then by Hilbert's syzygy theorem, there exists a free resolution $0 \to F_n \to \dots F_1 \to F_0 \to M \to 0$ where $F_0, \dots, F_n$ free R-submodules of M. Since M is finitely generated, then the F_i are also.

Then $\sum_{i=1}^{r+1}(-1)^i H(F_{r-i+1}, t) + (-1)^{r+2} H(M, t) = 0$. Thus

$$H(M,t) = (-1)^{r+1}\sum_{i=1}^{r+1}(-1)^i H(F_{r-i+1},t) = \sum_{i=0}^{r}(-1)^i H(F_i,t).$$

Hence, $H(M,t)$ is indeed a rational function since the $H(F_i,t)$ are. If M is non-negatively graded, the same is true for all of the F_i, and thus, the radius of convergence of $H(M,t)$ is at least 1.

Let R be a finitely generated graded algebra over $k = R_0$. Then for some n and some $d_1, \dots, d_n > 0$, we can define $A = k[X_1, \dots, X_n]$, such that x_i has degree $d_i > 0$ and such that there exists an ideal $I \subset A$ so as $A/I \cong R$. Indeed, we can consider, if R is generated by the $f_1, \dots, f_n$ (f_i of total degree d_i), we can define the morphism of graded algebra : $A \to R$, $x_i \mapsto f_i$. We can set I to be the kernel of this map, and then $A/I \cong R$.

Hence, R is a finitely generated graded A-module, and the result is proven. We can also notice that any finitely generated graded R-module is a finitely generated graded A-module. □

Lemma 14. *Let $R = \bigoplus_{d=0}^{\infty}, R_d$ be a finitely generated graded algebra over a field $k = R_0$, and such that R is a finitely generated module over $k[f_1, \dots, f_r]$, where the f_i are homogeneous (id est any of them belong to one V_d) and algebraically independant. r is the transcendence degree of R over k and then, $r = \dim R$ (Krull dimension) is equal to the pole order of $H(R,t)$ at $t = 1$.*

Proof. We set $A = k[f_1, \dots, f_r]$. We have seen that

$$H(A,t) = \frac{1}{(1-t^{d_1})\dots(1-t^{d_r})}.$$

Since $\frac{1}{1-t^d} = \frac{1}{1-(1+h)^d} \sim_0 \frac{1}{-hd} = \frac{1}{(1-t)d}$, it comes that $H(A,t)$ has pole order r and indeed, $\lim_{t\to 1^-}(1-t)^r H(A,t) = \prod_{i=1}^{r} d_i^{-1}$.

With Hilbert's syzygy lemma, there exists an A-free resolution : $0 \to F_n \to \dots F_1 \to F_0 \to M \to 0$, with $n \leq r$. Hence : $H(M,t) = \sum_{i=0}^{n}(-1)^i H(F_i,t)$. With the F_i finitely generated free A-module, non-negatively graded, the $H(F_i,t)$ have pole order at most r, and then, $H(R,t)$ has also pole order at most r.

Since $A \subset R$, then $H(R,t) \geq H(A,t)$ for $0 < t < 1$. If $H(R,t)$ has pole orderd strictly smaller than r, then

$$0 = \lim_{t\to 1^-}(1-t)^r H(R,t) \geq \lim_{t\to 1^-}(1-t)^r H(A,t) = \prod_{i=1}^{r} d_i^{-1} > 0,$$

which is absurd. So $H(R,t)$ has indeed pole order r at 1. □

Definition 34. Let $R = \bigoplus_{d=0}^{\infty}, R_d$ be a finitely generated graded algebra over a field $k = R_0$, and such that R is a finitely generated module over $k[f_1, \ldots, f_r]$, where the f_i are homogeneous and algebraically independant. $r = \dim R$ and we define **the degree of** R as :

$$deg(R) = \lim_{t \to 1^-} (1-t)^r H(R,t).$$

4.4 Primary and Secondary Invariants

4.4.1 Some Definitions

Now we shall look at how generators of invariant rings could be more efficiently computed. To do so, we will first define what primary and secondary invariants are.

Definition 35. Let $R = \bigoplus_{d=0}^{\infty}, R_d$ be a finitely generated graded algebra over a field $k = R_0$. A set $\{f_1, \ldots, f_r\}$ is called a **homogeneous system of parameters** if $f_1, \ldots, f_r$ are algebraically independent and R is a finitely generated module over $k[f_1, \ldots, f_r]$.

Definition 36. If $f_1, \ldots, f_r \in k[X_1, \ldots, X_n]^G$ is a homogeneous system of parameters of $k[X_1, \ldots, X_n]^G$, then the f_i are called **primary invariants**. If $F = k[f_1, \ldots, f_r]$, then $k[X_1, \ldots, X_n]^G$ is a finitely generated F-module : $k[X_1, \ldots, X_n]^G = Fg_1 + \cdots + Fg_s$ for some $g_s \in k[X_1, \ldots, X_n]^G$. The invariant polynomials $g_1, \ldots, g_s$ are called **secondary invariants**.

Some very interesting things happen in the case of invariant rings :

Proposition 29. *If $n \in \mathbb{N}^*$, if G is a finite matrix group in $GL_n(k)$, k a field, then a set of primary invariants of $k[X_1, \ldots, X_n]^G$ has cardinal n.*

We shall also admit that if G is a finite matrix group in $GL_n(k)$, then $k[X_1, \ldots, X_n]^G$ is a free module as a module over as a module over $k[f_1, \ldots, f_n]$, where the f_i are primary invariants of $k[X_1, \ldots, X_n]^G$.

4.4.2 An Algorithm to Compute Primary Invariants

The following mostly relies on [9].

We will use the following algorithm to compute the primary invariants of $k[X_1, \ldots, X_n]^G$, which we have implemented in Magma (to some extent...).

Algorithm 10 function primary(Pi,R,param), where Pi is a generating set of G, param is an integer parameter, $R = k[X_1, \ldots, X_n]$

```
d:=0;i:=0;
n:=Rank(R); k:=BaseRing(R);
P:=∅;
repeat
  d:=d+1;
  Compute a basis B of k[X_1,...,X_n]_d^G, with Pi and some linear algebra;
  while n − i > dim(⟨P ∪ B⟩) do
    if IsFinite(k) then
      Find a new k-linear combination q of B
      if dim(⟨P⟩) > dim(⟨P ∪ {q}⟩) then
        i:=i+1; Append( P,q);
      end if
      if all elements of Vect(B) have been tried then
        Break the WHILE loop;
      end if
    else
      i:=i+1;
      Find a k-linear combination of B, p_i such that dim(⟨P⟩) >
      dim(⟨P ∪ {p_i}⟩) {and that is the tricky part...}
      Append( P,p_i);
    end if
  end while
until i == n
return P;
```

As written, the tricky part is the "find a k-linear combination of B, p_i such that $\dim(\langle P\rangle) > \dim(\langle P \cup \{p_i\}\rangle)$".

Since in that part of the algorithm, $n - i > \dim(\langle P \cup B\rangle)$ and by definition of P, $i = \sharp P$ and $\dim(\langle P\rangle) = n - i$ (elements of P are algebraically independant), $\dim(\langle P\rangle) > \dim(\langle P \cup B\rangle)$ and therefore, almost any element p_i of B is such that $\dim(\langle P\rangle) > \dim(\langle P \cup \{p_i\}\rangle)$.

Yet, it does not give us a way to compute one of them... For my computation, I had no better idea than introducing a parameter *param* and try combination of elements of B with coefficients in $[-param, param]$ at random until one is found. Hence, if with [9] we can ensure that if our implementation gives a result, it should be correct, we can not ensure a result will be given...

4.4.3 An Algorithm to Compute Secondary Invariants

We have admitted that if G is a finite matrix group in $GL_n(k)$, then $k[X_1, \ldots, X_n]^G$ is a free module as a module over as a module over $F = k[f_1, \ldots, f_n]$, where the f_i are primary invariants of $k[X_1, \ldots, X_n]^G$.

Therefore, we can write :

$$k[X_1, \ldots, X_n]^G = Fg_1 \oplus \cdots \oplus Fg_s$$

for some g_j. Such decomposition of $k[X_1, \ldots, X_n]^G$ is called a **Hironaka decomposition**.

We have seen that $H(F,t) = \frac{1}{\prod(1-t^{d_i})}$ where d_i is the total degree of f_i. Hence,

$$H(k[X_1, \ldots, X_n]^G, t) = \frac{\sum_{j=1}^{s} t^{e_j}}{\prod_{i=1}^{r}(1-t^{d_i})}$$

where e_j is the total degree of g_j.

Therefore, if we know the Molien series of $k[X_1, \ldots, X_n]^G$ and the primary invariants, we can deduce the e_j and therefore, have a good idea of where to find the g_j. That is the idea of the following algorithm.

Algorithm 11 function secundary(P,Pi,G,R), where Pi is a generating set of G and P primary invariants of $k[X_1, \ldots, X_n]^G$, $R = k[X_1, \ldots, X_n]$

$G_i := \emptyset$
G:=grobner(P,R,grevlexord);
Calculate the $e_1, \ldots, e_m$ with P and Molien's formula
for i in $\{1, \ldots, m\}$ **do**
  Compute a basis B of $k[X_1, \ldots, X_n]^G_{e_i}$, with Pi and some linear algebra;
  Find an element of B such that its rest modulo G is linearly independant to the rests of the elements of G_i modulo G
end for
return G_i;

4.5 Computational Results and Exemples

First, we will consider the 'simple' exemple of the action of the cyclic group of order 4 over $GF(7)[X, Y]$:

```
> Q:=PolynomialRing(GF(7),2);
> c4:=ZeroMatrix(Q,2,2);
> c4[1,1]:=0;
> c4[2,1]:=1;
> c4[1,2]:=-1;
> c4[2,2]:=0;
> C4:={c4,c4*c4,c4*c4*c4,c4*c4*c4*c4};
> invhomogeneousbasis(C4,Q);
{
    0,
    4*$.1^4 + 4*$.2^4,
    3*$.1^3*$.2 + 4*$.1*$.2^3,
    4*$.1^2 + 4*$.2^2,
    4*$.1^3*$.2 + 3*$.1*$.2^3,
    $.1^2*$.2^2
}
> primary([c4],Q,0);
[
    $.1^2 + $.2^2,
    $.1^4 + $.2^4
]
> P:=primary([c4],Q,0);
> secundary(P,[c4],C4,Q);
[
    1,
    $.1^3*$.2 + 6*$.1*$.2^3
]
```

We can compare with the results for the action of the 'same' group, but this time, over $\mathbb{Q}[X, Y]$:

```
> R:=PolynomialRing(RationalField(),2);
> c4e:=ZeroMatrix(R,2,2);
> c4e[1,1]:=0;
> c4e[2,1]:=1;
> c4e[1,2]:=-1;
> c4e[2,2]:=0;
```

```
>
> a:=primary([c4e],R,15);
> IsAlgebraicallyDependent({a[1],a[2]});
false
> C4e:={c4e,c4e*c4e,c4e*c4e*c4e,c4e*c4e*c4e*c4e};
> a;
[
    -14*$.1^2 - 14*$.2^2,
    -2*$.1^4 - $.1^3*$.2 + $.1^2*$.2^2 + $.1*$.2^3 - 2*$.2^4
]
> secundary(a,[c4e],C4e,R);
[
    1,
    $.1^4 + $.2^4
]
> molien0(C4e,R);
(t^4 + 1)/(t^6 - t^4 - t^2 + 1)
```

Finally, we can look at the action of the Klein group over $\mathbb{Q}[X_1, \ldots, X_4]$:

```
> M:=PolynomialRing(RationalField(),4);
> a1:=ZeroMatrix(M,4,4);
> a2:=ZeroMatrix(M,4,4);
> a1[2,1]:=1;
> a1[1,2]:=1;
> a1[3,4]:=1;
> a1[4,3]:=1;
> a2[4,1]:=1;
> a2[1,4]:=1;
> a2[2,3]:=1;
> a2[3,2]:=1;
> G:=[a1,a1^2,a1*a2,a2];
> Pi:=[a1,a2];
> molien0(G,M);
(t^2 - t + 1)/(t^6 - 2*t^5 - t^4 + 4*t^3 - t^2 - 2*t + 1)
> P:=primary(Pi,M,2);
> P:=primary(Pi,M,2);
> P;
[
2*$.1 + 2*$.2 + 2*$.3 + 2*$.4,
$.1^2 + 2*$.1*$.2 + 2*$.1*$.4 + $.2^2
```

```
     + 2*$.2*$.3 + $.3^2 + 2*$.3*$.4 + $.4^2,
2*$.1^2 + $.1*$.2 - $.1*$.3 + 2*$.1*$.4
 + 2*$.2^2 + 2*$.2*$.3 - $.2*$.4 + 2*$.3^2 + $.3*$.4 + 2*$.4^2,
$.1^2 + $.2^2 + $.3^2 + $.4^2
]
> secundary(P,Pi,G,M);
[
    1,
    $.1^3 + $.2^3 + $.3^3 + $.4^3
]


invhomogeneousbasis(G,M);
{
    1/4*$.1^2 + 1/4*$.2^2 + 1/4*$.3^2 + 1/4*$.4^2,
    1/4*$.1^4 + 1/4*$.2^4 + 1/4*$.3^4 + 1/4*$.4^4,
    1/4*$.1^2*$.2*$.4 + 1/4*$.1*$.2^2*$.3 + 1/4*$.1*$.3*$.4^2 + 1/4*$.2*$.3^2*$.4,
    1/2*$.1*$.4 + 1/2*$.2*$.3,
    1/2*$.1*$.2 + 1/2*$.3*$.4,
    1/4*$.1^3*$.3 + 1/4*$.1*$.3^3 + 1/4*$.2^3*$.4 + 1/4*$.2*$.4^3,
    1/2*$.1*$.3 + 1/2*$.2*$.4,
    1/4*$.1^2*$.3 + 1/4*$.1*$.3^2 + 1/4*$.2^2*$.4 + 1/4*$.2*$.4^2,
    1/4*$.1*$.2*$.3 + 1/4*$.1*$.2*$.4 + 1/4*$.1*$.3*$.4 + 1/4*$.2*$.3*$.4,
    1/4*$.1^3*$.2 + 1/4*$.1*$.2^3 + 1/4*$.3^3*$.4 + 1/4*$.3*$.4^3,
    1/4*$.1 + 1/4*$.2 + 1/4*$.3 + 1/4*$.4,
    1/4*$.1^3 + 1/4*$.2^3 + 1/4*$.3^3 + 1/4*$.4^3,
    1/4*$.1^2*$.3*$.4 + 1/4*$.1*$.2*$.3^2 + 1/4*$.1*$.2*$.4^2 + 1/4*$.2^2*$.3*$.4,
    1/2*$.1^2*$.2^2 + 1/2*$.3^2*$.4^2,
    1/2*$.1^2*$.4^2 + 1/2*$.2^2*$.3^2,
    1/2*$.1^2*$.3^2 + 1/2*$.2^2*$.4^2,
    1/4*$.1^2*$.2 + 1/4*$.1*$.2^2 + 1/4*$.3^2*$.4 + 1/4*$.3*$.4^2,
    1/4*$.1^2*$.2*$.3 + 1/4*$.1*$.2^2*$.4 + 1/4*$.1*$.3^2*$.4 + 1/4*$.2*$.3*$.4^2,
    1/4*$.1^3*$.4 + 1/4*$.1*$.4^3 + 1/4*$.2^3*$.3 + 1/4*$.2*$.3^3,
    1/4*$.1^2*$.4 + 1/4*$.1*$.4^2 + 1/4*$.2^2*$.3 + 1/4*$.2*$.3^2,
    $.1*$.2*$.3*$.4
}
```

4.6 An exemple of computation of the invariant ring under the action of a not-necessarily finite group

This last subsection present an exemple of the action of a not-necessarily finite group such that we can find generators of the invariant ring. It is not related to the algorithms given previously, but this example can still be found interesting.

We consider the action of $G = GL_n(k)$ on $End(k^n)$ by conjugation : for $\sigma \in GL(k^n)$ and $A \in End(V)$, $\sigma.A = \sigma A \sigma^{-1}$. If $A \in End(k^n)$, its characteristic polynomial is $\chi(t) = \det(tId - A) = t^n - g_{1,A}t^{n-1} + g_{2,A}t^{n-2} - \cdots + (-1)^n g_{n,A}$. We view the g_i as functions of A, we now take a basis of k^n, and see the g_i as polynomials in the coefficients of A written in this basis. We will denote $k[End(k^n)]$ such polynomials. The g_i are of course invariant under the action of $GL(k^n)$. Since we have taken a basis, we now can only talk about matrices.

Our result will be :

Proposition 30. $k[End(k^n)]^G = k[g_1, \ldots, g_n]$.

We consider D, the set of the diagonal matrices. We have the fact that D is invariant under the action of S_n, the subgroup of GL_n of permutation matrix. Restricting χ to D yields the elementary symmetric polynomials in the coefficient of the diagonal. Therefore, the g_i are algebraically independent, and if $h \in k[End(k^n)]^G$, then the restriction of h to D is S_n-invariant and we can find a polynomial ψ such that the restriction of h to D is equal to $\psi(f_1, \ldots, f_n)$.

Let U be the set of matrix that have distinct eigenvalues. $U \subset G.D$ since any matrix in U can be conjugated to a diagonal matrix. Since the application $End(k^n) \to k$, $A \mapsto \Delta(\chi_A)$, where Δ is the discriminant of the polynomial, is continuous (it is a polynomial in the coefficient of A), $\{\Delta(\chi) = 0\}$ is a closed subset of $End(k^n)$. Hence, U, which is the complement of $\{\Delta(\chi) = 0\}$, is an open subset. It is non-empty and since $k[End(k^n)]$ is irreducible, it is Zariski dense as an open and non-empty subset.

Hence, $h - \psi(g_1, \ldots, g_n)$ vanishes on $G.D \supset U$, $G.D$ is dense and $h - \psi(g_1, \ldots, g_n)$ is continuous. So $h = \psi(g_1, \ldots, g_n)$, and finally, $k[End(k^n)]^G = k[g_1, \ldots, g_n]$.

Conclusion

Finally, we are able to compute any basic operation on ideals (intersection, quotient, radical, dimension, ...) or primary and secondary invariants of an invariant ring, given a set of generators of the group we consider, in Magma almost from scratch : the magma functions for polynomials, linear algebra, the lcm, and Molien series in positive characteristic are the only functions of Magma I had to use. Yet, the most advanced computation are not very efficient, and could be improved by implementing the computation of minimal or reduced Gröbner Basis.

Another very interesting field to study would be that of ideal decomposition, which might be necessary to prove that some of the algorithms given here are correct, and which can lead to algorithm that can reveal to be faster for the computation of radicals.

Gröbner basis also have a lot of wonderful applications, which could be worth studying, to geometry, with automatic theorem proofs, and even engineering science !

Thanks

I would like to specially thank Yokoyama Kazuhiro for having been my internship supervisor, for his warm welcome and his attention throughout all my internship.

Thanks to thank Guénaël Renault : without him, I could not have expeienced such a wonderful internship in Japan.

Thanks to Takeshi, Shibuta, all my friends from the grad student's room, and everyone from Rikkyo University, that made feel at home there.

And thank you for reading.

References

[1] Cox, David, Little John & O'Shea Donal Ideals, Varieties and Algorithms (Springer)

[2] Derksen, Harm & Kemper, Gregor Computational Invariant Theory (Springer)

[3] Kreuzer, Martin & Robbiano, Lorenzo Computational Commutative Algebra 1 (Springer)

[4] Greuel, Gert-Martin & Pfister, Gerhard A Singular Introduction to Commutative Algebra (Springer)

[5] Vasconcelos, Wolmer V. Computational Methods in Commutative Algebra and Algebraic Geometry (Springer)

[6] Becker, Thomas & Weispfenning, Volker Gröbner Bases, A Computational Approach to Commutative Algebra

[7] Hartshorne, Robin Algebraic Geometry

[8] Matsumoto, Ryutaroh Computing the Radical of an Ideal in Positive Characteristic (2001)

[9] Decker, Wolfram, Heydtmann, Agnes Eileen & Screyer, Frank-Olaf Generating a Noetherian Normalization of the Invariant Ring of a Finite Group

Annex : Implementation in Magma

order

```
function lexord(f,g,K)
a:=Exponents(K!f);
b:=Exponents(K!g);
test:=true;
n:=#a;
i:=1;
ord:=true;
while (i le n) and test do
if a[i] gt b[i] then
  test:=false;
  ord:=true;
else
  if a[i] lt b[i] then
  test:=false;
  ord:=false;
  else i:=i+1;
  end if;
end if;
end while;
return ord;
end function;

function invlexord(f,g,K)
a:=Exponents(K!f);
b:=Exponents(K!g);
test:=true;
n:=#a;
i:=0;
ord:=true;
while (i lt n) and test do
if a[n-i] gt b[n-i] then
  test:=false;
  ord:=true;
else
  if a[n-i] lt b[n-i] then
  test:=false;
  ord:=false;
  else i:=i+1;
  end if;
```

```
end if;
end while;
return ord;
end function;

function norme(f,K)
if f eq 0 then
return -1;
else
a:=Exponents(K!f);
acc:=0;
n:=#a;
for i in [1..n] do
acc:=acc+a[i];
end for;
return acc;
end if;
end function;

function grlexord(f,g,K)
x:=norme(f,K);
y:=norme(g,K);
ord:=true;
if x lt y then
  ord:=false;
  else if x eq y then ord:=lexord(f,g,K);
end if;
end if;
return ord;
end function;


function invlex(f,g,K)
a:=Exponents(K!f);
b:=Exponents(K!g);
test:=true;
n:=#a;
i:=0;
ord:=true;
while (i lt n) and test do
if a[n-i] gt b[n-i] then
  test:=false;
  ord:=false;
```

```
else
  if a[n-i] lt b[n-i] then
  test:=false;
  ord:=true;
  else i:=i+1;
  end if;
end if;
end while;
return ord;
end function;

function grevlexord(f,g,K)
x:=norme(f,K);
y:=norme(g,K);
if x gt y then
  ord:=true;
  else
    if x eq y then
      ord:=invlex(f,g,K);
    else ord:=false;
    end if;
end if;
return ord;
end function;
```

divisionstool

```
function maxseq(tab,ord,K)
n:=#tab;
if n eq 0 then
return 0,-1;
else
max:=tab[1];
imax:=1;
for i in [1..n] do
if not(ord(max,tab[i],K)) then
  max:=tab[i];
  imax:=i;
end if;
end for;
return imax,max;
end if;
end function;
```

```
function maxcardens(S)
n:=#S;
if n eq 0 then
return 0,{};
else
max:=Random(S);
dmax:=#max;
for V in S do
if #V gt dmax then
  max:=V;
  dmax:=#V;
end if;
end for;
return dmax,max;
end if;
end function;

function intervar(T,S)
intervide:=true;
for s in S do
  for x in T do
    a,b:=IsDivisibleBy(s,x);
    if a then
      intervide:=false;
      break x;
      break s;
    end if;
  end for;
end for;
return intervide;
end function;


function LT(P,K,ord)
if P eq 0 then
return 0;
else
tab:=Monomials(K!P);
imax,max:=maxseq(tab,ord,K);
return Terms(K!P)[imax];
end if;
end function;
```

```
function LTens(F,R,ord)
s:=#F;
F2:={};
for i in [1..s] do
F2:=F2 join {LT(F[i],R,ord)};
end for;
return F2;
end function;

function LC(P,K,ord)
if P eq 0 then
return 0;
else
tab:=Monomials(K!P);
imax,max:=maxseq(tab,ord,K);
return (Coefficients(K!P)[imax]);
end if;
end function;

function LM(P,K,ord)
if P eq 0 then
return 0;
else
tab:=Monomials(K!P);
imax,max:=maxseq(tab,ord,K);
return tab[imax];
end if;
end function;

function normemultidegree(P,R,ord)
return norme(LM(P,R,ord) ,R);
end function;

function lcm(f,g,R)
a:=Exponents(R!f);
b:=Exponents(R!g);
n:=#a;
x:=1;
//on ne considère pas de polynômes nuls !
for i in [1..n] do
x:=x*(R.i)^(Max(a[i],b[i]));
end for;
```

```
return x;
end function;

function SPOL(f,g,R,ord)
x:=lcm(LM(f,R,ord),LM(g,R,ord),R);
a1,b1:=IsDivisibleBy(x,LT(f,R,ord));
a2,b2:=IsDivisibleBy(x,LT(g,R,ord));
return (b1*f-b2*g);
end function;

function division(f,F,K,ord)
r:=0;
p:=f;
s:=#(F);
a:=ZeroMatrix(K,1,s);
while p ne 0 do
  i:=1;
  divisionoccured:=false;
  while (i le s) and not(divisionoccured) do
    u:=LT(p,K,ord);
    v:=LT(F[i],K,ord);
    t,w:=IsDivisibleBy(u,v);
    //if u is divisible by v, then t is true and w is u/v, else t is false
    if t then
      a[1,i]:=a[1,i]+w;
      p:=p-F[i]*w;
      divisionoccured:=true;
    else
      i:=i+1;
    end if;
  end while;
  if not(divisionoccured) then
      r:=r+u;
      p:=p-u;
  end if;
end while;
return a,r;
end function;
```

grobner

```
function grobner(F,R,ord)
G:=F;
```

```
while (0 in G) do
Exclude(~G,0);
end while;
n:=#G;
if n eq 0 then
return [R!0];
else
list:=[[G[p],G[q]] : p in [1..n],q in [1..n] | p lt q];
while #list gt 0 do
a:=list[1];
p:=a[1];q:=a[2];
Remove(~list,1);
a,S:=division(SPOL(p,q,R,ord),G,R,ord);
if S ne 0 then
Append(~(G),S);
m:=#G;
l2:=[[G[u],G[v]] : u in [1..(m-1)],v in [m..m]];
list cat:= l2;
end if;
end while;
return G;
end if;
end function;
```

operationstool

```
function appartientelim(P,k,R)
i:=1;
app:=true;
while (i lt k) and app do
app:= Degree(R!P,i) le 0;
i:=i+1;
end while;
return app;
end function;

function elimination(F,k,R)
G:=grobner(F,R,lexord);
n:=#G;
F:=[];
for i in [1..n] do
if appartientelim(G[i],k,R) then
Append(~F,G[i]);
```

```
end if;
end for;
return F;
end function;

function appartientelimset(P,S,R)
i:=1;
app:=true;
for i in S do
if (Degree(R!P,i) gt 0) then
app:= false;
break i;
end if;
end for;
return app;
end function;

function eliminationset(F,S,R,ord)
G:=grobner(F,R,ord);
n:=#G;
F:=[];
for i in [1..n] do
if appartientelimset(G[i],S,R) then
Append(~F,G[i]);
end if;
end for;
return F;
end function;

function expand(R,l)
n:=Rank(R);
K:=BaseRing(R);
R2:=PolynomialRing(K,n+l);
Z:=[R2.i : i in [1..n]];
f:=hom<R->R2 | Z>;
return f,R2;
end function;

function proj(R,l)
n:=Rank(R);
K:=BaseRing(R);
R2:=PolynomialRing(K,l);
Z:=[R2.i : i in [1..l]] cat [0 : i in [(l+1)..n]];
```

```
f:=hom<R->R2 | Z>;
return f,R2;
end function;
```

operations

```
function intersection(F1,F2,R)
n:=Rank(R);
n1:=#F1;
n2:=#F2;
f,R2:=expand(R,1);
F3:=[f(F1[i])*R2.(n+1) : i in [1..n1]] cat [f(F2[i])*(1-R2.(n+1)) : i in [1..n2]];
G:=grobner(F3,R2,invlexord);
G2:=eliminationset(G,{n+1},R2,invlexord);
g,R3:=proj(R2,n);
s:=#G2;
G3:=[R!(g(G2[i])) : i in [1..s]];
return G3;
end function;

function colon1(F,f,R)
G:=intersection(F,[f],R);
n:= #G;
G2:=[];
for i in [1..n] do
a,b:=IsDivisibleBy(G[i],f);
G2 cat:= [b];
end for;
return G2;
end function;

function colon(F,G,R)
Iacc:=colon1(F,G[1],R);
k:=#G;
for i in [2..k] do
Iacc:=intersection(Iacc,colon1(F,G[i],R),R);
end for;
return Iacc;
end function;

function saturation(F,f0,R)
n:=Rank(R);
n1:=#F;
```

```
f,R2:=expand(R,1);
F3:=[f(F[i]) : i in [1..n1]] cat [f(f0)*R2.(n+1)-1];
G:=grobner(F3,R2,lexord);
G2:=eliminationset(G,{n+1},R2,invlexord);
g,R3:=proj(R2,n);
s:=#G2;
G3:=[R!(g(G2[i])) : i in [1..s]];
return G3;
end function;



function dimrec(S,k,U,M,R)
M2:=M;
n:=Rank(R);
for i in [k..n] do
  inter:=true;
  U2:={1..n} diff (U join {i});
  for s in S do
    if appartientelimset(s,U2,R) then
      inter:=false;
      break s;
    end if;
  end for;
  if inter then
      M2:=dimrec(S,i+1,U join {i},M2,R);
  end if;
end for;
ncontain:=true;
for V in M2 do
  if U subset V then
    ncontain:=false;
    break V;
  end if;
end for;
if ncontain then
  M2:=M2 join {U};
end if;
return M2;
end function;

function dim(G,R,ord)
M:=dimrec(LTens(G,R,ord),1,{},{},R);
```

```
d,U:=maxcardens(M);
return M,d,U;
end function;


function dimension(F,R)
G:=grobner(F,R,grevlexord);
s:=#G;
n:=Rank(R);
for i in [1..s] do
l:=normemultidegree(G[i],R,grevlexord);
if  l eq 0 then
  return {},-1,{};
  break i;
else
  if l lt 0 then
    Remove(~G,i);
  end if;
end if;
end for;
return dim(G,R,grevlexord);
end function;


function equal(F1,F2,R)
equ:=true;
G1:=grobner(F1,R,grevlexord);
for i in [1..(#(F2))] do
a,b:=division(F2[i],G1,R,grevlexord);
if b ne 0 then
  equ:= false;
  break i;
end if;
end for;

if equ then
G2:=grobner(F2,R,grevlexord);
for i in [1..(#(F1))] do
a,b:=division(F1[i],G2,R,grevlexord);
if b ne 0 then
  equ:= false;
  break i;
end if;
```

```
end for;
end if;

return equ;
end function;
```

radicalstool

```
function reduceuni(f,R)
a,P,i:=IsUnivariate(f);
n:=Rank(R);
K:=BaseRing(R);
R2:=PolynomialRing(K);
a,P,i:=IsUnivariate(f);
Z:=([0 : j in [1..(i-1)]] cat [R2.1]) cat [0 : j in [(i+1)..n]];
g:=hom<R->R2 | Z>;
return g,R2;
end function;

function reexpand(i,R2,R)
Z:=[R.i];
g:=hom<R2->R | Z>;
return g;
end function;

function squarefreezero(u,R)
if u eq 0 then
return 0;
else
x,y,i:=IsUnivariate(u);
s,R2:=reduceuni(u,R);
f:=s(u);
s1,a,b:=XGCD(f,Derivative(f));
a,b:=IsDivisibleBy(f,s1);
inj:=reexpand(i,R2,R);
return (inj(b));
end if;
end function;


function univariate(F,R)
n:=Rank(R);
k:=BaseRing(R);
```

```
G:=grobner(F,R,grevlexord);
U:=[];
for i in [1..n] do
  trouve:=false;
  d:=0;
  t,L:=division(R.i^d,G,R,grevlexord);
//we assume 1 is not in I...
  M:=Monomials(R!L);
  C:=Coefficients(R!L);
  s:=#M;
  M2:=Seqset(M);
//ensemble de tout les monomes considérés
  l:={<M[i],i> : i in [1..s]};
  a:=map<M2->{1..s} |l >;
  matrice0:=ZeroMatrix(k,s,1);

  for j in [1..s] do
    matrice0[a(M[j]),1]:=C[j];
  end for;

  while not(trouve) do
    d:=d+1;
    t,L:=division(R.i^d,G,R,grevlexord);
    M3:=Monomials(R!L);
    C:=Coefficients(R!L);
    s2:=#M3;
    for i2 in [1..s2] do
      if not(M3[i2] in M2) then
M2:=M2 join {M3[i2]};
l:= l join {<M3[i2],#M2>};
      end if;
    end for;
    s:=#M2;
    a:=map<M2->{1..s} |l >;
    matrice:=ZeroMatrix(k,s,d+1);

    InsertBlock(~matrice, matrice0, 1, 1);
    matrice0:=matrice;
    for j2 in [1..s2] do
      matrice0[a(M3[j2]),d+1]:=C[j2];
    end for;

    base:=Basis(NullspaceOfTranspose(matrice0));
```

```
    if #base ge 1 then
      trouve:=true;
    end if;
  end while;

  acc:=0;
  for l in [0..d] do
    acc:=acc+base[1][l+1]*(R.i)^l;
  end for;
  U cat:=[acc];
end for;

return U;
end function;



function contraction(F,U,R)
G:=grobner(F,R,lexord);
s:=#G;
r:=#U;
k:=BaseRing(R);
n:=Rank(R);
Uc:={1..n} diff U;
k2:=FunctionField(k,#(Uc));
R2:=PolynomialRing(k2,r);

Y:=[R2.i : i in [1..r]] cat [k2.i : i in [1..(#(Uc))]];
f1:=hom<R->R2 | Y>;
H2:=[f1(G[i]) : i in [1..s]];

Z:=[R.(r+i) : i in [1..(#(Uc))]];
f2:=hom<k2->R | Z>;

lead:=[f2(LC(H2[i],R2,lexord)) : i in [1..s]];

//now lcm...

u:=LCM(lead);


//LCM function not ok for RealField()...
```

```
return u,H2,R2;
end function;
```

```
function renumber(U,R);
n:=Rank(R);
k:=BaseRing(R);
R2:=PolynomialRing(k,n);
Z1:=[];
Z2:=[];
Z:=[];
i:=1;
Uc:={1..n} diff U;
for x in U do
Z1 cat:= [<R.x,R2.i>];
Z2 cat:= [R.x];
i:=i+1;
end for;
for x in Uc do
Z1 cat:= [<R.x,R2.i>];
Z2 cat:= [R.x];
i:=i+1;
end for;
Z3:=Reverse(Sort(Z1));
for i in [1..n] do
Z cat:= [Z3[i][2]];
end for;
f:=hom<R->R2 | Z>;
g:=hom<R2->R | Z2>;
return f,g,R2;
end function;
```

```
function retour(J,R2,R3)
G:=grobner(J,R3,grevlexord);
s:=#G;
for i in [1..s] do
  acc:=1;
  for t in Coefficients(G[i]) do
    acc:=acc*Denominator(t);
  end for;
  G[i]:=acc*G[i];
end for;

Lead:=[LC(G[i],R3,grevlexord) : i in [1..s]];

r:=Rank(R3);
L:=BaseRing(R3);
n:=r+Rank(L);

Z:=[R2.i : i in [(r+1)..n]];
Z2:=[R2.i : i in [1..r]];
a:=hom<L->R2 | Z>;
b:=hom<R3->R2 | Z2>;

g:=function(h)
acc:=0;
M:=Monomials((R3)!h);
C:=Coefficients((R3)!h);
m:=#M;
for i in [1..m] do
acc:= acc+ b(M[i])*a(C[i]);
end for;
return acc;
end function;

Lead2:=[g(Lead[i]) : i in [1..s]];
F:=[g(G[i]) : i in [1..s]];

// and lcm... still not ok with RealField()

f:=LCM(Lead2);

return saturation(F,f,R2);
end function;
```

```
function rac(f,p)
g:=0;
L:=Monomials(f);
M:=Coefficients(f);
n:=#L;
for i in [1..n] do
if M[i] ne 0 then
g:=g+M[i]^(1/p)*(Parent(f)).1^(IntegerRing()!((i-1)/p));
end if;
end for;
return g;
end function;


function squarefreep(u,p,R)
x,y,i:=IsUnivariate(u);
s,R2:=reduceuni(u,R);
f:=s(u);
s1:=GCD(f,Derivative(f));
while s1 ne 1 do
  if Derivative(s1) eq 0 then
    g:=rac(s1,p);
  else
    s2:=GCD(s1,Derivative(s1));
    while Derivative(s2) ne 0 do
      s2:=GCD(s2,Derivative(s2));
    end while;
    g:=rac(s2,p);
  end if;
  a,b:=IsDivisibleBy(f*g,s1);
  f:=b;
  s1:=GCD(f,Derivative(f));
end while;
inj:=reexpand(i,R2,R);
return (inj(f));
end function;
```

```
function raccoef(f,p,R)
g:=0;
L:=Monomials(R!f);
M:=Coefficients(R!f);
n:=#L;
for i in [1..n] do
if M[i] ne 0 then
g:=g+Root(M[i],p)*L[i];
end if;
end for;
return g;
end function;
```

radical

```
G:=F;
U:=univariate(F,R);
s:=#U;
K:=BaseRing(R);
p:=Characteristic(K);
if p eq 0 then
  for i in [1..s] do
    Append(~G,squarefreezero(U[i],R));
  end for;
else
  for i in [1..s] do
    Append(~G,squarefreep(U[i],p,R));
  end for;
end if;
return G;
end function;

function radical0(F,R)
n:=Rank(R);
G:=F;
s:=#G;
M,d,U0:=dimension(G,R);
U:={1..n} diff U0;
r:=#U;
if d eq (-1) then
  return G;
```

```
else if d eq 0 then
  return zerodimradical(G,R);
  else
    f,g,R2:=renumber(U,R);
    H:=[f(G[i]) : i in [1..s]];
    u,H2,R3:=contraction(H,{1..r},R2);
    J:=zerodimradical(H2,R3);
    Jc:=retour(J,R2,R3);
    L:=radical0(Append(H,u),R2);
    rad:=intersection(L,Jc,R2);
    s2:=#rad;
    rad2:=[g(rad[i]):i in [1..(s2)]];
    return rad2;
  end if;
end if;
end function;

function radicalp(B,q,R)
C:=B;
trouve:=false;
repeat
B0:=[];
s:=#C;
n:=Rank(R);
for i in [1..s] do
  B0 cat:=[raccoef(C[i],q,R)];
end for;
f,R2:=expand(R,n);
B1:=[f(B0[i]) : i in [1..s]];
B1 cat:= [ R2.(n+i)-(R2.i)^q : i in [1..n]];
B1:=grobner(B1,R2,lexord);
B2:=elimination(B1,n+1,R2);
Z:=[R2.i : i in [1..n]];
Z cat:= Z;
a:=hom<R2->R2|Z>;
g,R3:=proj(R2,n);
B3:=[R!(g(a(B2[i]))) : i in [1..(#(B2))]];
if equal(C,B3,R) then
  trouve:=true;
else C:=B3;
end if;
until trouve;
return B3;
```

```
end function;


function radical(F,R)
k:=BaseRing(R);
p:=Characteristic(k);
G:=grobner(F,R,grevlexord);
if p eq 0 then
return radical0(G,R);
else
return radicalp(G,p,R);
end if;
end function;
```

invariants

```
function sigma(R)
F:=[];
n:=Rank(R);
S:={R.i : i in [1..n]};
for i in [1..n] do
T:=Subsets(S,i);
U:={&*v : v in T};
F cat:=[&+U];
end for;
return F;
end function;

function symmetricelementary(f,R)
n:=Rank(R);
F:=sigma(R);
g1,R1:=expand(R,n);
F2:=[g1(F[i]) : i in [1..n]];
G0:=[F2[i]-R1.(i+n) : i in [1..n]];
G:=grobner(G0,R1,lexord);
A,r:=division(g1(f),G,R1,lexord);
if appartientelim(r,n+1,R1) then
  R2:=PolynomialRing(BaseRing(R),n);
  h:=hom<R1->R2 | [ 0 : i in [1..n]] cat [R2.i : i in [1..n]]>;
  return true,h(r);
else
  return false,0;
end if;
```

```
end function;

function action(f,A,R)
n:=Rank(R);
x:=ZeroMatrix(R,n,1);
for i in [1..n] do
  x[i,1]:=R.i;
end for;
y:=A*x;
g:=hom<R->R | [y[i,1] : i in [1..n]]>;
return g(f);
end function;

function reynold(f,G,R)
N:=#G;
S:=[action(f,A,R) : A in G];
return (&+S)/N;
end function;

function monomborne(N,R)
n:=Rank(R);
a:=CartesianPower({0..N},n);
x:={[y[i] : i in [1..n]] : y in a};
y:={u : u in x | &+u le N};
z:={&*{(R.i)^(v[i]) : i in [1..n]} : v in y};
return z;
end function;

function invhomogeneousbasis(G,R)
N:=#G;
n:=Rank(R);
a:=CartesianPower({0..N},n);
x:={[y[i] : i in [1..n]] : y in a};
y:={u : u in x | (&+u le N) and (&+u ge 1)};
z:={reynold(&*{(R.i)^(v[i]) : i in [1..n]},G,R) : v in y};
return z;
end function;

function invariantexpression(f,G,R)
n:=Rank(R);
F:=Setseq(invhomogeneousbasis(G,R));
m:=#F;
g1,R1:=expand(R,m);
```

```
F2:=[g1(F[i]) : i in [1..m]];
G0:=[F2[i]-R1.(i+n) : i in [1..m]];
G:=grobner(G0,R1,lexord);
A,r:=division(g1(f),G,R1,lexord);
if appartientelim(r,n+1,R1) then
  R2:=PolynomialRing(BaseRing(R),m);
  h:=hom<R1->R2 | [ 0 : i in [1..n]] cat [R2.i : i in [1..m]]>;
  return true,h(r);
else
  return false,0;
end if;
end function;
```

primary

```
function randomcombi(B,R,param)
n:=#B;
a:=ZeroMatrix(R,1,n);
b:=ZeroMatrix(R,n,1);
for i in [1..n] do
  a[1,i]:=B[i];
  b[i,1]:=Random(-param,param);
end for;
return (a*b)[1,1];
end function;
```

```
function primary(Pi,R,param)
d:=0;
i:=0;
n:=Rank(R);
k:=BaseRing(R);
finiteness,f2:=IsFinite(k);
kg:=#Pi;
P:=[];
G0:=[];
trouve:=false;

repeat
d:=d+1;
dim:=Binomial(n+d-1,n-1);
```

```
mon:=Setseq(MonomialsOfDegree(R,d));

a:=ZeroMatrix(k,dim,dim);
V:=Kernel(a);

for i0 in [1..kg] do
a:=ZeroMatrix(k,dim,dim);
  for j in [1..dim] do
    q,r:=division(action(mon[j],Pi[i0],R),mon,R,lexord);
    for l in [1..dim] do
a[l,j]:=k!(q[1,l]);
    end for;
  end for;
//a is normally the matrix of Pi[i0] (as linear morphism over k[x_1,...,x_n]_d)

V:= V meet Kernel(Transpose(a-ScalarMatrix(dim, 1)));
end for;

B0:=Basis(V);
cd:=#(B0);
B:=[];

for j in [1..cd] do
  acc:=0;
  for l in [1..dim] do
    acc:=acc+B0[j][l]*mon[l];
  end for;
  Append(~(B),acc);
end for;

u,v,w:=dimension(P cat B,R);
u0,v0,w0:=dimension(P,R);

if not(finiteness) then
while (n-i) gt v do
    i:=i+1;
    trouve:=false;
    while not(trouve) do
      pi:=randomcombi(B,R,param);
      u1,v1,w1:=dimension(P cat [pi],R);
if v0 gt v1 then
  Append(~P,pi);
  u,v,w:=dimension(P cat B,R);
```

```
  u0,v0,w0:=dimension(P,R);
  trouve:=true;
end if;
    end while;
end while;
else
for pi0 in V do
  if (n-i) le v then
    break;
  else
      if pi0 ne 0 then
pi:=0;
for l in [1..dim] do
  pi:=pi+pi0[l]*mon[l];
end for;
u1,v1,w1:=dimension(P cat [pi],R);
if v0 gt v1 then
  Append(~P,pi);
  i:=i+1;
  u,v,w:=dimension(P cat B,R);
  u0,v0,w0:=dimension(P,R);
end if;
      end if;
  end if;
end for;
end if;

until n eq i ;
return P;
end function;
```

molien

```
function molien0(G,R)
N:=#G;
n:=Rank(R);
k:=BaseRing(R);
L<t>:=RationalFunctionField(R);
M:=MatrixAlgebra(L,n);
S:=[1/Determinant(ScalarMatrix(n, 1)-ScalarMatrix(n,t)*(M!A)) : A in G];
return (&+S)/N;
end function;
```

secundary

```
function isinde(l,P)
if P eq 0 then
return false;
else
monom:={};
k:=BaseRing(Parent(P));
m:=#l;
if m eq 0 then
return true;
else
for p in l do
  monom:=monom join Seqset(Monomials(p));
end for;
monom:=monom join Seqset(Monomials(P));
monom:=Setseq(monom);
n:=#monom;
M:=ZeroMatrix(k,n,m+1);
for j in [1..m] do
  for i in [1..n] do
    M[i,j]:=MonomialCoefficient(l[j], monom[i]);
  end for;
end for;
for i in [1..n] do
    M[i,m+1]:=MonomialCoefficient(P, monom[i]);
end for;
return (Dimension(Kernel(M)) eq 0);
end if;
end if;

end function;



function secundary(P,Pi,Group,R)
G:=grobner(P,R,grevlexord);
k:=BaseRing(R);
n:=Rank(R);
g:=[];
NFg:=[];
kg:=#Pi;

if Characteristic(k) eq 0 then
```

```
  H:=molien0(Group,R);
else
  H:=MolienSeries(MatrixGroup<n,k|Pi>);
end if;
deg:=[TotalDegree(f) : f in P];
N:=Numerator(H*(&*([1-(Parent(H).1)^di : di in deg])));
Mon:=Monomials(N);
e:=[];

for a in Mon do
  Append(~e,Degree(a));
end for;

m:=#e;

for i in [1..m] do
  d:=e[i];
  dim:=Binomial(n+d-1,n-1);
  mon:=Setseq(MonomialsOfDegree(R,d));

  a:=ZeroMatrix(k,dim,dim);
  V:=Kernel(a);

  for i0 in [1..kg] do
    a:=ZeroMatrix(k,dim,dim);
    for j in [1..dim] do
      q,r:=division(action(mon[j],Pi[i0],R),mon,R,lexord);
      for l in [1..dim] do
a[l,j]:=k!(q[1,l]);
      end for;
    end for;
//a is normally the matrix of Pi[i0] (as linear morphism over k[x_1,...,x_n]_d)

    V:= V meet Kernel(Transpose(a-ScalarMatrix(dim, 1)));
  end for;

  B0:=Basis(V);
  cd:=#(B0);

  for j in [1..cd] do
    acc:=0;
    for l in [1..dim] do
      acc:=acc+B0[j][l]*mon[l];
```

```
    end for;
    gi:=acc;
    q,NFgi:=division(gi,G,R,grevlexord);
    if isinde(NFg,NFgi) then
Append(~g,gi);
Append(~NFg,NFgi);
break j;
    end if;
  end for;
end for;

return g;

end function;
```

In what order should they be loaded ?

```
load "stage 2010/order";
load "stage 2010/divisionstool";
load "stage 2010/grobner";
load "stage 2010/operationstool";
load "stage 2010/operations";
load "stage 2010/radicalstool";
load "stage 2010/radical";
load "stage 2010/invariants";
load "stage 2010/primary";
load "stage 2010/molien";
load "stage 2010/secundary";
```